

z/OS  
2.4

*IBM z/OS Management Facility  
Programming Guide*



**Note**

Before using this information and the product it supports, read the information in [“Notices” on page 1003](#).

This edition applies to Version 2 Release 4 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2021-06-23

© **Copyright International Business Machines Corporation 2013, 2021.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures.....</b>	<b>ix</b>
<b>Tables.....</b>	<b>xxix</b>
<b>About this document.....</b>	<b>xlx</b>
Who should use this document.....	xlx
Where to find more information.....	xlx
<b>How to send your comments to IBM.....</b>	<b>li</b>
If you have a technical problem.....	li
<b>Summary of changes.....</b>	<b>liii</b>
Summary of changes for z/OSMF Programming Guide for Version 2 Release 4 (V2R4).....	liii
Summary of changes made in z/OSMF Version 2 Release 3, as updated June 2019.....	lvii
Summary of changes made in z/OSMF Version 2 Release 3, as updated March 2019.....	lviii
Summary of changes made in z/OSMF Version 2 Release 3, as updated December 2018.....	lviii
Summary of changes made in z/OSMF Version 2 Release 3, as updated August 2018.....	lix
Summary of changes made in z/OSMF Version 2 Release 3, as updated July 2018.....	lix
Summary of changes made in z/OSMF Version 2 Release 3, as updated June 2018.....	lx
Summary of changes made in z/OSMF Version 2 Release 3, as updated March 2018.....	lxi
Summary of changes made in z/OSMF Version 2 Release 3, as updated December 2017.....	lxi
Summary of changes made in z/OS Version 2 Release 3, as updated October 2017.....	lxii
Summary of changes made in z/OS Version 2 Release 3.....	lxii
Summary of changes made in z/OSMF Version 2 Release 2, as updated April 2017.....	lxiii
Summary of changes made in z/OSMF Version 2 Release 2, as updated March 2017.....	lxiv
Summary of changes made in z/OSMF Version 2 Release 2, as updated December 2016.....	lxiv
Summary of changes made in z/OSMF Version 2 Release 2, as updated September 2016.....	lxvi
Summary of changes made in z/OSMF Version 2 Release 2, as updated June 2016.....	lxvii
Summary of changes made in z/OSMF Version 2 Release 2, as updated March 2016.....	lxvii
Changes made in z/OSMF Version 2 Release 2, SC27-8420-00.....	lxvii
Information applicable to all releases.....	lxviii
<b>Chapter 1. Using the z/OSMF REST services.....</b>	<b>1</b>
Application Linking Manager interface services.....	7
Event types, requestors, and handlers shipped with z/OSMF.....	9
Register an event type.....	23
Register an event handler.....	25
Obtain a list of all tasks that are eligible to be handlers.....	28
Obtain a list of handlers for an event type.....	29
Unregister an event handler.....	30
Unregister an event type.....	31
Application server routing services.....	32
Retrieve data from an application server.....	35
Update data for an application server.....	39
Delete data from an application server.....	42
Cloud provisioning services.....	45
Resource pool services.....	49
Resource management services.....	76
Software services template services.....	163

Published software service template services.....	250
Software services instance services.....	286
Software service instance name services.....	356
Data persistence services.....	366
Persist user or application data.....	367
Retrieve persisted user or application data.....	369
Delete persisted user or application data.....	372
Multisystem routing services.....	374
Retrieve data from one or more systems.....	379
Update data for one or more systems.....	386
Delete data from one or more systems.....	392
Authenticate with a secondary z/OSMF instance.....	398
Authenticate with an HTTP proxy server.....	400
MVS subsystem services.....	401
List MVS subsystems.....	402
Notification services.....	404
Get all of the notifications received by the current user.....	405
Send a notification from a z/OSMF task, when the content is the message from the bundle file ..	407
Send a notification and mail from a z/OSMF task or z/OSMF user.....	409
Software management services.....	412
List the software instances defined to z/OSMF.....	414
Retrieve the properties of a software instance.....	417
List the data sets included in a software instance.....	423
Add a new software instance.....	428
Export a defined software instance.....	434
Modify the properties of a software instance.....	440
Load the products, features, and FMIDs for a software instance.....	446
Delete a software instance.....	452
List the portable software instances defined to z/OSMF.....	453
Retrieve the properties of a portable software instance.....	455
Add a new portable software instance.....	460
Delete a portable software instance.....	464
Topology services.....	465
List the systems defined to z/OSMF.....	467
List the groups defined to z/OSMF.....	470
List the systems included in a group.....	471
List the sysplexes defined to z/OSMF.....	474
List the systems included in a sysplex.....	476
List the systems included in a CPC.....	478
TSO/E address space services.....	481
Start or reconnect to a TSO/E address space.....	485
Start an application in a TSO/E address space.....	490
Issue a TSO/E command with z/OSMF REST API.....	492
Get the response to a command that was issued with the TSO/E REST API.....	501
Send messages to a TSO/E address space.....	504
Send messages to an application.....	506
Ping a TSO/E address space.....	507
Receive messages from a TSO/E address space.....	509
Receive messages from an application.....	510
End a TSO/E address space.....	511
WLM resource pooling services.....	513
Prime a WLM resource pool.....	514
Delete a WLM resource pool.....	516
Construct a WLM service definition.....	517
Construct a WLM service definition with remove and install.....	519
RMF metering services.....	521
Get metered data.....	522
Get DDS server time data.....	525

z/OS console services.....	526
Issue a command from a system console.....	528
Get a command response.....	545
Get the detect result for unsolicited messages.....	551
Get messages from logs.....	558
z/OS data set and file REST interface.....	563
List the z/OS data sets on a system.....	569
List the members of a z/OS data set.....	572
Retrieve the contents of a z/OS data set or member.....	575
Write data to a z/OS data set or member.....	580
Create a sequential or partitioned data set.....	585
Delete a sequential and partitioned data set.....	588
Delete a partitioned data set member.....	590
z/OS data set and member utilities.....	592
Access Method Services Interface.....	596
List the files and directories of a UNIX file path.....	598
Retrieve the contents of a z/OS UNIX file.....	603
Write data to a z/OS UNIX file.....	607
Create a UNIX file or directory.....	610
Delete a UNIX file or directory.....	613
z/OS UNIX file utilities.....	615
List z/OS UNIX Filesystems.....	621
Create a z/OS UNIX zFS filesystem.....	623
Delete z/OS UNIX zFS Filesystem.....	625
Mount a z/OS UNIX file system.....	626
Unmount a UNIX file system.....	629
JSON document specifications for z/OS data set and file REST interface requests.....	631
Error reporting categories.....	641
z/OS jobs REST interface.....	647
Obtain the status of a job.....	652
List the jobs for an owner, prefix, or job ID.....	655
List the spool files for a job.....	659
Retrieve the contents of a job spool file.....	661
Submit a job.....	665
Hold a job.....	669
Release a job.....	672
Change the job class.....	675
Cancel a job.....	677
Cancel a job and purge its output.....	680
JSON document specifications for z/OS jobs REST interface requests.....	682
Error reporting categories.....	688
z/OSMF information retrieval service.....	699
Retrieve z/OSMF information.....	700
z/OSMF settings services.....	702
Persist z/OSMF settings for a user or plug-in.....	704
Retrieve z/OSMF settings data.....	706
Delete z/OSMF settings data.....	707
z/OSMF system variable services.....	709
Create or update system variables.....	710
Get system variables.....	712
Import system variables.....	714
Export system variables.....	715
Delete system variables.....	717
z/OSMF authentication services.....	718
Log in to the z/OSMF server.....	719
Change the user password or passphrase.....	721
Log out of the z/OSMF server.....	724
z/OSMF workflow services.....	725

Create a workflow.....	728
Get the properties of a workflow.....	736
List the workflows for a system or sysplex.....	759
Start a workflow.....	761
Cancel a workflow.....	766
Delete a workflow.....	768
Retrieve a workflow definition.....	770
Archive a workflow instance.....	783
List the archived workflows for a system.....	785
Get the properties of an archived workflow.....	787
Delete an archived workflow.....	801
<b>Chapter 2. Creating workflow definitions for z/OS.....</b>	<b>803</b>
Terms you should know.....	804
The Workflows task schema.....	806
Creating the workflow definition file.....	806
Structure of a workflow definition file.....	807
Creating and viewing the workflow definition file.....	808
Sample XML files for your reference.....	809
References to external files.....	810
Defining entities for a workflow.....	811
Specifying the workflow root element.....	812
Specifying the workflow metadata.....	813
Including a manifest of translated text.....	816
Enabling a workflow definition file for future upgrades.....	817
Collecting user feedback.....	818
Defining steps for your workflow.....	820
Parent steps and leaf steps.....	821
Template steps.....	822
REST steps.....	835
Calling steps.....	840
Automated steps.....	846
Making a step conditional.....	850
runAsUser identity for a step.....	852
Using translatable strings.....	855
Using rich translatable strings.....	856
Defining variables for your workflow.....	856
Using Velocity templates for variable substitution and other functions.....	856
Specifying the variable element and its attributes.....	858
Sub-elements of the variable element.....	859
Using the element atCreate to qualify a variable definition .....	861
How to refer to a variable.....	863
Workflow internal variables .....	864
System variables.....	866
Array variables.....	869
Caller scope variables.....	870
Providing a workflow variable input file.....	871
Workflow XML reference.....	874
<b>Chapter 3. Creating your own z/OSMF plug-ins.....</b>	<b>923</b>
Developing web-based applications.....	924
Using the z/OSMF core JavaScript APIs.....	924
Using the Application Linking Manager JavaScript APIs.....	935
Logging client messages in the z/OSMF log.....	946
Retrieving files and resources for your application.....	951
Authoring end user assistance.....	952
Creating help plug-ins.....	953

Adding links to help plug-ins.....	962
Adding your applications to z/OSMF.....	963
Securing your applications.....	969
<b>Chapter 4. Preparing software to exploit cloud provisioning.....</b>	<b>971</b>
<b>Appendix A. Enabling tracing for the z/OS jobs REST interface.....</b>	<b>977</b>
<b>Appendix B. Creating product information files for the Software Management task.....</b>	<b>979</b>
<b>Appendix C. Understanding the Portable Software Instance descriptor file.....</b>	<b>983</b>
<b>Appendix D. Software Management workflow variables.....</b>	<b>991</b>
<b>Appendix E. Accessibility.....</b>	<b>999</b>
Accessibility features.....	999
Consult assistive technologies.....	999
Keyboard navigation of the user interface.....	999
Dotted decimal syntax diagrams.....	999
<b>Notices.....</b>	<b>1003</b>
Terms and conditions for product documentation.....	1004
IBM Online Privacy Statement.....	1005
Policy for unsupported hardware.....	1005
Minimum supported hardware.....	1005
Trademarks.....	1006
<b>Index.....</b>	<b>1007</b>



---

# Figures

1. Key components in the application linking process.....	7
2. Registering an event type: request content.....	23
3. Example: Registering an event type.....	24
4. Example: Returned results of a successful event registration.....	25
5. Example: Returned results of an unsuccessful event registration.....	25
6. Sample response from a successful list tasks request.....	29
7. Example: Handlers enabled for the event type.....	30
8. Example: Returned results of a successful list handlers request.....	30
9. Example: Returned results of a successful list handlers request.....	30
10. Process for routing requests and responses between application servers.....	32
11. Sample request to retrieve wrapped data from an application server.....	37
12. Sample response for retrieving wrapped data from an application server.....	37
13. Sample request to retrieve unwrapped data from an application server.....	37
14. Sample response for retrieving unwrapped data from an application server.....	38
15. Sample request to retrieve binary data from an application server.....	38
16. Sample response for retrieving binary data from an application server.....	39
17. Sample request to update data on an application server.....	41
18. Sample response for updating data on an application server.....	41
19. Sample request to delete data from an application server.....	44
20. Sample response for deleting data from an application server.....	44
21. Cloud Provisioning Summary.....	45
22. Sample request to obtain an IP address, with the request body.....	56
23. Sample request to release an IP address, with the request body.....	58

24. Sample request to obtain a port, with request body.....	62
25. Sample request to release a port, with the request body.....	65
26. Sample request to obtain a SNA application name, with request body.....	67
27. Sample request to release a SNA application name.....	69
28. Sample request to add a classification rule.....	72
29. Sample request to remove a classification rule.....	74
30. Response from a get data set attributes request.....	76
31. Sample request to get data set attributes.....	76
32. Sample request to get a domain.....	83
33. Sample request to get a domain history.....	86
34. Sample request to list domains.....	89
35. Sample request to create a tenant.....	94
36. Sample request to get a tenant.....	99
37. Sample request to get a tenant history.....	102
38. Sample request to list tenants.....	108
39. Sample request to delete a tenant.....	110
40. Sample request to assign CPU properties to a tenant, with the request body.....	112
41. Sample request to assign memory capping properties to a tenant, with the request body.....	114
42. Sample request to assign a solution ID, with the request body.....	115
43. Sample request to disable CPU capping.....	117
44. Sample request to disable memory capping.....	118
45. Sample request to disable metering.....	119
46. Sample request to enable CPU capping.....	121
47. Sample request to enable memory capping.....	122
48. Sample request to enable metering.....	124

49. Sample request to add consumers to a tenant, with the request body.....	125
50. Sample request to remove tenant consumer, with the request body.....	127
51. Sample request to add tenant description, with the request body.....	129
52. Sample request to add tenant groups, with the request body.....	130
53. Sample request to remove tenant groups, with the request body.....	132
54. Sample request to get a resource pool.....	137
55. Sample request to get a domain resource pool.....	144
56. Sample request to get a resource pool history.....	146
57. Sample request to list resource pools.....	152
58. Sample request to list domain resource pools.....	158
59. Sample request to update security state, with the request body.....	160
60. Sample request to get security profile information.....	162
61. Sample request to create a standard software services template.....	176
62. Sample response body.....	177
63. Sample request to create a composite software services template.....	177
64. Sample response body for a composite template.....	177
65. Sample request to create a new version of a software services template.....	184
66. Sample response body.....	185
67. Sample request to create a new verison of a composite software services template.....	185
68. Sample request to create a new software services template based on an existing one, with request body.....	188
69. Sample response body.....	188
70. Sample request to modify a software services template.....	193
71. Sample request to modify a composite software services template.....	194
72. Sample request to delete a software services template.....	195
73. Sample request to retrieve a standard software services template.....	208

74. Sample request to retrieve a composite software services template.....	209
75. Sample request to retrieve a software template history.....	212
76. Sample request to retrieve software services template documentation.....	214
77. Sample request to retrieve prompt variables.....	216
78. Response body for the GET prompt variables request.....	216
79. Sample request to retrieve prompt variables, composite template.....	217
80. Response body for the GET prompt variables request, composite template.....	217
81. Sample request to retrieve source information.....	219
82. Response body for the get source request.....	219
83. Sample request to retrieve source information for a composite template.....	219
84. Response body for the get source request for a composite template.....	220
85. Sample request to list software services templates.....	224
86. Sample request to publish a software services template.....	228
87. Sample request to test a standard software services template.....	231
88. Sample response body.....	231
89. Sample request to test a composite software services template.....	231
90. Sample response body.....	232
91. Sample request to refresh a software services template.....	233
92. Sample request to archive a software services template.....	234
93. Sample request to add an approval record for a software services template.....	236
94. Sample response body.....	236
95. Sample request to get an approval record for a software services template.....	238
96. Sample response body.....	238
97. Sample request to list the approval records for a software services template.....	240
98. Sample response body.....	241

99. Sample request to approve an approval record for a software services template.....	243
100. Sample request to perform batch approval for a software services template.....	244
101. Sample request to perform batch rejection of approvals for a software services template.....	244
102. Sample request to perform batch approval and rejection of approvals for a software services template.....	245
103. Sample request to reject an approval record for a software services template.....	246
104. Sample request to delete an approval record for a software services template.....	247
105. Sample request to indicate security is complete for a software services template.....	248
106. Sample request to run a software services template.....	254
107. Sample response body.....	255
108. Sample request to test a composite software services template.....	255
109. Sample response body.....	255
110. Sample request to get a standard software services template.....	267
111. Sample request to retrieve a published software template history.....	270
112. Sample request to get consumer documentation for a software services template.....	272
113. Sample request to get the prompt variables for a published software service template.....	275
114. Sample request to retrieve prompt variables, composite template.....	276
115. Response body for the GET prompt variables request, composite template.....	277
116. Sample request to list all published software service templates.....	282
117. Response body for the GET request.....	282
118. Sample request to run a software services template.....	284
119. Sample request to create a software services instance.....	299
120. Sample request for a composite.....	300
121. Sample request to get software services instance properties.....	310
122. Sample request to list software services instances.....	317
123. Sample response from a list software services instances request .....	318

124. Sample request to get software services instance variables.....	322
125. Sample response from a get software services instance variables request .....	322
126. Sample request to get software services instance variables in key-value format.....	324
127. Sample response from a get key-value variables request .....	324
128. Sample request to update a software services instance property.....	330
129. Sample request to update variables for a software services instance.....	332
130. Sample request to delete a software services instance.....	334
131. Sample request to resume a provisioning workflow for a software services instance.....	336
132. Sample request to perform an action against a software services instance variables.....	339
133. Sample response from a get software services instance variables request .....	339
134. Sample request to resume an action workflow for a software services instance.....	341
135. Sample request to retry a provisioning workflow for a software services instance.....	343
136. Sample request to retry an action workflow for a software services instance.....	345
137. Sample request to get software services instance actions.....	349
138. Sample response for performed actions .....	349
139. Sample request to list performed actions.....	353
140. Sample response from a list actions request .....	353
141. Sample request to delete a response for a performed action .....	355
142. Sample request to create SSINs.....	359
143. Sample request to list SSINs.....	361
144. Sample request to create a variable name.....	363
145. Sample request to create a variable name.....	365
146. Sample JSON structure for persisted data.....	368
147. Sample request to persist user-specific data.....	369
148. Sample response from a request to persist user-specific data.....	369

149. Sample JSON structure for persisted data.....	370
150. Sample request to retrieve persisted data.....	371
151. Sample response from a request to retrieve persisted data.....	371
152. Sample JSON structure for persisted data.....	372
153. Sample request to delete persisted data.....	373
154. Sample response from a request to delete persisted data.....	374
155. Example sysplex and system configuration.....	375
156. Sample request to retrieve data from one system.....	381
157. Sample response from a request to retrieve data from one system.....	381
158. Sample request to retrieve data from a list of systems.....	381
159. Sample response from a request to retrieve data from a list of systems.....	382
160. Sample request to retrieve data from all the systems in a group.....	382
161. Sample response from a request to retrieve data from all the systems in a group.....	383
162. Sample request to retrieve data from all the systems in a sysplex.....	384
163. Sample response from a request to retrieve data from all the systems in a sysplex.....	384
164. Sample request to retrieve data from all the systems in a CPC.....	385
165. Sample response from a request to retrieve data from all the systems in a CPC.....	385
166. Sample request to update data for one system.....	388
167. Sample response from a request to update data for one system.....	388
168. Sample request to update data for a list of systems.....	388
169. Sample response from a request to update data for a list of systems.....	389
170. Sample request to update data for all the systems in a group.....	389
171. Sample response from a request to update data for all the systems in a group.....	390
172. Sample request to update data for all the systems in a sysplex.....	390
173. Sample response from a request to update data for all the systems in a sysplex.....	391

174. Sample request to update data for all the systems in a CPC.....	391
175. Sample response from a request to update data for all the systems in a CPC.....	392
176. Sample request to delete data from one system.....	394
177. Sample response from a request to delete data from one system.....	394
178. Sample request to delete data from a list of systems.....	395
179. Sample response from a request to delete data from a list of systems.....	395
180. Sample request to delete data from all the systems in a group.....	395
181. Sample response from a request to delete data from all the systems in a group.....	396
182. Sample request to delete data from all the systems in a sysplex.....	396
183. Sample response from a request to delete data from all the systems in a sysplex.....	397
184. Sample request to delete data from all the systems in a CPC.....	397
185. Sample response from a request to delete data from all the systems in a CPC.....	398
186. Successful response when authenticating with a system.....	399
187. Response when the authentication request fails.....	399
188. Sample request to authenticate with a system.....	399
189. Successful response when authenticating with an HTTP proxy server.....	401
190. Sample response when the authentication request fails.....	401
191. Sample request to authenticate with an HTTP proxy server.....	401
192. List MVS subsystems.....	404
193. Sample request to retrieve a list of software instances.....	416
194. Sample response from a request to retrieve a list of software instances.....	416
195. Sample request to retrieve the properties of a software instance.....	422
196. Sample response from a request to retrieve the properties of a software instance.....	422
197. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server.....	424
198. Sample request to list the data sets included in a software instance.....	427

199. Sample response for a list data sets request.....	427
200. Sample request to obtain the status of a list data sets request.....	427
201. Sample get status response when the list data sets request is in progress.....	427
202. Sample get status response when the list data sets request is complete.....	428
203. Adding a software instance: request content.....	429
204. Sample request to add a software instance.....	434
205. Sample request.....	438
206. Sample response.....	438
207. Sample response.....	438
208. Sample response.....	439
209. Modifying a software instance: request content.....	441
210. Sample request to modify a software instance.....	446
211. Sample response for a modify software instance request.....	446
212. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server.....	447
213. Sample request to retrieve the product, feature, and FMID information for a software instance.....	450
214. Sample response for a retrieve product, feature, and FMID information request.....	451
215. Sample request to obtain the status of a retrieve product, feature, and FMID information request.....	451
216. Sample get status response when the retrieve product, feature, and FMID information request is in progress.....	451
217. Sample get status response when the retrieve product, feature, and FMID information request is complete.....	451
218. Sample request to delete a software instance.....	453
219. Sample response for a delete software instance request.....	453
220. Sample request to retrieve a list of portable software instances.....	455
221. Sample response from a request to retrieve a list of portable software Required authorizations....	455
222. Sample request to retrieve the properties of a portable software instance.....	459
223. Sample response from a request to retrieve the properties of a portable software instance.....	459

224. Adding a portable software instance: request content.....	460
225. Sample request to add a portable software instance.....	462
226. Response to sample request to add a portable software instance.....	463
227. Sample GET request to the status monitor URL.....	463
228. Response to sample GET request to the status monitor URL.....	463
229. Sample final request to the status monitor URL.....	463
230. Response to sample final request to the status monitor URL.....	463
231. Sample request to delete a portable software instance.....	465
232. Sample response for a delete portable software instance request.....	465
233. Sample request to retrieve a list of systems.....	469
234. Sample response from a request to retrieve a list of systems.....	469
235. Sample request to retrieve a list of groups.....	471
236. Sample response from a request to retrieve a list of groups.....	471
237. Sample request to retrieve a list of systems included in a group.....	473
238. Sample response from a request to retrieve a list of systems included in a group.....	474
239. Sample request to retrieve a list of sysplexes.....	475
240. Sample response from a request to retrieve a list of sysplexes.....	476
241. Sample request to retrieve a list of systems included in a sysplex.....	478
242. Sample response from a request to retrieve a list of systems included in a sysplex.....	478
243. Sample request to retrieve a list of systems included in a CPC.....	480
244. Sample response from a request to retrieve a list of systems included in a CPC.....	481
245. Sample request to create a new TSO/E address space on the local system.....	487
246. Sample response from create TSO/E address space request on the local system.....	488
247. Sample request to create a new TSO/E address space on a remote system.....	488
248. Sample response from create TSO/E address space request on a remote system.....	488

249. Sample request to reconnect to an existing TSO/E address space on the local system .....	489
250. Sample response from a reconnect to TSO/E address space request on the local system.....	489
251. Sample request to reconnect to an existing TSO/E address space on a remote system.....	489
252. Sample response from a request to reconnect to a TSO/E address space on a remote system.....	490
253. Starting an application: example of the request content.....	490
254. Sample request to start an application in a TSO/E address space.....	491
255. Sample response from a start an application in a TSO/E address space request.....	492
256. Sample request to send a message to a TSO/E address space.....	505
257. Sample response from send message to TSO/E address space request.....	506
258. Sample request to send a message to an application.....	507
259. Sample response from send message to an application request.....	507
260. Sample request to ping a TSO/E address space.....	508
261. Sample response from ping TSO/E address space request.....	508
262. Sample request to receive a message from a TSO/E address space.....	509
263. Sample response from receive message from a TSO/E address space request.....	510
264. Sample request to receive messages from an application.....	511
265. Sample response for request to receive messages from an application.....	511
266. Sample request to logoff a TSO/E address space.....	512
267. Sample response for logoff a TSO/E address space request.....	512
268. Sample request to cancel a TSO/E address space.....	513
269. Sample response for a cancel TSO/E address space request.....	513
270. Sample request to issue a prime WRP request.....	516
271. Sample response body.....	516
272. Sample request to issue a delete WLM resource pool request.....	517
273. Sample response body.....	517

274. Sample request to construct a service definition based on the current installed definition.....	519
275. Sample response body.....	519
276. Sample request to construct a service definition by removing the classification rule, then installing the new service definition.....	521
277. Sample response body.....	521
278. Sample request to get metered data.....	524
279. Sample response.....	524
280. Sample request to get DDS server time data.....	525
281. Sample response.....	526
282. Sample request to issue a command synchronously.....	543
283. Sample response body.....	543
284. Sample request to issue a command and detect a keyword.....	544
285. Sample response body.....	544
286. Sample request to issue a system command asynchronously.....	544
287. Sample response body.....	544
288. Sample request to issue an s PEGASUS command synchronously and detect the keyword PEGASUS.....	544
289. Sample response body.....	544
290. Sample request to issue an s PEGASUS command asynchronously and detect the keyword XIAOX.....	545
291. Sample response body.....	545
292. Sample request to get the response for a system command that was issued asynchronously.....	551
293. Sample response body for a get command response request.....	551
294. Sample request to get the detect result.....	557
295. Sample response body for a get detect result request.....	557
296. Sample request to get the detect result.....	557
297. Sample response body for a successful get detect result request.....	558

298. Sample request to get messages from a 1-hour duration.....	561
299. Sample response body for a Get Messages request.....	562
300. Sample request for a Get Messages request that starts at a specific timestamp.....	562
301. Sample response body for a Get Messages request.....	562
302. Example: list all of the data sets.....	571
303. Example: List all of the cataloged data sets with specified base attributes.....	572
304. Example: List all of the members of a data set.....	574
305. Example: List all of the members of a data set with specified base attributes.....	575
306. Example: Returned contents of the SMFPRM00 member of sys1.parmlib.....	579
307. Example: Returned contents of the SMFPRM00 member of sys1.parmlib.....	579
308. Example: Retrieve the contents of a sequential data set.....	580
309. Example: Request header for a write request to the SMFPRM00 member of sys1.parmlib.....	583
310. Example: Request body for a write request to the SMFPRM00 member of sys1.parmlib.....	584
311. Example: Contents of a sequential data set.....	585
312. Example: Contents of a member of data set with regular expression.....	585
313. Example: Create a data set .....	587
314. Example: Create data set.....	588
315. Example: Create PDSE data set.....	588
316. Example: Delete a data set .....	590
317. Example: Delete uncataloged data set.....	590
318. Example: Delete a member of a cataloged partitioned data set.....	592
319. Example: Delete a member of an uncataloged partitioned data set.....	592
320. 'rename' request.....	592
321. 'copy' request.....	592
322. 'higrate', 'hrecall', or 'hdelete' request.....	592

323. Example: Rename MY.OLD.DSN to MY.NEW.DSN.....	596
324. Example: copy member MYMEM1 from MY.OLD.DSN to MY.NEW.DSN(MYMEM2).....	596
325. IDCAMS Access Methods Services.....	598
326. Example: Returned list of UNIX files and directories in path /usr.....	602
327. Example: Returned list of UNIX files .....	602
328. Example: List UNIX files with more query parameters. ....	603
329. Example: Response body for a GET request to the UNIX file /etc/inetd.conf.....	607
330. Example: Request body for a PUT request to the UNIX file /etc/inetd.conf.....	610
331. Example: Create a UNIX file.....	612
332. Example: Create a UNIX directory.....	613
333. Example: Delete a UNIX file.....	614
334. Example: Delete a UNIX directory.....	615
335. Example: Rename a UNIX file.....	621
336. List UNIX Filesystems.....	623
337. Create UNIX Filesystems.....	625
338. Delete UNIX Filesystems.....	626
339. Example: Mount a UNIX file system.....	628
340. Example: Unmount a UNIX file system.....	630
341. Format of resource URLs for z/OS jobs REST interface.....	649
342. Example: Returned job status.....	655
343. Example: Returned status for an active step.....	655
344. Example: Returned list of the jobs for a specific owner and job name prefix.....	658
345. Example: Returned list of spool files.....	660
346. Example: Returned spool file content.....	664
347. Example: Returned spool file content (a range of records).....	664

348. Example: Returned job content (the job JCL).....	664
349. Example: Returned results of a job submission.....	669
350. Example: Returned results of a job hold request.....	672
351. Example: Returned results of a job release request.....	674
352. Example: Returned results of a job class change.....	677
353. Example: Returned results of a job cancellation.....	679
354. Example: Results of a job delete request.....	682
355. Sample request to retrieve z/OSMF information.....	702
356. Sample request to persist user-specific z/OSMF settings data.....	705
357. Sample request to retrieve z/OSMF settings data.....	707
358. Sample response from a request to retrieve z/OSMF settings data.....	707
359. Sample request to delete persisted z/OSMF settings data.....	708
360. Sample response from a request to retrieve z/OSMF settings data.....	708
361. Sample request to retrieve z/OSMF settings data.....	709
362. Sample response from a request to retrieve z/OSMF settings data.....	709
363. Sample request to create system variables.....	712
364. Sample request body for a create system variables request.....	712
365. Sample request to get system variables.....	713
366. Sample response from a get system variables request.....	713
367. Sample request to import system variables.....	715
368. Sample request body for an import system variables request.....	715
369. Sample request to export system variables.....	716
370. Sample request body for an export system variables request.....	717
371. Sample request to delete system variables.....	718
372. Sample request body for a delete system variables request.....	718

373. Log in and obtain a JSON Web Token for user authentication.....	721
374. Change a user password.....	724
375. Delete the JSON Web Tokens for this user.....	725
376. Sample request to create a workflow.....	735
377. Sample response from a create workflow request.....	736
378. Sample request to get workflow properties.....	755
379. Sample response from a get workflow properties request (Part 1 of 3).....	756
380. Sample response from a get workflow properties request (Part 2 of 3).....	757
381. Sample response from a get workflow properties request (Part 3 of 3).....	758
382. Sample request to list workflows.....	761
383. Sample response from a list workflows request.....	761
384. Sample request to start a workflow.....	766
385. Sample response from a start workflow request.....	766
386. Sample request to cancel a workflow.....	768
387. Sample response from a cancel workflow request.....	768
388. Sample request to delete a workflow.....	769
389. Sample response from a delete workflow request.....	770
390. Sample request to get a workflow definition.....	780
391. Sample response from a get workflow definition request (Part 1 of 3).....	780
392. Sample response from a get workflow definition request (Part 2 of 3).....	781
393. Sample response from a get workflow definition request (Part 3 of 3).....	782
394. Sample request to archive a workflow.....	784
395. Sample response from an archive workflow request.....	784
396. Sample request to list archived workflows.....	786
397. Sample response from a list archived workflows request.....	787

398. Sample request to get archived workflow properties.....	800
399. Sample response from a get archived workflow properties request.....	801
400. Sample request to delete an archived workflow.....	802
401. Sample response from a delete archived workflow request.....	802
402. General metadata for a workflow.....	814
403. You can define various questions for step owners to answer.....	819
404. How feedback questions can be included in the steps in your workflow.....	820
405. This sample step submits a REXX exec for immediate processing.....	831
406. Sample REST step definition with substitution variables and property mapping variables.....	839
407. Example: Defining a called workflow on the step element tag.....	846
408. You can designate a step as automated by adding the autoEnable element to the <step> element tag.....	848
409. Example of a suspend step.....	849
410. You can designate a step as conditional by adding the condition element to the <step> element tag.....	852
411. You can use variable values in the condition to be satisfied.....	852
412. Specifying attributes on the variable element.....	859
413. Variable definition in this example.....	862
414. How the atCreate element is used to specify variable attributes for required and prompt.....	863
415. Specifying the use of the simplified instance variable format in a workflow definition .....	863
416. You can use variable values in the condition to be satisfied.....	864
417. Example of how a system variable can be used in a workflow step.....	867
418. Example of the returned object from the Get system variables service.....	868
419. Variable substitution for a system variable in the Workflows task customize job statements window.....	868
420. Results of variable substitution for a system variable in the Workflows task.....	868
421. Example of how an array variable might be used in a workflow step.....	869

422. Example of how an array variable might be used in a workflow step.....	870
423. Format of a workflow variable input file.....	871
424. Example of a workflow variable input file.....	871
425. Using variable substitution in a workflow variable input file.....	872
426. How variable substitution in the workflow variable input file works with the Workflows task Step Perform wizard.....	873
427. Sample JavaScript code for importing and instantiating the global zosmfExternalTools object.....	925
428. Syntax to use for the isContentChanged function.....	926
429. Sample confirmation window for a close task request.....	927
430. Sample code for the isContentChanged function.....	928
431. Syntax to use for the shouldClose function.....	928
432. Syntax to use to call the programmaticallyCloseTab function.....	929
433. Sample code for the programmaticallyCloseTab function.....	930
434. Syntax to use for the cleanupBeforeDestroy function.....	930
435. Sample code for the cleanupBeforeDestroy function.....	931
436. Syntax to use to call the cleanupBeforeDestroyComplete function.....	931
437. Sample code for the cleanupBeforeDestroyComplete function.....	932
438. Syntax to use to call the getUserSessionId function.....	932
439. Sample code for the getUserSessionId function.....	932
440. Syntax to use to call the definePublicObject function.....	933
441. Sample code for the definePublicObject function.....	934
442. Syntax to use to call the retrievePublicObject function.....	934
443. Sample code for the retrievePublicObject function.....	934
444. Syntax to use to call the deletePublicObject function.....	935
445. Sample code for the deletePublicObject function.....	935
446. Sample code for importing and instantiating the Applinker API.....	936

447. Syntax to use to call the sendEvent function.....	938
448. Sample code for the sendEvent function.....	939
449. Syntax to use to call the getHandlers function.....	939
450. Syntax to use to call the hasLaunchContext function.....	940
451. Sample code for the hasLaunchContext function.....	941
452. Syntax to use to call the getEventFromUrl function.....	941
453. Sample code for the getEventFromUrl function.....	943
454. Syntax to use to call the subscribe function.....	944
455. Sample code for the subscribe function.....	944
456. Syntax to use to call the onLoadingComplete function.....	945
457. Sample code for creating a log prefix.....	946
458. Sample code for setting up client side logging.....	947
459. Sample z/OSMF client side log data.....	948
460. Sample request to retrieve a file.....	952
461. Sample response for a request to retrieve a file.....	952
462. Screen capture of the z/OSMF help system.....	952
463. Table of contents template for panel help plug-ins.....	959
464. Sample table of contents for the System Status task.....	959
465. Table of contents template for message help plug-ins.....	960
466. Sample table of contents for the messages issued by z/OSMF core.....	960
467. Sample code for linking to a help topic.....	963
468. Sample plug-in property file.....	968
469. Bootstrap properties for z/OSMF.....	978
470. Sample product information file for the Software Management task.....	981
471. Using the Add software instance REST API to specify a data set property in the request content...	996

472. Sample entry in the list for the subject data set.....	996
473. Sample template step to determine the specified property.....	997

---

# Tables

1. SAF identifiers for the z/OSMF REST interfaces.....	6
2. Operations provided through the Application Linking Manager interface services.....	8
3. Event types shipped with z/OSMF.....	11
4. Event requestors shipped with z/OSMF.....	15
5. Event handlers shipped with z/OSMF.....	20
6. Valid applId values for the z/OSMF plug-ins.....	26
7. Valid taskId values for the z/OSMF tasks.....	26
8. Operations provided through the application server routing services.....	33
9. Supported input parameters for the application server routing services.....	35
10. Supported input parameters for the application server routing services.....	40
11. Supported input parameters for the application server routing services.....	42
12. Resources for Cloud Provisioning.....	47
13. User roles for Cloud Provisioning.....	47
14. Objects for Cloud Provisioning.....	48
15. SAF resources for Cloud Provisioning Roles.....	48
16. z/OSMF resource pool services: operations summary.....	50
17. Request content for the obtain IP address request.....	51
18. Network parameters fields.....	52
19. Default values for recovery-method.....	53
20. Systems field.....	54
21. HTTP error response codes for an obtain IP address request.....	55
22. Response from an obtain IP address request.....	55
23. Request content for the release IP address request.....	57

24. Network parameters fields.....	57
25. Systems field.....	57
26. HTTP error response codes for a release IP address request.....	58
27. Request content for the obtain port request.....	59
28. Network parameters fields.....	60
29. Systems field.....	61
30. HTTP error response codes for an obtain port request.....	61
31. Response from an obtain port request.....	62
32. Request content for the release port request.....	63
33. Network parameters fields.....	64
34. Systems field.....	64
35. HTTP error response codes for a release port request.....	64
36. Request content for the obtain SNA application name request.....	66
37. Network parameters fields.....	66
38. HTTP error response codes for an obtain SNA application name request.....	67
39. Response from an obtain SNA application name request.....	67
40. Request content for the release SNA application name request.....	68
41. Network parameters fields.....	69
42. HTTP error response codes for a release SNA application name request.....	69
43. Request content for the add classification rule request.....	70
44. WLM parameters fields.....	71
45. HTTP error response codes for a release SNA application name request.....	71
46. Response from an add classification rule request.....	71
47. Request content for the remove classification rule request.....	73
48. WLM parameters fields.....	73

49. HTTP error response codes for a release SNA application name request.....	73
50. HTTP error response codes for a get data set attributes request.....	75
51. z/OSMF resource management services: operations summary.....	76
52. HTTP error response codes for a get domain request.....	79
53. Response from a get domain request.....	80
54. Response from a get request: Systems.....	82
55. Response from a create request: SAF-resource object.....	83
56. HTTP error response codes for a get domain history request.....	85
57. Response from a get request: History object.....	85
58. HTTP error response codes for a list domains request.....	87
59. Response from a list domains request.....	87
60. Properties of domains.....	87
61. Request content for the create tenant request.....	91
62. Tenant metering and capping properties object.....	92
63. Tenant managed by request object.....	93
64. Systems request object.....	93
65. HTTP error response codes for a get domain request.....	93
66. Response from a create tenant request.....	94
67. HTTP error response codes for a get tenant request.....	95
68. Response from a get tenant request.....	95
69. Response from a get request: tenant-templates.....	99
70. Response from a create request: SAF-resource object.....	99
71. HTTP error response codes for a get tenant history request.....	101
72. Response from a get request: History object.....	102
73. HTTP error response codes for a list tenants request.....	103

74. Response from a list tenants request.....	104
75. Tenant list.....	104
76. HTTP error response codes for a delete tenant request.....	110
77. Request content for the assign CPU properties to a tenant request.....	111
78. HTTP error response codes for an assign CPU properties to a tenant request.....	112
79. Request content for the assign memory capping properties to a tenant request.....	113
80. HTTP error response codes for an assign CPU properties to a tenant request.....	113
81. Request content for the assign solution ID request.....	114
82. HTTP error response codes for an assign solution ID request.....	115
83. HTTP error response codes for a disable CPU capping request.....	116
84. HTTP error response codes for a disable memory capping request.....	118
85. HTTP error response codes for a disable metering request.....	119
86. HTTP error response codes for an enable CPU capping request.....	120
87. HTTP error response codes for an enable memory capping request.....	122
88. HTTP error response codes for an enable metering request.....	123
89. Request content for the add tenant consumer request.....	125
90. HTTP error response codes for an add tenant consumer request.....	125
91. Request content for the remove tenant consumer request.....	126
92. HTTP error response codes for a remove tenant consumer request.....	126
93. Request content for the add tenant description request.....	128
94. HTTP error response codes for an add tenant description request.....	128
95. Request content for the add tenant groups request.....	129
96. HTTP error response codes for an add tenant groups request.....	130
97. Request content for the remove tenant groups request.....	131
98. HTTP error response codes for a remove tenant groups request.....	131

99. HTTP error response codes for a get resource pool request.....	133
100. Response from a get resource pool request.....	133
101. Response from a get request: Systems.....	137
102. HTTP error response codes for a get domain resource pool request.....	139
103. Response from a get domain resource pool request.....	139
104. Response from a get request: Systems.....	143
105. HTTP error response codes for a get resource pool request.....	145
106. Response from a get request: History object.....	146
107. HTTP error response codes for a get resource pool request.....	147
108. Response from a list resource pool request.....	148
109. Resource pool list.....	148
110. HTTP error response codes for a get domain resource pool request.....	153
111. Response from a list domain resource pool request.....	153
112. Domain resource pool list.....	154
113. Request content for the update security state request.....	159
114. HTTP error response codes for an update security state request.....	159
115. HTTP error response codes for a get security resources request.....	161
116. Response from a get security resources request.....	161
117. SecurityValidationItem structure.....	161
118. z/OSMF software services template services: operations summary.....	163
119. Response from a software services template request failure.....	168
120. Request content for a request to create a software services template.....	170
121. Composite-definition structure.....	174
122. Connector object.....	174
123. Response from a create software services template request.....	175

124. Response from a create request: SAF-resource object.....	175
125. Response from a software services template request failure.....	176
126. Request content for the software services template request.....	179
127. Composite-definition structure.....	182
128. Connector object.....	183
129. Response from a create new version of a software services template request.....	183
130. Response from a create request: SAF-resource object.....	184
131. Request content for the software services template request.....	186
132. Response from a successful request.....	187
133. Response from a create request: SAF-resource object.....	187
134. Request content for a request to modify a software services template.....	189
135. Composite-definition structure.....	192
136. Connector object.....	193
137. Response from the software services template request.....	195
138. Composite-affected object.....	195
139. Response from a request to get a software services template.....	196
140. Response from a get request: Approval-Object.....	202
141. Response from a get request: Action-Object.....	203
142. Response from a get request: Variable-Object.....	204
143. Response from a get request: Prompt-Variable-Object.....	204
144. Response from a create request: SAF-resource object.....	205
145. Response from a get request: RunAsUser object.....	205
146. Response from a get request: Composite-definition object.....	206
147. Connector object.....	207
148. Response from a get request: Security Workflow Information.....	208

149. Response from a get request: History object.....	212
150. Response from a get prompt variables request.....	215
151. Response from a get request: Prompt-Variable-Object .....	215
152. Response from a get request: Composite-Prompt-Variable-Object.....	216
153. Response from a get source request.....	218
154. Response from a get request: Source-Info-Object.....	219
155. Array of objects.....	221
156. Fields for each software services template.....	221
157. Contents of composite-definition object.....	224
158. Request content for the software services template request.....	227
159. Response from the software services template request.....	228
160. Composite-affected object.....	228
161. Request content for the test software services template request.....	229
162. Runtime properties .....	230
163. Response from a test software services template request.....	230
164. Composite child registry information objects.....	231
165. Response from the software services template request.....	234
166. Composite-affected object.....	234
167. Request content for the add approval request.....	235
168. Response from an add approval request.....	236
169. Response from a get approval request.....	237
170. Response from a list approvals request.....	239
171. Response from a get approval request.....	240
172. Request content for the software services template request.....	242
173. Request content for the software services template request.....	243

174. Approval objects.....	244
175. Request content for the software services template request.....	245
176. z/OSMF published software service template services: operations summary.....	250
177. Response from a software services template request failure.....	251
178. Request content for the run software services template request.....	252
179. Runtime properties .....	253
180. Response from a run software services template request.....	254
181. Composite child registry information objects.....	254
182. Response from a get software services template request.....	258
183. Response from a get request: Prompt-Variable Object.....	262
184. Response from a get request: Approval-Object.....	263
185. Response from a get request: Action-Object.....	264
186. Response from a get request: Variable-Object.....	264
187. Response from a create request: SAF-resource object.....	265
188. Response from a get request: composite-definition object.....	265
189. Connector object.....	266
190. Response from a get request: RunAsUser object.....	267
191. Response from a get request: History object.....	270
192. Response from a get prompt variables request.....	274
193. Response from a get request: Prompt-Variable Object.....	274
194. Response from a get request: Composite-Prompt-Variable Object.....	275
195. Array of objects.....	279
196. Fields for each software services template.....	279
197. Contents of composite-definition object.....	281
198. Response from a software services template request failure.....	282

199. Request content for the modify software services template request.....	283
200. z/OSMF software services instance services: operations summary.....	286
201. Response from a request failure.....	287
202. Request content for a create software services instance request.....	289
203. Action structure for a "create a software services instance" request.....	293
204. Variable structure.....	294
205. Response from a get request: Prompt-Variable-Object.....	294
206. Composite-data structure.....	295
207. Connector object.....	296
208. HTTP error response codes for a create software services instance request.....	297
209. Response from a create software services instance request.....	297
210. Response from a request failure.....	297
211. HTTP error response codes for a get software services instance contents request.....	301
212. JSON object that is returned for a get software services instance property request.....	302
213. Action structure for a "get the contents of a software services instance" request .....	306
214. Variable structure.....	308
215. Response from a get request: Prompt-Variable-Object.....	308
216. Composite-instance data.....	309
217. Composite child object.....	309
218. Response from a request failure.....	310
219. HTTP error response codes for a list software services instances request.....	313
220. JSON object that is returned for a list software services instances request.....	314
221. Composite child object.....	316
222. Response from a request failure.....	316
223. HTTP error response codes for a get software services instance variables request.....	320

224. Get variables request: Format of the variables object.....	321
225. Variable structure.....	321
226. Response from a request failure.....	321
227. HTTP error response codes for a get software services instance key-value variables request.....	323
228. Get key-value variables request: Format of the variables object.....	324
229. Response from a request failure.....	324
230. Request content for an update software services instance request.....	325
231. Action structure for an "update software services instance" request.....	327
232. Variable structure.....	328
233. Response from a get request: Prompt-Variable-Object.....	328
234. HTTP error response codes for a update software services instance request.....	330
235. Response from a request failure.....	330
236. Request content for the update software services instance variables request.....	331
237. Variable structure.....	331
238. HTTP error response codes for a update software services instance request.....	332
239. Response from a request failure.....	332
240. HTTP error response codes for a delete software services instance request.....	333
241. Response from a request failure.....	334
242. HTTP error response codes for a resume provisioning workflow request.....	335
243. Response from a request failure.....	336
244. Request content for the perform action software services instance request.....	337
245. Input variable structure.....	337
246. HTTP error response codes for a do action request.....	338
247. Response body for the do action request.....	338
248. Response from a request failure.....	338

249. HTTP error response codes for a resume action workflow request.....	340
250. Response from a request failure.....	341
251. HTTP error response codes for a retry provisioning workflow request.....	342
252. Response from a request failure.....	343
253. HTTP error response codes for a retry action workflow request.....	344
254. Response from a request failure.....	345
255. HTTP error response codes for a get software services instance contents request.....	346
256. JSON object that is returned for a get actions request.....	347
257. Response from a request failure.....	348
258. HTTP error response codes for a "get software services instance contents" request.....	352
259. JSON object that is returned for a list actions request.....	352
260. Action object for a list actions request.....	352
261. Response from a request failure.....	353
262. HTTP error response codes for a delete action response request.....	354
263. Response from a request failure.....	355
264. SSIN services: operations summary.....	356
265. Response from a request failure.....	356
266. Request content for the create SSIN request.....	358
267. HTTP error response codes for a create SSIN request.....	358
268. Response from a create SSIN request.....	358
269. Fields in the ssin-list array.....	358
270. Response from a request failure.....	358
271. HTTP error response codes for a list SSIN request.....	360
272. Response from a list SSINs request.....	361
273. Fields in the ssin-list array.....	361

274. Response from a request failure.....	361
275. Request content for the create variable name request.....	362
276. HTTP error response codes for a create variable request.....	363
277. Response from a create variable name request.....	363
278. Response from a request failure.....	363
279. Request content for the create unique variable names request.....	364
280. HTTP error response codes for a create variable request.....	364
281. Response from a create unique variable names request.....	365
282. Response from a request failure.....	365
283. Operations provided through the data persistence services.....	366
284. Operations provided through the multisystem routing services.....	376
285. Supported input parameters for the multisystem routing services.....	379
286. Supported input parameters for the multisystem routing services.....	386
287. Supported input parameters for the multisystem routing services.....	393
288. Subsystem services method.....	401
289. Response.....	403
290. Notification methods.....	404
291. Response content from the notifications received by the current user.....	405
292. Request content for the send notification request.....	407
293. Response content for the send notification request.....	408
294. Request content from a notification that requires user input.....	410
295. Response content from a notification that requires user input.....	410
296. Operations provided through the software management services.....	412
297. Operations provided through the topology services.....	465
298. Operations provided through the TSO/E address space services.....	481

299. Supported parameters for the start and reconnect TSO/E session requests.....	486
300. Supported parameters for the TSO/E command with z/OSMF REST API.....	493
301. Response content for a successful issue command request.....	495
302. Response JSON object for an unsuccessful issue command request.....	496
303. Response codes for unsuccessful issue command requests.....	496
304. ....	501
305. Response content for a successful get response request.....	502
306. Response JSON object for an unsuccessful issue command request.....	503
307. Operations provided through the WLM resource pooling services.....	513
308. Request content for the prime WLM resource pool request.....	515
309. Response content for a successful prime WLM resource pool request.....	515
310. Response content for a delete WLM resource pool request.....	517
311. Request content for the construct a WLM service definition request.....	518
312. Response content for a successful construct a WLM service definition request.....	518
313. Request content for the construct a WLM service definition request.....	520
314. Response content for a successful construct a WLM service definition request.....	520
315. Operations provided through the RMF services.....	521
316. Query parameters.....	522
317. HTTP error response codes for a get metered data request.....	523
318. HTTP error response codes for a get DDS server time data request.....	525
319. Operations provided through the z/OS console services.....	526
320. Request content for the issue command request.....	529
321. HTTP error response codes for an issue command request.....	533
322. Response content for a successful synchronous issue command request.....	540
323. Response content for a successful asynchronous issue command request.....	541

324. Response content for an unsuccessful issue command request.....	543
325. HTTP error response codes for a get command response request.....	546
326. Response content for a successful get command response request.....	550
327. Response content for an unsuccessful get command response request.....	551
328. HTTP error response codes for a detect result for unsolicited messages request.....	552
329. Response content for a successful get detect result request.....	557
330. Response content for an unsuccessful get detect result request.....	557
331. Query parameters for a Get Messages request.....	558
332. HTTP Error response codes for a Get Messages request.....	560
333. Response content for a successful Get Messages request.....	560
334. Messages JSON object.....	561
335. Response content for an unsuccessful Get Messages request.....	561
336. Operations provided through the z/OS data set and file REST interface services.....	563
337. Request body to create a sequential or partitioned data set .....	586
338. Request.....	594
339. Input Content.....	597
340. Response.....	597
341. Filter parameters for a "List UNIX Files and Directories" request.....	599
342. Tree traversal parameters for a "List UNIX Files and Directories" request.....	601
343. Request body to create a UNIX file or directory .....	611
344. JSON request document.....	616
345. Input Content.....	624
346. Request body to mount a UNIX file system.....	627
347. Request body to unmount a UNIX file system .....	629
348. Contents of the JSON data set list document.....	631

349. X-IBM-Attributes=vol.....	632
350. X-IBM-Attributes=dsname.....	633
351. X-IBM-Attributes=base.....	633
352. Items key:value pairs.....	633
353. Contents of the JSON PDS/PDSE member list document.....	634
354. X-IBM-Attributes=base and data set RECFM=F or V.....	635
355. Items key:value pairs for data set RECFM=F or V.....	635
356. X-IBM-Attributes=base and data set RECFM=U.....	636
357. Items key:value pairs for data set RECFM=U.....	636
358. Contents of the create a sequential and partitioned data set document.....	637
359. Contents of the JSON file list document.....	638
360. Contents of the unix file and directory.....	638
361. Unix items key:value pairs.....	639
362. Contents of the Mount and unmount a file system document.....	639
363. Contents of the create a UNIX file document.....	640
364. Contents of the JSON error report document.....	641
365. Error categories for z/OS data set and file REST interface operations.....	641
366. Category 1 errors for z/OS data set and file REST interface operations.....	642
367. Category 2 errors for z/OS data set and file REST interface operations.....	643
368. Category 3 errors for z/OS data set and file REST interface operations.....	645
369. Category 4 errors for z/OS data set and file REST interface operations.....	645
370. Category 5 errors for z/OS data set and file REST interface operations.....	645
371. Category 6 errors for z/OS data set and file REST interface operations.....	646
372. Category 7 JSON parser conditions.....	647
373. Category 8 z/OS XL C/C++ Conditions.....	647

374. Category 9 errors for z/OS data set and file REST interface operations.....	647
375. Operations provided through the z/OS jobs REST interface services.....	648
376. Job modify operations provided through the z/OS jobs REST interface services.....	650
377. JESJOBS class authorizations needed for performing job modify operations.....	651
378. Variations of a PUT request for submitting a job from a data set or UNIX file.....	668
379. Contents of the JSON job document.....	682
380. Contents of the JSON job completed document.....	684
381. Contents of the JSON job feedback document.....	685
382. Contents of the JSON job file document.....	686
383. Contents of the JSON job step data document.....	687
384. Contents of the JSON error report document.....	688
385. Error categories for z/OS jobs REST interface operations.....	688
386. Category 1 errors.....	690
387. Category 3 errors.....	691
388. Category 4 errors.....	691
389. Category 5 errors.....	691
390. Category 6 errors.....	692
391. Category 7 errors.....	697
392. Category 8 errors.....	697
393. Category 9 errors.....	698
394. Category 10 errors.....	698
395. Operations provided through the z/OSMF settings services.....	703
396. z/OSMF system variable services: operations summary.....	709
397. z/OSMF system variable services: URI path variables.....	709
398. Request content for the create or update system variables request.....	711

399. Fields in a JSON object for the create or update system variables request.....	711
400. HTTP error response codes for a create or update system variables request.....	711
401. HTTP error response codes for a get system variables request.....	713
402. Get system variables request response body.....	713
403. Get system variables request: objects.....	713
404. Request content for the import system variables request.....	714
405. HTTP error response codes for a create system variables request.....	714
406. Request content for the export system variables request.....	716
407. HTTP error response codes for a create system variables request.....	716
408. HTTP error response codes for a create system variables request.....	717
409. z/OSMF authentication services method.....	718
410. Response body for a "log in to the z/OSMF server" request.....	720
411. HTTP error response codes for a "log in to the z/OSMF server" request.....	720
412. Request content for a change user password or passphrase request.....	722
413. Response body for a change user password or passphrase request.....	722
414. HTTP error response codes for a change user password or passphrase request.....	723
415. z/OSMF workflow services: operations summary.....	725
416. z/OSMF workflow services: URI path variables.....	726
417. Error response body elements for the z/OSMF workflow services API.....	727
418. Request content for the create workflow request.....	729
419. HTTP error response codes for a create workflow request.....	734
420. Response from a create workflow request.....	735
421. HTTP error response codes for a get workflow properties request.....	738
422. JSON object that is returned to a get workflow properties request.....	739
423. Get Workflow Properties request: Format of the automation-info object.....	743

424. Get Workflow Properties request: Format of the step-info object.....	745
425. Get Workflow Properties request: Format of the jobInfo object.....	753
426. Get Workflow Properties request: Format of the jobstatus object.....	753
427. Get Workflow Properties request: Format of the jobfiles object.....	754
428. Get Workflow Properties request: Format of the variable-reference object.....	755
429. Get Workflow Properties request: Format of the variable-info object.....	755
430. List workflows request: Format of the workflow-info object.....	760
431. Request content for the start workflow request.....	763
432. HTTP error response codes for a start workflow request.....	764
433. Structure of the JSON object that is returned to the notification URL.....	765
434. HTTP error response codes for a cancel workflow properties request.....	767
435. HTTP error response codes for a delete workflow request.....	769
436. HTTP error response codes for a retrieve a workflow definition request.....	771
437. JSON object that is returned to a retrieve a workflow definition request.....	771
438. Retrieve a workflow definition request: Fields included in every step-definition object.....	773
439. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a calling step (a step that calls another workflow).....	773
440. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step.....	774
441. Retrieve a workflow definition request: Format of the variable-specification-info object.....	778
442. Retrieve a workflow definition request: Format of the variable-definition object.....	779
443. HTTP error response codes for an archive workflow request.....	784
444. Response from an archive workflow request.....	784
445. HTTP error response codes for a get archived workflow properties request.....	786
446. List archived workflows request: Format of the workflow-info object.....	786
447. HTTP error response codes for a get archived workflow properties request.....	788
448. JSON object that is returned to a get archived workflow properties request.....	788

449. Get Archived Workflow Properties request: Format of the automation-info object.....	792
450. Get Archived Workflow Properties request: Format of the step-info object.....	793
451. Get Archived Workflow Properties request: Format of the variable-reference object.....	800
452. Get Archived Workflow Properties request: Format of the variable-info object.....	800
453. HTTP error response codes for a delete archived workflow request.....	802
454. Summary of template step elements.....	823
455. Elements for saving a generated job.....	827
456. Summary of REST step elements.....	835
457. Use scope and isCallable to coordinate workflow-to-workflow actions.....	841
458. Major step elements for defining a called workflow in your workflow definition file.....	842
459. Information to specify for notifying a user about a suspended step.....	849
460. How a runAsUser ID is used in workflow processing.....	853
461. Basic usage scenarios: How the static or dynamic determination is made for the runAsUser value	855
462. Translatable strings.....	855
463. Internal variables: Variables that are provided with z/OSMF.....	865
464. Reference tables for the Workflows XML schema.....	874
465. Workflow metadata elements .....	876
466. Sub-elements for a configuration type workflow.....	880
467. Sub-elements for a provisioning type workflow.....	882
468. Workflow upgrade elements .....	884
469. Manifest elements summary table.....	886
470. Step elements summary table: Elements to use for defining all steps.....	888
471. Step elements summary table: Additional sub-element for parent steps.....	890
472. Step elements summary table: Additional sub-elements for leaf steps.....	892
473. Step elements summary table: Additional sub-elements for REST steps.....	906

474. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow).....	910
475. Variable definition elements summary.....	914
476. Variable definition type-specific sub-elements summary.....	917
477. atCreate element summary.....	921
478. Functions provided in the zosmfExternalTools API.....	925
479. Functions you can add to the zosmfExternalTools API.....	926
480. Functions provided in the AppLinker API.....	936
481. Expected results by launching option and task state.....	942
482. Methods provided for the client side logger.....	947
483. URL to use for each z/OSMF category.....	961

## About this document

---

This document is intended to help you write programs that use the services and facilities of IBM® z/OS® Management Facility (z/OSMF).

The programming interfaces in z/OSMF include the following.

- Representational State Transfer (REST) services for working with z/OS and z/OSMF.
- XML schema for creating a workflow definition for performing activities on a z/OS system, such as configuring a component or product.
- Services for creating plug-ins that add installation-specific function to z/OSMF.

## Who should use this document

---

This information is intended for the programmer responsible for writing programs that use the z/OSMF infrastructure. Such activities, include, for example:

- Using an application programming interface (API) as a client to obtain information about a batch job on z/OS
- Developing a JavaScript program that includes a graphical user interface (GUI).

This document assumes that you are familiar with the z/OS operating system and its accompanying products.

## Where to find more information

---

For an overview of the information associated with z/OS, see *z/OS Information Roadmap*.

### **z/OSMF home page**

Visit the z/OSMF home page at <http://www.ibm.com/systems/z/os/zos/zosmf/>.

### **The z/OS Basic Skills Information Center**

The z/OS Basic Skills Information Center is a web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your web browser to the following web site, which is available to all users (no login required): <http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp>.



# How to send your comments to IBM

---

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

**Important:** If your comment regards a technical question or problem, see instead [“If you have a technical problem”](#) on page li.

Submit your feedback by using the appropriate method for your type of comment or question:

## **Feedback on z/OS function**

If your comment or question is about z/OS itself, submit a request through the [IBM RFE Community](#) ([www.ibm.com/developerworks/rfe/](http://www.ibm.com/developerworks/rfe/)).

## **Feedback on IBM Documentation function**

If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at [ibmdocs@us.ibm.com](mailto:ibmdocs@us.ibm.com).

## **Feedback on the z/OS product documentation and content**

If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to [mhvrcfs@us.ibm.com](mailto:mhvrcfs@us.ibm.com). We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: IBM z/OSMF Programming Guide, SC27-8420-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

# If you have a technical problem

---

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the [IBM Support Portal](#) ([support.ibm.com](http://support.ibm.com)).
- Contact your IBM service representative.
- Call IBM technical support.



# Summary of changes

---

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

## Summary of changes for z/OSMF Programming Guide for Version 2 Release 4 (V2R4)

---

The following content is new, changed, or no longer included in V2R4. The most recent updates are listed at the beginning of each section.

### New

The following content is new.

#### June 2021 refresh

- The following z/OS console service was added: [“Get messages from logs” on page 558](#).
- The following TSO/E address space services were added:
  - [“Get the response to a command that was issued with the TSO/E REST API” on page 501](#)
  - [“Issue a TSO/E command with z/OSMF REST API” on page 492](#)
- Usage notes were added to [“TSO/E address space services” on page 481](#).
- For the service described in [“Start a workflow” on page 761](#), the JSON object that is returned to the notification URL now includes the user ID of the user who stopped automation.

#### May 2021 refresh

- New Software Management workflow variables are added. For more information, see [Appendix D, “Software Management workflow variables,” on page 991](#).

#### February 2021 refresh

The z/OS jobs REST interface services are enhanced, as follows:

- Customer header X-IBM-Target-System-User is new. This header is used to supply a z/OS user ID for accessing the target system.
- Customer header X-IBM-Target-System-Password is new. This header is used to supply the password that is associated with the z/OS user ID specified on the X-IBM-Target-System-User header.

For more information, see [“z/OS data set and file REST interface” on page 563](#) and [“z/OS jobs REST interface” on page 647](#).

#### December 2020 refresh

New function is available for IBM Cloud Provisioning and Management when you install the PTF for APAR PH29813.

The Cloud Provisioning REST services are enhanced in the following ways:

- A resource pool defines the scope of shared z/OS resources within a cloud domain that has multiple tenants. In this release, the concept of a shared resource pool is expanded to include sharing resources across an entire domain. Here, the resource pool is referred to as a *domain-shared resource pool*. Previously, you were limited to sharing a resource pool within a single tenant. By allowing multiple tenants within a domain to share a resource pool, you simplify the management of resources in a cloud provisioning environment. In contrast, if resource isolation is wanted for a template, it is recommended that you define a tenant-specific shared resource pool or a dedicated resource pool. The Resource Management services are enhanced with new properties to create, view, and modify domain-shared

resource pools. The Software Services services are updated to allow a domain-shared resource pool to be associated with a template. The names of shared resource pools end with two wildcard qualifiers (\*.\*).

The following REST services are new in this release:

- [“Get a domain history” on page 84](#)
- [“Get a tenant history” on page 100](#)
- [“Get a domain resource pool” on page 138](#)
- [“Get a resource pool history” on page 145](#)
- [“List domain resource pools” on page 152](#)
- [“Get a software services template history” on page 211](#)
- [“Get a published software service template history” on page 269](#)
- [“Modify a published software service template” on page 282](#)
- The following Cloud Provisioning REST services have new or enhanced functions in this release:
  - [“Get a domain” on page 79](#)
  - [“List the domains” on page 86](#)
  - [“Get a resource pool” on page 132](#)
  - [“Create a software services template” on page 170](#)
  - [“Create a new version of a software services template ” on page 178](#)
  - [“Modify a software services template” on page 188](#)
  - [“Get a software services template” on page 195](#)
  - [“Test a software services template” on page 228](#)
  - [“Run a published software service template” on page 252](#)
  - [“Get a published software service template” on page 257](#)
  - [“Create a software services instance” on page 288](#)
  - [“Get the contents of a software services instance” on page 301](#)
  - [“List the software services instances” on page 312](#)
  - [“Update a software services instance” on page 325](#)
  - [“Get the response for an action performed against a software services instance” on page 346](#)
  - [“List the responses for actions performed against a software services instance” on page 351](#)
- When you create a template, you can specify new options to do the following:
  - Delete instances automatically when they are deprovisioned. Previously, it was always necessary to delete a deprovisioned instance manually.
  - Archive provisioning workflows automatically when they complete. Previously, you were limited to selecting to either keep or delete a workflow. Archiving is the new default behavior for the workflows disposition property.
- The cloud provisioning actions schema file is updated to support a description element for each action. With the description element, you can optionally provide a text description of the action and the function it performs. For more information, see [Chapter 4, “Preparing software to exploit cloud provisioning,” on page 971](#).
- The actions editor is enhanced to work with variables of all types. Previously, the editor was limited to working with string variables only.
- It is now possible to provide an optional description for the provisioning actions that you define. A new "description" field is returned in the action structure for the following REST services:
  - [“Create a software services instance” on page 288](#)
  - [“Update a software services instance” on page 325](#)

Common HTTP request header **X-IBM-Session-Limit-Wait** is added to the z/OS data set and file REST interface services. You can include this header to extend the amount of time that a request can take to find an available CEA TSO address space. For more information, see [“z/OS data set and file REST interface” on page 563](#).

#### Prior to the December refresh

- APAR PH24527 adds new REST APIs, which can be used to enable and disable z/OSMF plug-ins and services. For more information, see [“z/OSMF settings services” on page 702](#).
- In support of the new ability to manipulate portable software instance objects, the following REST APIs are added to software management services:
  - [“List the portable software instances defined to z/OSMF” on page 453](#)
  - [“Retrieve the properties of a portable software instance” on page 455](#)
  - [“Add a new portable software instance” on page 460](#)
  - [“Delete a portable software instance” on page 464](#)
- The following topic is new: [“System variables” on page 866](#).
- The z/OS console services REST interface now includes properties that you can use to control console attributes, such as 'auth', 'routcode', 'mscope', 'storage', and 'auto'. For more information, see [“Issue a command from a system console” on page 528](#).
- In Cloud Provisioning, you can now provision composite templates and clustered composite templates in a multiple sysplex domain. Previously, this capability was available for standard templates only.
- The z/OS jobs REST interface services are enhanced, as follows:
  - Custom header **X-IBM-Target-System** is added to the z/OS jobs REST interface services. You can include this header to direct a REST request to a remote system in your enterprise. For more information, see [“z/OS jobs REST interface” on page 647](#).
  - Custom header **X-IBM-Intrdr-File-Encoding** is new. You can include this header to specify the appropriate code page for data that is written to the internal reader. For more information, see [“Submit a job” on page 665](#).
  - With the addition of the exec -data query parameter, the following z/OS jobs REST interface services can optionally return execution data about jobs on z/OS:
    - [“Obtain the status of a job” on page 652](#)
    - [“List the jobs for an owner, prefix, or job ID” on page 655](#)
  - With the addition of the status query parameter, the following GET request can be limited to active jobs only: [“List the jobs for an owner, prefix, or job ID” on page 655](#).
  - With the addition of the **fileEncoding** query parameter, the following GET request can specify an alternative code page for the spool file on z/OS: [“Retrieve the contents of a job spool file” on page 661](#).
- The z/OS data set and file REST interface services are enhanced, as follows:
  - The following options are added to the customer header X-IBM-Data-Type:
    - Value `"text;fileEncoding=<codepage>"` can be used to select an alternative EBCDIC code page. The default code page is IBM-1047.
  - Value `text;CrLf=true` can be used to control whether each input text line is ended with a carriage return line feed (CRLF), rather than a line feed (LF), which is the default.
  - Customer header Content-Encoding is new. This header is used to compress the response data. If present, its value indicates which encoding method can be used to decompress the media-type that was specified in the Accept-Encoding header. The supported file format is gzip.

For more information, see [“z/OS data set and file REST interface” on page 563](#).

- In Cloud Provisioning, you can define a multiple sysplex domain, with which you can provision middleware across multiple sysplexes in your enterprise. In this configuration, creating and modifying objects is done from a primary z/OSMF system, from which you can provision templates on other,

secondary z/OSMF systems. This enhancement allows your cloud provisioning environment to scale beyond the scope of a single sysplex.

In support of multiple sysplex domains, the following Cloud Provisioning services REST APIs are added:

- [“Create a tenant” on page 91](#)
- [“Delete a tenant” on page 109](#)
- [“Assign CPU properties to a tenant” on page 110](#)
- [“Assign memory capping properties to a tenant” on page 112](#)
- [“Assign a solution ID” on page 114](#)
- [“Disable CPU capping” on page 115](#)
- [“Disable memory capping” on page 117](#)
- [“Disable metering” on page 118](#)
- [“Enable CPU capping” on page 120](#)
- [“Enable memory capping” on page 121](#)
- [“Enable metering” on page 123](#)
- [“Add tenant consumer” on page 124](#)
- [“Remove tenant consumer” on page 126](#)
- [“Add tenant description” on page 127](#)
- [“Add tenant groups” on page 129](#)
- [“Remove tenant groups” on page 130](#)
- [“Update the security state for a tenant” on page 159](#)
- [“Get data set attributes” on page 74](#)
- New REST APIs that can be used to obtain and delete web tokens for authentication with an application server. For more information, see [“z/OSMF authentication services” on page 718](#).
- The z/OSMF Workflows schema is enhanced. z/OSMF makes all of the publicly visible instance variables for the calling workflow available to the called workflow. These variables are referred to as *caller scope* variables. They are shared only with the called workflow. For more information, see [“Caller scope variables” on page 870](#).
- A new property for workflows was added to the software management REST API. For more information, see [“Software management services” on page 412](#) and [Appendix C, “Understanding the Portable Software Instance descriptor file,” on page 983](#).
- A new property for data set property labels was added to the software management REST API. For more information, see [“Software management services” on page 412](#) and [Appendix C, “Understanding the Portable Software Instance descriptor file,” on page 983](#).
- A z/OSMF workflow includes a number of "built-in" variables that workflow authors can use. These variables are called *workflow internal variables*. You can reference them in your workflow definition without the need for you to define them. For more information, see [“Workflow internal variables ” on page 864](#).
- A new field that is called `requires-zcx-addr` is added to [“Obtain an IP address” on page 51](#).

## Changed

The following content is changed.

### April 2021 refresh

- The HTTP error status response codes are changed. For more information, see [Table 411 on page 720](#)

### January 2021 refresh

- The izud-datasets workflow variable was updated to include the izud-cataloged and izud-catalog properties. For more information, see [Appendix D, “Software Management workflow variables,”](#) on page 991.

#### **December 2020 refresh**

The description for z/OS data set and file REST interface services error reporting category 8 is updated with information about dynamic allocation errors. For more information, see [“Category 8 — z/OS XL C/C++ Conditions”](#) on page 647.

#### **Prior to the August 2020 refresh**

- The description for the volume-serial field was corrected in [“Add a new software instance”](#) on page 428.
- The software management services Export REST API was updated to allow the input of a work data set name prefix and volume or storage class. For more information, see [“Export a defined software instance”](#) on page 434.
- Global variables are deprecated. Use instance variables or system variables instead. Future releases might not support global variables.

### **Deleted**

The following content was deleted.

#### **December 2020 refresh**

- Mention of return of a stack trace for error handling was removed from the Software Management REST APIs section. For more information, see [“Software management services”](#) on page 412.

#### **Prior to the July 2020 refresh**

- The tables that list all HTTP status reason codes were removed from the software management services section. The reason codes themselves, which are returned to the calling client, include all needed information. For more information, see [“Software management services”](#) on page 412.

## **Summary of changes made in z/OSMF Version 2 Release 3, as updated June 2019**

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

This information contains new or changed information for maintenance.

### **New information**

Provider defined properties can be specified when you use the Software Management REST services to add or modify a software instance. See:

- [“Add a new software instance”](#) on page 428
- [“Modify the properties of a software instance”](#) on page 440
- [Appendix D, “Software Management workflow variables,”](#) on page 991

You can use the Swagger interface to display information about the following z/OSMF REST APIs:

- [“Notification services”](#) on page 404
- [“TSO/E address space services”](#) on page 481
- [“z/OS console services”](#) on page 526
- [“z/OSMF information retrieval service”](#) on page 699
- [“z/OSMF workflow services”](#) on page 725

## Changed information

Global variables are deprecated, as of z/OS V2R3. IBM recommends that you use instance variables or system variables, instead. Global variables might not be supported in a future release.

z/OSMF includes a number of "built-in" workflow variables, which might provide the function that you require. For more information, see [“Workflow internal variables ” on page 864](#).

## Summary of changes made in z/OSMF Version 2 Release 3, as updated March 2019

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

This information contains new or changed information for maintenance.

### New information

Software Management support for workflow variables is added with the PTFs for APAR PH09032. Using workflows in Software Management you can perform setup and configuration tasks for a software instance. See [Appendix D, “Software Management workflow variables,” on page 991](#).

## Summary of changes made in z/OSMF Version 2 Release 3, as updated December 2018

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

This information contains new or changed information for maintenance.

### New information

New function is available for z/OSMF V2R3 when you install the December 2018 functional updates.

Cloud Provisioning services REST APIs are enhanced as follows.

- New clustered composite templates allow you to leverage sysplex capabilities to provision a continuously available middleware environment. With a single provisioning action, you can provision network-clustered instances of a specific middleware in a sysplex. See [“Composite templates” on page 165](#). These APIs are enhanced: [“Obtain an IP address” on page 51](#), [“Obtain a port” on page 59](#), [“Get a resource pool” on page 132](#), [“List the resource pools” on page 147](#), [“Create a software services template” on page 170](#), [“Create a new version of a software services template ” on page 178](#), [“Modify a software services template” on page 188](#), [“List the software services templates” on page 220](#), [“Get a published software service template” on page 257](#), [“List the published software service templates” on page 278](#), [“Create a software services instance” on page 288](#), [“Get the contents of a software services instance” on page 301](#) and [“List the software services instances” on page 312](#).
- Workflow definition files and file templates can now be in sequential or partitioned data sets. These APIs are affected: [“Create a software services template” on page 170](#), [“Create a new version of a software services template ” on page 178](#), [“Modify a software services template” on page 188](#), [“Get a published software service template” on page 257](#) and [“Get a published software service template” on page 257](#).
- Several APIs are updated to improve diagnostic capability related to provisioning a software services instance: [“Create a software services instance” on page 288](#), [“Get the contents of a software services instance” on page 301](#) and [“List the software services instances” on page 312](#).
- A new REST API lets you submit multiple approvals or rejections with a single request. See [“Batch approve approval records for a software services template” on page 243](#).

The z/OSMF Workflows schema is enhanced, as follows:

- REST steps can now specify the HTTPS protocol for secure connections. Here, you must also specify the host name for the receiving system, and the user name and password that are to be used for logging in to the receiving system. Previously, REST steps were limited to using the HTTP protocol.
- An array is a new type of variable that you can define for your workflow. Use an array variable to represent a list of values or name-value pairs. As with other types of variables, you can preset the values by using a workflow variable input file.
- The files for a workflow — the primary XML file, fragment files, and the variable input file — can now reside in sequential or partitioned data sets. Previously, these objects were required to be z/OS UNIX files.
- When you create a file template step, you can now use variable substitution for the file template location.

The Workflow Editor now includes a "toolbox" of IBM-supplied steps, which are designed for performing common tasks on z/OS, such as creating a data set or submitting a REST request. When you create a workflow definition, check the step toolbox to see whether any of the steps would be applicable to your needs. If so, you can save time by importing an IBM-supplied step from the shared step library and modifying it, rather than creating your own step.

For more information about z/OSMF workflows, see [Chapter 2, “Creating workflow definitions for z/OS,”](#) on page 803.

## Changed Information

The Software Management task is enhanced to allow you to view, create and perform workflows defined to a specified software instance and for software that is being deployed.

- Request content for adding a new software instance, modifying the properties of a software instance and retrieving the properties of a software instance has been updated to support workflows.
  - [“Add a new software instance”](#) on page 428
  - [“Modify the properties of a software instance”](#) on page 440
  - [“Retrieve the properties of a software instance”](#) on page 417
- The portable software instance descriptor file has been updated to support workflows: [Appendix C, “Understanding the Portable Software Instance descriptor file,”](#) on page 983.
- The expected response for the list the data sets included in a software instance page has been updated to support workflows: [“List the data sets included in a software instance”](#) on page 423.

## Summary of changes made in z/OSMF Version 2 Release 3, as updated August 2018

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

This information contains new or changed information for maintenance.

### New information

Enhancements are made to the z/OS data set and file REST interface API for the list MVS subsystem service. See [“List MVS subsystems”](#) on page 402.

## Summary of changes made in z/OSMF Version 2 Release 3, as updated July 2018

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

This information contains new or changed information for maintenance.

## New information

In support of APAR PI98415, new return and reason codes are added to the HTTP error responses for z/OS console services API. See [“z/OS console services”](#) on page 526.

## Summary of changes made in z/OSMF Version 2 Release 3, as updated June 2018

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

## New information

New function is available for z/OSMF V2R3 when you install the June 2018 functional updates.

As a workflow author, you can request that automated steps be run in parallel, rather than sequentially. Through parallel processing, a workflow can take less time to complete. A workflow with steps that can be run in parallel is called a *parallel-steps workflow*. For more information, see [“Enabling automated steps for parallel processing”](#) on page 849.

Enhancements are made to the z/OS Workflows REST interface services API. The following services are updated to return the properties of a workflow that contains parallel steps. The properties are included in the JSON object that is returned to the requester.

- [“Get the properties of a workflow”](#) on page 736
- [“Retrieve a workflow definition”](#) on page 770
- [“Get the properties of an archived workflow”](#) on page 787

z/OSMF includes a number of internal variables that you can reference in your workflow definition without having to define them. For more information, see [“Workflow internal variables”](#) on page 864.

Cloud Provisioning services REST APIs are enhanced as follows.

- You can use the Swagger interface to display information about the Cloud Provisioning REST APIs. See [“Using the Swagger interface”](#) on page 49.

## Changed information

Cloud Provisioning services REST APIs are enhanced as follows.

- Resource Management REST APIs support an option to allow members of a tenant to view and perform actions against provisioned software instances that are associated with the tenant, without the requirement that the users be owners of the instances or domain administrators. See [“Get a resource pool”](#) on page 132 and [“List the resource pools”](#) on page 147. The authorization requirements of the following APIs are updated: [“Get the contents of a software services instance”](#) on page 301, [“List the software services instances”](#) on page 312, [“Update variables in a software services instance”](#) on page 330, [“Get key-value variables for a software services instance”](#) on page 323, [“Update a software services instance”](#) on page 325, [“Update variables in a software services instance”](#) on page 330, [“Perform an action against a software services instance”](#) on page 337, [“Resume an action workflow”](#) on page 340, [“Retry an action workflow”](#) on page 344, [“Get the response for an action performed against a software services instance”](#) on page 346, [“List the responses for actions performed against a software services instance”](#) on page 351, and [“Delete the response for an action performed against a software services instance”](#) on page 354.
- Updates are made to support selecting a source variable from the set of public variables when defining connector variables for a composite software services template. See [“Get a software services template”](#) on page 195 and [“Get a published software service template”](#) on page 257.
- External files referenced by a workflow or action definition file using the fileTemplate element can now reside in sequential or partitioned data sets as well as in z/OS UNIX files. See [“Create a software services template”](#) on page 170, [“Create a new version of a software services template”](#) on page 178,

[“Modify a software services template” on page 188](#), and [Chapter 4, “Preparing software to exploit cloud provisioning,” on page 971](#).

- Several APIs are enhanced to provide toleration support of the data that cloud provisioning maintains in the z/OSMF persistent data. New fields for provisioning version and support for provisioning versions are added. See [“Get a domain” on page 79](#), [“List the domains” on page 86](#), [“Get a tenant” on page 94](#), [“List the tenants” on page 103](#), [“Get a resource pool” on page 132](#), [“List the resource pools” on page 147](#), [“Get a software services template” on page 195](#), [“List the software services templates” on page 220](#), [“Get a published software service template” on page 257](#), [“List the published software service templates” on page 278](#), [“Get the contents of a software services instance” on page 301](#), [“List the software services instances” on page 312](#), and [“List the software service instance names” on page 360](#).
- A new workflow-message-text field is added to [“Get the contents of a software services instance” on page 301](#).
- [“Get a domain” on page 79](#) and [“List the domains” on page 86](#) are enhanced to show details about a domain's error state.

## Summary of changes made in z/OSMF Version 2 Release 3, as updated March 2018

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

This information contains new or changed information for maintenance.

### New information

Enhancements are made to the TSO/E address space REST interface services API to support starting a TSO/E address space or TSO application that is running on a remote system in the sysplex. See [“Start or reconnect to a TSO/E address space” on page 485](#).

## Summary of changes made in z/OSMF Version 2 Release 3, as updated December 2017

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

### Changed information

Enhancements to:

- [“z/OS data set and file REST interface” on page 563](#)
- [“Write data to a z/OS data set or member” on page 580](#)
- [“List the files and directories of a UNIX file path” on page 598](#)
- [“List the z/OS data sets on a system” on page 569](#)
- [“Create a sequential or partitioned data set” on page 585](#)
- [“Write data to a z/OS UNIX file” on page 607](#)
- [“JSON document specifications for z/OS data set and file REST interface requests” on page 631](#)
- [“Error reporting categories” on page 641](#)

Function is changed for z/OSMF V2R3 when you install the December 2017 functional updates.

Cloud Provisioning services REST APIs are enhanced as follows.

- Software services instance REST APIs support a new composite template, which can be used to provision multiple software services with a single Run operation. With the introduction of composite

templates, the previously supported templates, which can be used to provision only a single software service, are referred to as standard templates. For more information, see [“Software services template services”](#) on page 163, [“Create a software services template”](#) on page 170, [“Get a software services template”](#) on page 195, [“List the software services templates”](#) on page 220, [“Test a software services template”](#) on page 228, [“Get a published software service template”](#) on page 257, [“List the published software service templates”](#) on page 278, [“Run a published software service template”](#) on page 252, [“Create a software services instance”](#) on page 288, [“Get the contents of a software services instance”](#) on page 301, and [“List the software services instances”](#) on page 312.

- Support is added for shared resource pools, which let you use a single resource pool for multiple templates that are associated with a single tenant. See [“Get a tenant”](#) on page 94, [“List the tenants”](#) on page 103, and [“Get a resource pool”](#) on page 132.
- Support is added for tenant-level metering and capping. See [“Get a tenant”](#) on page 94 and [“Get a resource pool”](#) on page 132.
- [“Get a domain”](#) on page 79 now returns a domain state.
- Support is added for RunAsUser user IDs being supplied dynamically by a workflow. See [“Get a software services template”](#) on page 195, [“Create a software services instance”](#) on page 288, and [“Get the contents of a software services instance”](#) on page 301.
- Tenants now support a Solution ID for use with Container Pricing for IBM Z. See [“Get a tenant”](#) on page 94.
- The Workflows XML schema is enhanced with new elements and attributes, which enable workflow authors to:
  - Define a runAsUser ID dynamically during workflow processing by using variable substitution. When the step is performed, the Workflows task processes the variable substitution to derive the actual user ID for the step. For more information, see [“runAsUser identity for a step”](#) on page 852.
  - For a template step, specify a JES spool file that is produced by a job step as the location for the properties file. To do so, include the sysoutDD= attribute on the output element and set this attribute to true. For more information, see [“Creating a properties file”](#) on page 827.
- For template steps, the output properties file is read under the step performer identity. Previously, it was necessary to grant the z/OSMF server user ID access to the properties file. For more information, see [“Template steps”](#) on page 822.

## Summary of changes made in z/OS Version 2 Release 3, as updated October 2017

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-30, which supports IBM z/OS Management Facility Version 2 Release 3.

### New information

For users of the z/OSMF Representational State Transfer (REST) APIs, new information is provided, as follows:

- [“Allowing cross-site access to REST services”](#) on page 5
- [“Enabling cross-origin resource sharing \(CORS\) for REST services”](#) on page 5.

## Summary of changes made in z/OS Version 2 Release 3

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-06, which supported IBM z/OS Management Facility Version 2 Release 2.

### New information

In z/OS V2R3, the base element z/OSMF is started by default at system IPL. This behavior, which is referred to as *z/OSMF autostart*, means that z/OSMF is available for use as soon as the system is up.

Authorized programs can use the event notification facility (ENF) to determine whether the z/OSMF server is up or down.

For more information, see [Autostart concepts in z/OSMF](#) in *IBM z/OS Management Facility Configuration Guide*.

## Changed information

Enhancements are made to the Notification services, as follows:

- The service that is described in [“Get all of the notifications received by the current user”](#) on page 405 can return the email addresses of recipients in the property assignees.
- The service that is described in [“Send a notification from a z/OSMF task, when the content is the message from the bundle file”](#) on page 407 can specify the following information on a notification request:
  - Email addresses of all the recipients
  - An email notification format.
- The service that is described in [“Send a notification and mail from a z/OSMF task or z/OSMF user”](#) on page 409 can specify the following information on a notification request:
  - Email addresses of all the recipients
  - An email notification format.

For more information, see [“Notification services”](#) on page 404.

## Summary of changes made in z/OSMF Version 2 Release 2, as updated April 2017

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-05, which supports IBM z/OS Management Facility Version 2 Release 2.

## New information

New function is available for z/OSMF V2R2 when you install the April 2017 functional updates.

- Software services instance REST API to indicate that security setup for a software services template is complete. For more information, see [“Set security complete for a software services template”](#) on page 248.

## Changed information

Function is changed for z/OSMF V2R2 when you install the April 2017 functional updates.

Cloud Provisioning services REST APIs are enhanced as follows:

- Domains can have more than one system. For more information, see [“Resource management services”](#) on page 76 and [“Software services instance services”](#) on page 286. See also new system-nicknames fields for [“Test a software services template”](#) on page 228 and [“Run a published software service template”](#) on page 252.
- Switching to a runAsUser user ID can be audited. See [“Get a software services template”](#) on page 195, [“Get a published software service template”](#) on page 257, and [“Create a software services instance”](#) on page 288.
- Security can be defined manually, rather than being defined automatically by z/OSMF. See [“Get a domain”](#) on page 79, [“Get a software services template”](#) on page 195, and [“Get a published software service template”](#) on page 257.
- The provisioning workflow that is provided when creating or modifying a software services template must be of type provisioning. See [“Create a software services template”](#) on page 170, [“Create a new](#)

[version of a software services template](#)” on page 178, and [“Modify a software services template”](#) on page 188.

- New fields indicate when a software services template was published or archived. See [“Get a software services template”](#) on page 195 and [“Get a published software service template”](#) on page 257.
- Provisioning and actions workflows can be suspended and resumed. For more information, see [“Software services instance services”](#) on page 286.
- The disposition of workflows and jobs after they complete can be specified. See [“Create a software services template”](#) on page 170, [“Create a new version of a software services template”](#) on page 178, [“Modify a software services template”](#) on page 188, [“Get a software services template”](#) on page 195, [“Get a published software service template”](#) on page 257, and [“Create a software services instance”](#) on page 288. With the addition of a new workflows-disposition field, the workflows-clean-after-provisioned field is deprecated.
- A new updateRegistry attribute on the action workflow variable definition indicates if the software services instance should be updated with the variable after the action workflow completes. For a description of the actions definition file, see [“Actions definition file”](#) on page 973. The update-registry field is added to software services instances APIs. See [“Create a software services instance”](#) on page 288, [“Get the contents of a software services instance”](#) on page 301, [“Get the variables for a software services instance”](#) on page 320, [“Update a software services instance”](#) on page 325 and [“Update variables in a software services instance”](#) on page 330.

The Workflows services REST API is enhanced to return more information about:

- Jobs that are run by steps
- Provisioning workflows.

For information, see [“Get the properties of a workflow”](#) on page 736.

Workflows XML schema is enhanced with new elements and attributes, which enable workflow authors to:

- Use substitution (variables) in the workflow variable input file. For information, see [“Using variable substitution in the workflow variable input file”](#) on page 872.
- Specify a relative path when referring to the location of a called workflow definition. For information, see [“Designing a step to call another workflow”](#) on page 842.
- Force a stop in a sequence of automated steps, and, optionally, have z/OSMF send an email notification to one or more recipients that you specify. For information, see [“Using the suspend element to control automation”](#) on page 848.

## Summary of changes made in z/OSMF Version 2 Release 2, as updated March 2017

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-04, which supports IBM z/OS Management Facility Version 2 Release 2.

### Updated information

- Updated Software Management REST APIs including Add a software instance, Modify a software instance, and Retrieve the properties of a software instance now support non-SMP/E managed product information.

## Summary of changes made in z/OSMF Version 2 Release 2, as updated December 2016

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SC27-8420-02, which supports IBM z/OS Management Facility Version 2 Release 2.

## New information

New function is available for z/OSMF V2R2 when you install the December 2016 functional updates.

- New REST APIs that can be used to provision z/OS software in support of IBM Cloud Provisioning and Management for z/OS are added. For an introduction, see [“Cloud provisioning services” on page 45](#). For more information about the APIs, see the following:
  - Resource management REST APIs, for managing domains and tenants. For more information, see [“Resource management services” on page 76](#).
  - Resource Pool REST APIs, for managing network resource pools. For more information, see [“Resource pool services” on page 49](#).
  - Software services template REST APIs, for managing templates that are used to provision software. For more information, see [“Software services template services” on page 163](#) and [“Published software service template services” on page 250](#).
  - Software services instance REST APIs that are used to manage software services instances that represent provisioned software. For more information, see [“Software services instance services” on page 286](#).
  - Software service instance name (SSIN) APIs. For more information, see [“Software service instance name services” on page 356](#).
  - Workload management resource APIs. For more information, see [“WLM resource pooling services” on page 513](#).

A new topic has been added for software providers, [Chapter 4, “Preparing software to exploit cloud provisioning,” on page 971](#).

- New REST APIs that can be used to work with archived workflows, as follows:
  - [“Archive a workflow instance” on page 783](#)
  - [“List the archived workflows for a system” on page 785](#)
  - [“Get the properties of an archived workflow” on page 787](#)
  - [“Delete an archived workflow” on page 801](#)
- New and changed REST APIs for working with z/OS data sets, including z/OS UNIX Systems Services files. See [“z/OS data set and file REST interface” on page 563](#).
- System variable REST APIs. For more information, see [“z/OSMF system variable services” on page 709](#).
- Enhancements are made to the Workflows XML schema that is supplied with z/OSMF. This file provides the XML syntax and rules for creating a workflow definition. The enhancements allow workflow authors to create workflow *REST steps*, which are steps that issue Representational State Transfer (REST) requests, such as GET or PUT.

For information about REST steps and the schema elements that you can use to create them, see [“REST steps” on page 835](#).

- Enhancements are made to the z/OS Workflows REST interface services API. This programming interface allows the caller to create and manage z/OSMF workflows on a z/OS system. The enhancements support the use of REST steps, as follows:
  - The service that is described in [“Get the properties of a workflow” on page 736](#) is updated to return the properties of a step that issues a REST request. The properties are included in the step-info JSON object that is returned to the requester.
  - The service that is described in [“Retrieve a workflow definition” on page 770](#) is updated to return the schema elements for the REST steps in a workflow.
- New terms are introduced. To help you distinguish REST steps from other types of steps, the terms *template step* and *REST step* are introduced in this publication, as follows:
  - *Template step* is used to indicate a step that runs a program, such as a JCL job, REXX exec, or UNIX shell script
  - *REST step* is used to indicate a step that issues a REST request, such as GET or PUT.

To give workflow authors more control over how variables are used, the following options are added to the Workflows XML schema:

- New attribute `visibility`, which is added to the element `variable` (`<variable>`), specifies whether a variable is intended for public or private use. See [“Specifying the variable element and its attributes” on page 858](#).
- New element `atCreate` (`<atCreate>`) can be used to set attributes for a variable across workflows. See [“Using the element `atCreate` to qualify a variable definition” on page 861](#).

The Workflows task of z/OSMF now includes an editor for workflows. You can use the Workflow Editor to view, create, and modify workflow definitions. The Workflow Editor provides a visual framework for working with the elements of a workflow definition. To get started with the Workflow Editor task, in the navigation area, select **Workflow Editor**. For more information, see [“Workflow Editor task in z/OSMF” on page 806](#).

## Summary of changes made in z/OSMF Version 2 Release 2, as updated September 2016

---

This information was previously presented in *IBM z/OS Management Facility Programming*, SC27-8420-02, which supported IBM z/OS Management Facility Version 2 Release 2.

### New

The z/OS console services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. For information, see [“z/OS console services” on page 526](#).

The Workflows XML schema is enhanced with new capabilities for creating workflows, as follows:

- It is possible to run an executable program (a REXX exec or UNIX shell script) in real time, directly from a workflow step. The program can be coded inline within the step or referenced from an external file. In previous releases, to run such programs, it was necessary to submit a batch job, which also meant that the results of the program were not available for immediate evaluation by the step owner. With this enhancement, the step owner can know the results of the program as soon as it completes.

For more information, see [“Template steps” on page 822](#).

- It is possible to collect feedback from the users of a workflow. A workflow author can optionally include a feedback form on one or more steps with customized questions for the user to answer at the conclusion of a step. Such feedback can be useful for determining the effectiveness of a workflow design, or collecting user requirements for future enhancements to a workflow.

For more information, see [“Collecting user feedback” on page 818](#).

A *portable software instance* can be used to simplify distribution of a software instance across a network, and can be deployed by the z/OSMF Software Management task. For information about this new packaging format, see [Appendix C, “Understanding the Portable Software Instance descriptor file,” on page 983](#).

### Changed

The following services are updated in support of new workflow schema elements:

- [“Create a workflow” on page 728](#)
- [“Get the properties of a workflow” on page 736](#)
- [“List the workflows for a system or sysplex” on page 759](#)

## Summary of changes made in z/OSMF Version 2 Release 2, as updated June 2016

---

This information was previously presented in *IBM z/OS Management Facility Programming*, SC27-8420-01, which supported IBM z/OS Management Facility Version 2 Release 2.

### New

z/OSMF Representational State Transfer (REST) services for notification services. See [“Notification services”](#) on page 404.

Enhancements are made to the z/OSMF workflow services API, as follows:

- *Access type* is a new security control for workflows. By specifying an access type for a workflow, you can restrict the amount of information that is available to users regarding workflow steps, variables, and notes.

The following services are updated in support of the access type:

- [“Create a workflow”](#) on page 728
- [“Get the properties of a workflow”](#) on page 736
- [“List the workflows for a system or sysplex”](#) on page 759

### Changed

z/OSMF REST APIs for z/OS jobs now include data for active job steps. For more information, see [“Obtain the status of a job”](#) on page 652 and [“JSON document specifications for z/OS jobs REST interface requests”](#) on page 682.

## Summary of changes made in z/OSMF Version 2 Release 2, as updated March 2016

---

This information was previously presented in *IBM z/OS Management Facility Programming*, SC27-8420-00, which supported IBM z/OS Management Facility Version 2 Release 2.

### New

New REST APIs for Files and Data sets have been added. See [“z/OS data set and file REST interface”](#) on page 563

For workflow authors, it is possible to specify a workflow scope of *none* to cause a new instance of a called workflow to always be created. See [“Coordinating workflow-to-workflow actions”](#) on page 841.

### Changed

z/OSMF REST APIs for Files and Data sets have been modified. See [“z/OS data set and file REST interface”](#) on page 563.

## Changes made in z/OSMF Version 2 Release 2, SC27-8420-00

---

This information was previously presented in *IBM z/OS Management Facility Programming Guide*, SA32-1066-04, which supported IBM z/OS Management Facility Version 2 Release 1.

### New information

The multisystem routing services are enhanced to allow a calling application to:

- Retrieve data from all the systems in a sysplex or CPC.

- Update data for all the systems in a sysplex or CPC.
- Delete data from all the systems in a sysplex or CPC.

For more information, see [“Multisystem routing services” on page 374](#).

New services are added to the software management API to allow a calling application to:

- List the data sets included in a software instance.
- Modify the properties of a software instance.
- Retrieve the products, features, and FMIDs for a software instance.
- Remove a software instance from z/OSMF.

For more information, see [“Software management services” on page 412](#).

The topology services adds support for central processor complexes (CPCs) so that you can retrieve a list of the systems included in a CPC. For more information, see [“List the systems included in a CPC” on page 478](#).

The z/OS jobs REST interface services are enhanced to allow a calling application to retrieve information about each step in a job, such as the step name, step number, and completion code, when obtaining the status of a job. For more information, see [“Obtain the status of a job” on page 652](#).

New elements in the Workflows XML schema provide additional capabilities, as follows:

- You can enable a workflow definition for future upgrades. If so, users of your workflow can upgrade a workflow to the latest level without losing their changes or having to start again with a new instance. For information, see [“Enabling a workflow definition file for future upgrades” on page 817](#).
- Workflows users can add their own steps to a workflow by using **Update Workflow Steps** action that is provided in the Workflows task. Users can also modify or delete the user-supplied steps, as needed. For more information, see the Workflows task online help.
- You can use substitution variables in the file path of an external file, such as a step template. For information, see [“Using variable substitution in the workflow definition file path” on page 811](#).
- You can use shorter names for instance variables; see [“Simplified instance variable format in substitution and conditions” on page 863](#).

## Changed information

z/OSMF adds support for managing multiple z/OS sysplexes from a single z/OSMF instance. If you are creating your own plug-in, use the new *task-multi-sysplex-scope* property in the plug-in property file to indicate whether the tasks in your plug-in can be used to manage or display data for multiple z/OS sysplexes. For more information, see [“Adding your applications to z/OSMF” on page 963](#).

If you are creating your own plug-ins and are supplying context-sensitive documentation, the URL for linking the help content to the user interface has changed to:

```
https://<host>:<port>/<context-root>/helps/SSB2H8_2.2.0/<help-plugin-name>/<help-
topic>
```

For information, see [“Adding links to help plug-ins” on page 962](#).

## Information applicable to all releases

This information contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line in the margin by the change.

The *Readers' Comments - We'd Like to Hear from You* section at the end of this publication has been replaced with a new section [“How to send your comments to IBM” on page li](#). The hardcopy mail-in form has been replaced with a page that provides information appropriate for submitting comments to IBM.

---

# Chapter 1. Using the z/OSMF REST services

z/OSMF supports the use of Representational State Transfer (REST) APIs, which are public APIs that your application can use to work with system resources and extract system data. As with implementations of REST services on other platforms, the z/OSMF APIs allow for easy-to-use services that are language- and platform-independent, stateless, scalable, and easily parsed.

## Processing overview

The z/OSMF REST services can be invoked by any hypertext transfer protocol (HTTP) client application, running on the z/OS local system or a remote system. The services support requests in either of the following protocols: HTTP/1.0 or HTTP/1.1.

Conceptually, your application (the client) issues requests to the target system (z/OS) in the form of request messages. The product name is z/OS. Each request message consists of a request line, optionally followed by request headers (HTTP headers), an empty line, and an optional message body. The request line includes the HTTP method, such as GET, a Universal Resource Locator (URL) and, where appropriate, parameters that further qualify the request. The URL is required and must be URI-encoded as specified in RFC 2396. For more information about RFC 2396, see the [Uniform Resource Identifiers \(URI\): Generic Syntax \(www.ietf.org/rfc/rfc2396.txt\)](http://www.ietf.org/rfc/rfc2396.txt) web page.

If the API determines that the request is valid, it performs the requested service. After the API performs the service, it creates an HTTP response. If the request is successful, this response takes the form of an HTTP 200 (OK) response and, if applicable, an object that contains a result set. Depending on which service was requested, the result set might be returned in a format that requires parsing by your program, for example, a JavaScript Object Notation (JSON) object. In other cases, results might be returned in another format, such as plain text or binary data. If the request is not successful, the response consists of a non-OK HTTP response code with details of the error that is provided in the form of a JSON object.

It is assumed that users of these services are familiar with the JSON standard and coding practices. The following references provide more helpful information:

- Hypertext Transfer Protocol 1.1: <http://www.w3.org/Protocols/>
- Multipurpose Internet Mail Extensions (MIME) media types: <http://www.iana.org/assignments/media-types/index.html>
- Introducing JSON: <http://www.json.org>.

## Using the Swagger interface

You can use the Swagger interface to display information about the following z/OSMF REST APIs:

- [“Cloud provisioning services” on page 45](#)
- [“Notification services” on page 404](#)
- [“TSO/E address space services” on page 481](#)
- [“z/OS console services” on page 526](#)
- [“z/OS data set and file REST interface” on page 563](#)
- [“z/OS jobs REST interface” on page 647](#)
- [“z/OSMF information retrieval service” on page 699](#)
- [“z/OSMF workflow services” on page 725](#)

To enable the use of Swagger at your installation, you must define the Swagger resources in your external security manager, and grant READ access to the appropriate users and groups. Ask your security administrator to do the following:

- You and z/OSMF administrators require READ access to the IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers resource in the EJBROLE class. The following example shows RACF® commands that grant the access:

```
RDEFINE EJBROLE IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers
UACC(NONE)

PERMIT IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE)
ID(IZUUSER) ACCESS(READ)

PERMIT IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE)
ID(IZUADMIN) ACCESS(READ)
```

- z/OSMF administrators require READ access to the IZUDFLT.com.ibm.ws.management.security.resource.Administrator resource in the EJBROLE class. The following example shows RACF commands that grant the access:

```
RDEFINE EJBROLE IZUDFLT.com.ibm.ws.management.security.resource.Administrator UACC(NONE)

PERMIT IZUDFLT.com.ibm.ws.management.security.resource.Administrator CLASS(EJBROLE)
ID(IZUADMIN) ACCESS(READ)
```

After the access is defined, use this address in a web browser to display information about the REST APIs:

```
https://<hostname>:<port>/ibm/api/explorer
```

Supply the appropriate value for hostname and port.

## Authenticating to z/OSMF

The z/OSMF REST services API is a secure interface that requires authentication. z/OSMF supports the following methods for passing user credentials to the API:

- Basic authentication
- Certificate authentication
- Web token authentication

Basic authentication means that the client program provides a z/OS user ID and password in the header of the initial request.

The following topics describe each method of authentication:

- [“Basic authentication” on page 2](#)
- [“Certificate authentication” on page 3](#)
- [“Web token authentication” on page 3](#)

## Basic authentication

Your client program authenticates to the z/OSMF REST services API with a valid z/OS user ID and password.

Typically, authentication is done through the HTTP header that is included in each client request. However, you might also consider using a single sign-on technique, such as the following:

1. On the first request to the server, the client request includes a basic HTTP authentication header that contains a valid user ID and password. The header property value pair should look like this example:

```
'Authorization':'Basic <encoding of userid:password>'
```

where <encoding of userid:password> is a base 64 encoding of <userid>:<password>.

2. On successful log-in, your application receives the following values in the response header:
  - HTTP status code 200

- Lightweight Third Party Access (LTPA) token, which contains the credentials for your program. For z/OSMF, the token is a LtpaToken2 value, which supports strong encryption.
3. On subsequent requests, your program supplies the LTPA token for authentication with the z/OSMF REST services API, instead of the basic HTTP authorization header. You can provide the LTPA token through the Cookie header property, for example:

You can provide the LTPA token through the Cookie header property, for example:

```
'Cookie': '<ltpaToken2=<tokenvalue>'
```

followed by the token, for example:

```
'Cookie': 'LtpaToken2=IExabotu2sfNbJij6rajHJcFiDi  
1H0hm13yKvy1wfJ4q8goCFEYH41FQN1AgdMMVP6/nVbH/IKw  
015b7ZqWuZ8nd0YcECAJg1ss2Vq4q21C1jLvVGTyNLk6rvbgs  
7oQWM98bSuAN1Qtvlrx9uZ8EY4GqqscErQ09vriAhwgkcedWB  
jn21LNjl+G8o1JA4uB+Cv5XamrUvziY2pcbCKjFjNt5EQ97Nf2  
sBzvlanfzrENhV9u0LRpw9DibrzKLh0R1f0rp5xySAe7Ery69  
eynt4ItaVWCcpt+CYHFbpHpW/C7INWHeNcaNktr0DBmHh6EW1; '
```

## Certificate authentication

Your client program authenticates to the z/OSMF REST services API with a certificate.

With a client certificate, your program can access z/OSMF without having to supply a user ID and password. If your client is browser-based, the certificate is stored in the browser itself. When you log in to z/OSMF, the server requests the certificate from your browser. If your browser stores more than one certificate, you might be prompted to select the correct one to use with z/OSMF. Otherwise, your browser sends the certificate to z/OSMF. After z/OSMF identifies you as the owner of the key that is associated with the certificate, a secure connection is established.

To use client certificate authentication for the z/OS jobs REST interface services, you:

1. Create the certificate. You can create the certificate in RACF or import it into RACF.
2. Distribute the certificate to the appropriate workstations, for example, FTP it in binary.
3. Import the certificate into the browser, by using the procedure that is appropriate for that browser.

For more information about creating and managing digital certificates, see [RACF and digital certificates in z/OS Security Server RACF Security Administrator's Guide](#). For an example, see [Scenario 5: Creating client browser certificates with a locally signed certificate in z/OS Security Server RACF Security Administrator's Guide](#).

## Web token authentication

The z/OSMF authentication services API can be used to obtain authentication tokens (a JSON Web Token and an LTPA token) on the user's authentication request. This API can also be used to delete the current store of JSON Web Tokens and LTPA tokens. This API is provided for z/OSMF tasks and vendor applications. For more information, see [“z/OSMF authentication services” on page 718](#).

Web token support must be enabled on your z/OSMF system. For more information, see [Enabling JSON Web Token support in IBM z/OS Management Facility Configuration Guide](#).

## Supported HTTP versions

The z/OSMF REST interface services support requests in either of the following protocols: HTTP/1.0 or HTTP/1.1

## Usage considerations for the z/OSMF REST services

Observe the following considerations when you use the z/OSMF REST interfaces:

- As with any z/OSMF task or function, the REST services compete for z/OSMF resources with users of the z/OSMF web browser interface. Thus, concurrent high usage of the REST services can affect response time for users of the z/OSMF web browser interface.
- During periods of concurrent high usage of the REST services, an application can experience connection failures, such as connection refused, connection timed out, or connection reset. In these cases, the application should try the request again. The number of retry attempts needed depends on how much work is being requested of the server. It might be necessary for your installation to modify the workload and reduce the arrival rate of requests.
- Some browsing environments do not support all of the HTTP methods, such as HTML 4 or XHTML 1, or might block applications from accessing response content that has a non-successful HTTP response status code (4nn and 5nn). As a workaround, your application can use the following custom HTTP request headers:

**X-IBM-Requested-Method:**

GET, PUT, and DELETE requests can be "tunneled" through a **POST** method by using this custom HTTP header.

**X-IBM-Bypass-Status:**

If set to true, all response status codes are set to 200, and the custom HTTP response header **X-IBM-Actual-Status** is included in the returned data. To determine the original status code, your application must check the **X-IBM-Actual-Status** header.

## Timestamp data type

The timestamp data type is used in the definition of some data models and notification message formats in the z/OSMF REST services APIs. Where it appears in this information, the timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time that is expressed as the number of milliseconds since midnight on January 1, 1970 UTC.

## Enabling browser log in through a client certificate

It is possible to run the z/OS jobs REST interface services directly from a web browser. Here, you must first authenticate to z/OSMF through your browser. In z/OSMF, authentication is typically done by entering your user ID and password at the **Welcome** page. However, it is also possible to log in with a client certificate, if your installation favors this approach. With a client certificate, you can access z/OSMF through your browser without having to supply a user ID and password.

When a client certificate is created, it requires security associations. For example, on a system with RACF installed, your client certificate can be associated with a RACF user ID. After the user ID is associated with the client certificate, the certificate can be exported. You must get the client certificate from the server side by using the FTP command and typing in the hostname or IP address of the server. When prompted, enter your user ID and password, and use the **bin** command to transfer the file in binary format. Then, import the client certificate into the client's browser.

In client certificate authentication, the certificate is stored in the browser itself. When you log in to z/OSMF, the server requests the certificate from your browser. If your browser stores more than one certificate, you might be prompted to select the correct one to use with z/OSMF. Otherwise, your browser sends the certificate to z/OSMF. After z/OSMF identifies you as the owner of the key that is associated with the certificate, a secure connection is established.

If z/OSMF does not accept your client certificate, z/OSMF displays the Welcome page for you to enter your user ID and password.

If your installation plans to enable client certificate login for the z/OS jobs REST interface services, understand that it is your responsibility to create the certificate and manage its distribution to users. It is recommended that you ensure that users have browsers that support importing a certificate.

For more information about creating digital certificates, see [RACF and digital certificates](#) in *z/OS Security Server RACF Security Administrator's Guide*.

## Allowing cross-site access to REST services

The z/OSMF REST services can be accessed by an HTTP client application, or by a web application that is running on the same host system. By default, z/OSMF blocks access attempts from web applications on other host systems. In such cases, the request is failed with error message IZUG846W, which indicates that a Cross Site Request Forgery (CSRF) was attempted. In security terminology, a *CSRF* is a type of malicious attack on a website in which the attacker sends unauthorized commands to a web server or web application from a user that the server or application trusts.

You can enable your applications to make cross-site z/OSMF REST requests. To do so, you must perform the following steps:

1. Review your applications and identify those applications that use the z/OSMF REST services.
2. For any applications that make cross-site requests to z/OSMF REST services, update the application by adding the following HTTP custom header to every cross-site request:

### **X-CSRF-ZOSMF-HEADER**

This header is required for both browser and non-browser applications. Set the header to any value or an empty string ("").

3. Because most modern web browsers block cross-site access due to same origin restrictions, an extra step is required for browser applications — you must define the origin site of the web application to your installation security "allow list." Instructions for doing so are provided in [“Enabling cross-origin resource sharing \(CORS\) for REST services”](#) on page 5.

Non-browser applications, such as Java™ applications, require only the custom header. They do not require an "allow list" definition.

**Note:** z/OSMF includes the parmlib member IZUPRMxx, which your installation can use to override the z/OSMF default settings. IZUPRMxx includes the setting `CSRF_SWITCH=ON | OFF` to allow for disabling CSRF checking for all requests to the z/OSMF server. By default, `CSRF_SWITCH` is set to `ON` to ensure that your installation is protected against CSRF attacks. However, in some limited cases, such as for testing, you might choose to temporarily disable CSRF checking by setting `CSRF_SWITCH=OFF`. However, it is recommended that you leave this setting enabled to prevent CSRF attacks.

## Origin header

The Origin header indicates the origin of the cross-site access request (that is, the server that initiated the request). If you include the Origin header in a cross-site request, include the internet protocol (HTTP or HTTPS) with the origin domain. Otherwise, your request is blocked by the CSRF filter, which is enabled by default (`CSRF_SWITCH=ON`).

The following example shows the use of the Origin header with a z/OSMF REST API request. Notice that the internet protocol HTTP is included in the Origin header (in bold text).

```
PUT /zosmf/restconsoles/consoles/P083598 HTTP/1.1
Host: sys123.yourco.com
Content-Type:application/json
X-CSRF-ZOSMF-HEADER:zosmf
Origin:https://sys456.yourco.com
{
  "cmd": "F AXR,JFPAXR D T",
  "system": "sys123"
}
```

## Enabling cross-origin resource sharing (CORS) for REST services

Your installation can allow browser applications from certain, trusted sites to access z/OSMF REST services on the host system. If so, you must enable cross-origin resource sharing (CORS) on the host system. This work involves creating an "allow list" of exceptions (the trusted sites), and enabling those exceptions in your external security manager.

Identify which sites are to be allowed, and which REST interfaces are to be made available for cross-site access. Then, work with your security administrator to create the appropriate authorizations in your

external security manager. In a RACF installation, for example, define generic or discrete profiles for the remote sites in the ZMFAPLA class, and permit the profiles to the z/OSMF REST interfaces.

To define a profile for a remote site, use the following format:

```
<SAF_PREFIX>.REST.<identifier>.<reversed-hostname>
```

Where:

- <SAF\_PREFIX> is the SAF prefix for your z/OSMF configuration. By default, the prefix is IZUDFLT.
- REST.<identifier> identifies the REST interface that is to be allowed for use by the remote site. Table 1 on page 6 shows the identifiers for each of the z/OSMF interfaces. To indicate all REST interfaces, specify an asterisk (\*) as the identifier.

*Table 1. SAF identifiers for the z/OSMF REST interfaces*

REST interface	Identifier
Application Linking Manager interface services	APPLINK
Application server routing services	GATEWAY
Cloud provisioning services	PROVISIONING
Data persistence services	PERSIST
Multisystem routing services	GATEWAY
MVS™ subsystem services	FILES
Software management services	SWMGMT
Topology services	SYSTEM
TSO/E address space services	TSO
z/OS console REST interface services	CONSOLE
z/OS data set and file REST interface services	FILES
z/OS jobs REST interface services	JOBS
z/OSMF workflow services	WORKFLOW

- <reversed-hostname> is the site's fully qualified domain name in reverse, for example, the domain WWW.IBM.COM would be specified as COM.IBM.WWW. Use uppercase letters for any alphabetic characters in the profile. Specify a domain name only; not the full URL. Omit the protocol (HTTP:// or HTTPS://).

If the site is known by an IP address, specify the IP address in reverse. For any IP addresses that you define, it is recommended that you create a discrete profile (without wildcards) for each address. Use a valid "dotted decimal" Internet Protocol version 4 (IPv4) address. IPv6 addresses and internationalized domain name (IDN) addresses are not supported.

Though not recommended, you can use generic profiles with wildcard characters to allow access from multiple domains or subdomains. You might do so temporarily to allow for internal testing of the REST interfaces across multiple sandbox systems.

For example, assume that your installation wants to allow a web browser application on the site "lab2.ibm.com" to send requests for z/OS jobs REST interface services on the site "lab1.ibm.com." To allow a web browser application to issue cross-site requests from the "lab2" site, your RACF security administrator would create the authorization on the "lab1.ibm.com site," as follows:

1. Create a profile for the "lab2.ibm.com" site in the ZMFAPLA class. In the profile, include the identifier to represent the resource (the REST interface), for example, JOBS to indicate the z/OS jobs REST interface. In RACF, enter the **RDEFINE** command, as follows:

```
RDEFINE ZMFAPLA IZUDFLT.REST.JOBS.COM.IBM.LAB2 UACC(NONE)
```

To allow this cross-site access for all of the REST interfaces, specify a wildcard (\*) in place of a specific resource identifier. For example:

```
RDEFINE ZMFAPLA IZUDFLT.REST.*.COM.IBM.LAB2 UACC(NONE)
```

2. Ensure that the z/OSMF server user ID (by default, IZUSVR) has READ access to the profile. In RACF, enter the **PERMIT** command, as follows:

```
PERMIT IZUDFLT.REST.JOBS.COM.IBM.LAB2 CLASS(ZMFAPLA) ID(IZUSVR) ACCESS(READ)
```

3. Refresh the ZMFAPLA class. In RACF, enter the **SETROPTS** command, as follows:

```
SETROPTS RACLIST(ZMFAPLA) REFRESH
```

**Note:** These settings do not override the security settings in your browser. For more information about cross-origin resource sharing (CORS), see the documentation for the web browser.

## Application Linking Manager interface services

To perform traditional system management tasks in z/OS, you might interact with several different interfaces, such as the TSO command line, graphical user interfaces, and web-style interfaces. With the z/OSMF Application Linking Manager, it is possible to link or connect some of these tasks and external applications together for a smoother user experience.

The key components in the Application Linking Manager process include the:

- **Event requestor.** z/OSMF task or external application that requests the launch of a specific function within another task or external application
- **Event.** Action requested by the event requestor. It includes the type of event and the event parameters.
- **Event type.** Object that connects an event requestor to an event handler. It identifies the handlers that can process an event and the possible parameters that can be supplied with an event.
- **Event handler.** z/OSMF task or external application that can process the event parameters and display the requested information.

Figure 1 on page 7 depicts the relationship of these components in the application linking process. For more information about these components and to obtain a list of the predefined event types, requestors, and handlers, see [“Event types, requestors, and handlers shipped with z/OSMF”](#) on page 9.

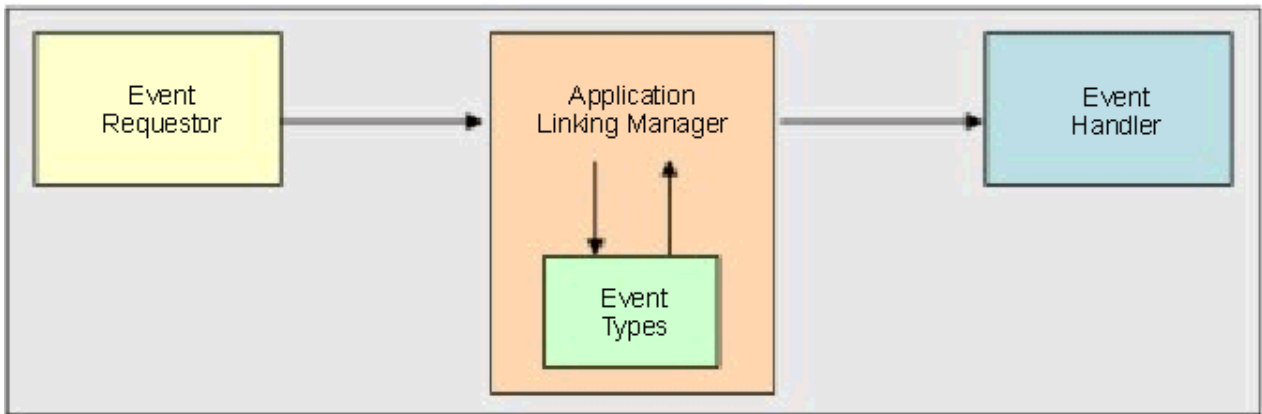


Figure 1. Key components in the application linking process

z/OSMF provides the following resources for working with the Application Linking Manager:

- **Application Linking Manager task**, which provides a graphical user interface that you can use to add, query, or remove event type and event handler definitions.
- **Application Linking Manager REST APIs**, which are a set of REST services that allows a client application to add, query, or remove event type and event handler definitions.
- **AppLinker JavaScript API**, which is a set of JavaScript services that allows a client application to send events to the Application Linking Manager or to define the context to be displayed. The JavaScript services are applicable only if you are creating your own z/OSMF plug-in.

The remainder of this section describes the Application Linking Manager REST APIs. For information about the Application Linking Manager task, see the z/OSMF online help. For details about the AppLinker JavaScript API, see [“Using the Application Linking Manager JavaScript APIs” on page 935](#).

## Operations provided through the Application Linking Manager interface services

Table 2 on page 8 lists the operations that the Application Linking Manager interface services provide.

Table 2. Operations provided through the Application Linking Manager interface services	
Operation	HTTP method and URI path
<a href="#">“Register an event type” on page 23</a>	POST /zosmf/izual/rest/eventtype
<a href="#">“Register an event handler” on page 25</a>	POST /zosmf/izual/rest/handler?eventId=<eventId>
<a href="#">“Obtain a list of all tasks that are eligible to be handlers” on page 28</a>	GET /zosmf/izual/rest/adm/getHandlerEligibleTasks?eventId=<eventId>
<a href="#">“Obtain a list of handlers for an event type” on page 29</a>	GET /zosmf/izual/rest/handler?eventId=<eventId>
<a href="#">“Unregister an event handler” on page 30</a>	DELETE /zosmf/izual/rest/handler/<handlerId>?eventId=<eventId>
<a href="#">“Unregister an event type” on page 31</a>	DELETE /zosmf/izual/rest/eventtype/<eventId>

## Required authorizations

To submit requests through the Application Linking Manager interface services, your user ID requires authorization to the Application Linking Manager task. Ensure that your user ID has READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.ADMINTASKS.APPLINKING. By default, any user ID with z/OSMF administrator authority can access the Application Linking Manager interface services.

Further, if you plan to use the Application Linking Manager interface services to list the registered event handlers for an event type, your user ID requires authorization to the z/OSMF SAF profile prefix on the target z/OS system, as follows:

- READ access to <SAF\_PREFIX> in the APPL class.
- READ access to the <SAF\_PREFIX>.\*.izuUsers profile in the EJBROLE class.

By default, the z/OSMF SAF profile prefix is IZUDFLT.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for sent data and returned data; for the detailed format of each returned object, see the JSON object description for each operation.

## Error handling

For errors that occur during the processing of a request, the Application Linking Manager interface returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the problem or provide it to IBM Support, if required.

The following HTTP status codes are valid:

### HTTP 200 OK

Success.

### HTTP 400 Bad Request

Request contained incorrect parameters.

### HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to the Application Linking Manager task.

### HTTP 404 Bad URL

Target of the request (a URL) was not found.

### HTTP 500 Internal server error

Programming error.

## Error logging

Errors from the Application Linking Manager interface services are logged in the z/OSMF runtime log files or the z/OSMF server logs directory. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Invoking a z/OSMF application link externally

It is possible for an external application (a web application, for example, or a desktop application), to launch a z/OSMF event on a z/OS system. Here, the application issues a command comprised of the URL for the local instance of z/OSMF, combined with the appropriate event information.

The following is an example of such a request. In the example, *eventType-1* identifies the event type and *value-1* specifies the parameter input for the event handler.

---

```
https://...:9443/zosmf/?izual.eventType={eventType-1}&parm1={value-1}
```

---

This command launches z/OSMF in a new browser window on the issuer's system, and sends the event to z/OSMF after session startup. If the user is authenticated to z/OSMF, the handler is launched in the user's session. Otherwise, the user is prompted to authenticate before the handler can be launched.

## Event types, requestors, and handlers shipped with z/OSMF

IBM ships several event types, requestors, and handlers with z/OSMF so that you can quickly start exploiting the application linking capability and easily navigate between multiple z/OSMF tasks.

For more details about these application linking objects, see the following sections:

- [“Event types” on page 10](#)
- [“Event requestors” on page 14](#)
- [“Event handlers” on page 18](#)

## Event types

An *event type* is the intermediary that connects an event requestor to an event handler. Event requestors and event handlers do not interact directly; instead, z/OSMF uses the event type to pass information, the event type and parameters, from the requestor to the handler.

Event handlers can process all, none, or a subset of the parameters that are provided with an event type. When creating an event requestor for an event type, ensure that the event requestor supplies the parameters that are required by the event handlers. To obtain a list of the parameters that are required for the IBM-supplied handlers to process an event type, see [“Event handlers” on page 18](#).

For a list of the event types that are shipped with z/OSMF, see [Table 3 on page 11](#). This table provides the following information for each event type: ID, display name, description, parameters, and the name of the plug-in that registers the event type. By default, these event types are listed in the Application Linking Manager task if the corresponding plug-in is configured in your z/OSMF instance.

**Note:** This table is formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

Table 3. Event types shipped with z/OSMF

Event Type ID	Display Name	Description	Parameters Provided	Registered By
<b>IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW</b>	Create a workflow	Perform a guided set of steps, for example, to configure components or products in your installation.	<p><b>workflow_definition_file_name</b> Fully-qualified UNIX file name or PDS name of the workflow definition file.</p> <p><b>workflow_name</b> Name that the user has provided for the workflow.</p> <p><b>workflow_owner</b> z/OS user id of the workflow owner.</p> <p><b>workflow_target_system</b> Name of the system for this workflow.</p>	Workflows plug-in
<b>IBM.ZOSMF.WORKFLOWS.EDIT_WORKFLOW</b>	Edit a Workflow Definition	Launches the workflow editor for a specified workflow definition file when context is provided. Otherwise, a dialog is displayed to prompt the user to select a workflow definition file for editing.	<p><b>workflow_definition_file_name</b> Fully-qualified UNIX file name of the workflow definition file.</p> <p><b>variable_input_file_name</b> Fully qualified UNIX file name of the workflow variable input properties file.</p> <p><b>target_step_name</b> Name of a step in the workflow definition file to be displayed on initial launch of the workflow editor.</p> <p><b>launch_read_only</b> A value of "true" indicates the workflow editor is launched in read-only mode. Modifications to the workflow definition and related files are not allowed.</p>	Workflow plug-in
<b>IBM.ZOSMF.CONFIGURE_NETWORK_POLICIES</b>	Configure network policies	Configure z/OS Communications Server network policies.	No parameters are provided for this event type.	Network Configuration Assistant plug-in
<b>IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION</b>	View Active Service Definition	View the active service definition.	<p><b>sysplex</b> Name of the sysplex for which to display the active service definition.</p> <p><b>timestamp</b> Timestamp in milliseconds when the service definition was active.</p>	Workload Management plug-in

Table 3. Event types shipped with z/OSMF (continued)

Event Type ID	Display Name	Description	Parameters Provided	Registered By
<b>IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.REPORT_CLASS</b>	View Report Class of Active Service Definition	View the report classes that are contained in the active service definition.	<b>sysplex</b> Name of the sysplex for which to display the active service definition. <b>timestamp</b> Timestamp in milliseconds when the service definition was active. <b>reportClass</b> Name of the report class to be viewed.	Workload Management plug-in
<b>IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS</b>	View Service Class of Active Service Definition	View the service classes that are contained in the active service definition.	<b>sysplex</b> Name of the sysplex for which to display the active service definition. <b>timestamp</b> Timestamp in milliseconds when the service definition was active. <b>serviceClass</b> Name of the service class to be viewed. <b>period</b> Period in the service class to be viewed.	Workload Management plug-in
<b>IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.WORKLOAD</b>	View Workload of Active Service Definition	View the workloads that are contained in the active service definition.	<b>sysplex</b> Name of the sysplex for which to display the active service definition. <b>timestamp</b> Timestamp in milliseconds when the service definition was active. <b>workload</b> Name of the workload to be viewed.	Workload Management plug-in
<b>IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_POLICY</b>	View Active Service Policy	View the active service policy.	<b>sysplex</b> Name of the sysplex for which to display the active service policy. <b>timestamp</b> Timestamp in milliseconds when the service policy was active.	Workload Management plug-in
<b>IBM.ZOSMF.VIEW_DATASET</b>	View Data Set	View or browse a data set.	<b>dataSetName</b> Name of the dataset to be viewed.	ISPF plug-in

Table 3. Event types shipped with z/OSMF (continued)

Event Type ID	Display Name	Description	Parameters Provided	Registered By
<b>IBM.ZOSMF.VIEW_JOB_STATUS</b>	View Job Status	View the status of a job.	<b>jobName</b> Name of the job for which to display status. <b>jobId</b> ID of the job for which to display status.	ISPF plug-in
<b>IBM.ZOSMF.VIEW_SYSPLEX_PERF</b>	View Sysplex Performance	View the overall performance of a sysplex.	<b>sysplex</b> Name of the sysplex for which to display the performance.	Resource Monitoring plug-in
<b>IBM.ZOSMF.VIEW_SYSPLEX_PERF_INDEX</b>	View Performance Index Details	View the performance index of a sysplex.	<b>sysplex</b> Name of the sysplex for which to display the performance index.	Resource Monitoring plug-in
<b>IBM.ZOSMF.VIEW_WLM_REPORT_CLASS_PERF</b>	View Report Class Performance	View the execution velocity and response time metrics for the report classes in a sysplex.	<b>sysplex</b> Name of the sysplex for which to display performance metrics. <b>reportClass</b> Name of the report class for which to display metrics.	Resource Monitoring plug-in
<b>IBM.ZOSMF.VIEW_WLM_SERVICE_CLASS_PERF</b>	View Service Class Performance	View the performance of active service class periods in a sysplex.	<b>sysplex</b> Name of the sysplex for which to display performance metrics. <b>serviceClass</b> Name of the service class for which to display metrics. <b>period</b> Period in the service class for which to display metrics.	Resource Monitoring plug-in
<b>IBM.ZOSMF.VIEW_WLM_STATUS</b>	View WLM Status	Display the status of WLM in the sysplex.	<b>sysplex</b> Name of the sysplex for which to display the status of WLM.	Workload Management plug-in
<b>IBM.ZOSMF.VIEW_WLM_WORKLOAD_PERF</b>	View Workload Performance	View the execution velocity and response time metrics for the workloads in a sysplex.	<b>sysplex</b> Name of the sysplex for which to display performance metrics. <b>workload</b> Name of the workload for which to display metrics.	Resource Monitoring plug-in

## Event requestors

An *event requestor* provides a user-interface control that when invoked passes an event type and the required parameters to the Application Linking Manager task so that they can be passed to an appropriate handler. [Table 4 on page 15](#) lists the name and description of the event requestors that are shipped with z/OSMF, and it provides a list of the event types the requestor can invoke, the user-interface (UI) control you can use to invoke the event type, and the IBM-supplied handlers for the event type.

When creating an event handler for the IBM-supplied event types, you can use the information provided in [Table 4 on page 15](#) to invoke the event type and verify that your handler is displaying the expected output.

### Notes:

1. The UI controls listed in [Table 4 on page 15](#) are available only if the corresponding event type is registered in z/OSMF, and one or more handlers are registered for the event type and are available to process the request. Otherwise, the UI control is disabled or is not provided in the user interface.
2. [Table 4 on page 15](#) is formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

Table 4. Event requestors shipped with z/OSMF				
Event Requestor	Description	Invoked Event Type	UI Controls that Invoke the Event Type	Event Handler
<b>Application Linking Manager task</b>	Create context-sensitive launch points between z/OSMF tasks or external applications.	IBM.ZOSMF.IMPORT_EXTERNAL_APP	<ul style="list-style-type: none"> <li><b>Import</b> action provided in the <b>Event Types</b> table.</li> <li><b>Import</b> action provided in the <b>Handlers</b> table.</li> </ul>	Import Manager task
<b>Incident Log task</b>	Diagnose system problems, and send diagnostic data to IBM or other vendors for further diagnostics.	IBM.ZOSMF.VIEW_DATASET	<ul style="list-style-type: none"> <li><b>View Log</b> button on the Diagnostic Data tab on the View Diagnostic Details page.</li> <li>Link in the <b>Source Name</b> column in the <b>Diagnostic Data</b> table on the Diagnostic Data tab.</li> </ul>	ISPF task
<b>Incident Log task</b>	Diagnose system problems, and send diagnostic data to IBM or other vendors for further diagnostics.	IBM.ZOSMF.VIEW_JOB_STATUS	<b>View Job Details</b> action provided in the <b>FTP Job Status</b> table on the FTP Job Status page.	By default, no handler is provided for this event type.
<b>Links task</b>	Add links to external sites for system management tools and information.	IBM.ZOSMF.IMPORT_EXTERNAL_APP	<b>New</b> action provided in the Links page.	Import Manager task
<b>Resource Monitoring task</b>	Monitor the performance of the z/OS, AIX®, Linux®, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS	<b>View WLM Service Class</b> action provided in the bar chart for metrics that are organized or filtered by WLM service class or WLM service class period.	Workload Management task
<b>Resource Monitoring task</b>	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.WORKLOAD	<b>View WLM Workload</b> action provided in the bar chart for metrics that are organized or filtered by WLM workload.	Workload Management task
<b>Resource Monitoring task</b>	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.REPORT_CLASS	<b>View WLM Report Class</b> action provided in the bar chart for metrics that are organized or filtered by WLM report class or WLM report class period.	Workload Management task
<b>Resource Monitoring task</b>	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_SYSPLEX_PERF	<b>System Status</b> link provided in the Select Sysplex window.	System Status task
<b>System Status task</b>	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_SYSPLEX_PERF_INDEX	<ul style="list-style-type: none"> <li><b>View &gt; Performance Index Details</b> action provided in the <b>Resources</b> table on the System Status page.</li> <li>Link in the <b>Performance Index Status</b> column in the <b>Resources</b> table on the System Status page.</li> </ul>	Resource Monitoring task

Table 4. Event requestors shipped with z/OSMF (continued)

Event Requestor	Description	Invoked Event Type	UI Controls that Invoke the Event Type	Event Handler
<b>System Status task</b>	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION	<ul style="list-style-type: none"> <li>• <b>View &gt; Active WLM Service Definition</b> action provided in the <b>Resources</b> table on the System Status page.</li> <li>• Link in the <b>Related Service Definition</b> column in the <b>Resources</b> table on the System Status page.</li> </ul>	Workload Management task
<b>System Status task</b>	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_POLICY	<ul style="list-style-type: none"> <li>• <b>View &gt; Active WLM Policy</b> action provided in the <b>Resources</b> table on the System Status page.</li> <li>• Link in the <b>Active WLM Policy</b> column in the <b>Resources</b> table on the System Status page.</li> </ul>	Workload Management task
<b>System Status task</b>	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_WLM_STATUS	<b>View &gt; WLM Status</b> action provided in the <b>Resources</b> table on the System Status page.	Workload Management task
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_SYSPLEX_PERF	<b>View performance of systems</b> link provided on the WLM Status page.	System Status task
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_WLM_SERVICE_CLASS_PERF	<ul style="list-style-type: none"> <li>• <b>View performance of active policy</b> link provided on the WLM Status page.</li> <li>• <b>View Performance of Active Policy</b> action provided in the table on the Service Policies for Sysplex page.</li> <li>• <b>View Performance of Active Policy</b> action provided in the Service Policies table in a View or Modify tab.</li> <li>• <b>View Performance of Selected and View Performance of All</b> actions provided in the Service Classes and Service Class Overrides tables in a View or Modify tab.</li> </ul>	Resource Monitoring task
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_WLM_WORKLOAD_PERF	<b>View Performance of Selected and View Performance of All</b> actions provided in the Workloads table in a View or Modify tab.	Resource Monitoring task

Table 4. Event requestors shipped with z/OSMF (continued)				
Event Requestor	Description	Invoked Event Type	UI Controls that Invoke the Event Type	Event Handler
Workload Management task	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMFVIEW_WLM_REPORT_CLASS_PERF	View Performance of Selected and View Performance of All actions provided in the Report Classes table in a View or Modify tab.	Resource Monitoring task

## Event handlers

An *event handler* is a z/OSMF task or external application that can handle requests sent by event requestors. Event handlers support specific event types and can support all, none, or a subset of the parameters that are supplied with an event type. For a handler to process a request, it must receive the correct parameters from the event requestor.

For a list of the event types and parameters (optional and required) that are supported by the handlers shipped with z/OSMF, see [Table 5 on page 20](#). This table also provides a description of each handler and the expected output for each event type and handler combination.

**Note:** This table is formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.



Table 5. Event handlers shipped with z/OSMF					
Event Handler	Description	Supported Event Type	Required Parameters	Optional Parameters	Expected Output
Network Configuration Assistant task	Configure TCP/IP policy-based networking functions.	IBM.ZOSMF.CONFIGURE_NETWORK_POLICIES	None	None	Displays the main page in the Network Configuration Assistant task.
Import Manager task	Add installation-specific function to z/OSMF in the form of plug-ins.	IBM.ZOSMF.IMPORT_EXTERNAL_APP	tab	None	Opens to the Import Manager task.
ISPF task	Access traditional ISPF applications.	IBM.ZOSMF.VIEW_DATASET	dataSetName	None	Displays the data set.
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_SYSPLEX_PERF_INDEX	None	<b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex. <ul style="list-style-type: none"><li>Important service class periods</li><li>All service class periods</li><li>Report class periods</li></ul>	Opens a dashboard that contains the performance index for the following items: <ul style="list-style-type: none"><li>Important service class periods</li><li>All service class periods</li><li>Report class periods</li></ul>
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_WLM_REPORT_CLASS_PERF	None	<b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex. <b>reportClass</b> If unspecified, the metrics for all report classes are displayed.	Opens a dashboard that contains the following metrics for each WLM report class: <ul style="list-style-type: none"><li>Execution velocity</li><li>Response time</li></ul>
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_WLM_SERVICE_CLASS_PERF	None	<b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex. <b>serviceClass</b> If unspecified, the metrics for all service classes are displayed. <b>period</b> If unspecified, the metrics for all periods in the specified service class are displayed. This value is used only if a service class is specified.	Opens a dashboard that contains the following metrics for each WLM service class period: <ul style="list-style-type: none"><li>Performance index</li><li>Execution velocity</li><li>Execution velocity goal</li><li>Response time</li><li>Response time goal</li></ul>
Resource Monitoring task	Monitor the performance of the z/OS, AIX, Linux, and Windows systems in your enterprise.	IBM.ZOSMF.VIEW_WLM_WORKLOAD_PERF	None	<b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex. <b>workload</b> If unspecified, the metrics for all workloads are displayed.	Opens a dashboard that contains the following metrics for each WLM workload: <ul style="list-style-type: none"><li>Execution velocity</li><li>Response time</li></ul>
System Status task	Quickly assess the workload performance on the systems in your enterprise, and define the systems to be monitored.	IBM.ZOSMF.VIEW_SYSPLEX_PERF	None	<b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.	Displays a list of the sysplexes that are defined in the System Status task.

Table 5. Event handlers shipped with z/OSMF (continued)					
Event Handler	Description	Supported Event Type	Required Parameters	Optional Parameters	Expected Output
<b>Workflows task</b>	Perform a guided set of steps, for example, to configure components or products in your installation.	IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW	<p><b>workflow_definition_file_name</b> Fully-qualified UNIX file name or PDS name of the workflow definition file.</p> <p><b>workflow_name</b> Name that the user has provided for the workflow.</p> <p><b>workflow_owner</b> z/OS user id of the workflow owner.</p> <p><b>workflow_target_system</b> Name of the system for this workflow.</p>	<p><b>workflow_callback</b> Callback URL to use when the workflow is created. For example, the caller can be notified with the name of workflow, to be used for a subsequent invocation to launch the workflow.</p> <p><b>workflow_comments</b> Comments to be added to the workflow on creation.</p>	Opens to the <i>Workflow Details</i> panel for the newly created workflow in the Workflows task.
<b>z/OSMF Workflow Editor Handler</b>	Provide a set of tools for viewing and editing a workflow definition.	IBM.ZOSMF.WORKFLOWS.EDIT_WORKFLOW	None	<p><b>workflow_definition_file_name</b> Fully-qualified UNIX file name of the workflow definition file.</p> <p><b>variable_input_file_name</b> Fully qualified UNIX file name of the workflow variable input properties file.</p> <p><b>target_step_name</b> Name of a step in the workflow definition file to be displayed on initial launch of the workflow editor.</p> <p><b>launch_read_only</b> A value of "true" indicates the workflow editor is launched in read-only mode. Modifications to the workflow definition and related files are not allowed.</p>	Opens to the workflow editor main page for a specified workflow definition file, or a dialog that prompts the user to select a workflow definition file for editing.
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION	None	<p><b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.</p> <p><b>timestamp</b> If unspecified, the service definition that is currently active is displayed.</p>	Displays the service definition that is active in the sysplex.
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.REPORT_CLASS	None	<p><b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.</p> <p><b>timestamp</b> If unspecified, the service definition that is currently active is displayed.</p> <p><b>reportClass</b> If unspecified, all the report classes in the service definition are displayed.</p>	Displays the specified report class or a list of all the report classes defined in the active service definition.

Table 5. Event handlers shipped with z/OSMF (continued)					
Event Handler	Description	Supported Event Type	Required Parameters	Optional Parameters	Expected Output
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS	None	<p><b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.</p> <p><b>timestamp</b> If unspecified, the service definition that is currently active is displayed.</p> <p><b>serviceClass</b> If unspecified, all the service classes in the service definition are displayed.</p> <p><b>period</b> If unspecified, all the periods in the service class are displayed. This value is used only if a service class is specified.</p>	Displays the specified service class and period, or a list of all the service classes defined in the active service definition.
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.WORKLOAD	None	<p><b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.</p> <p><b>timestamp</b> If unspecified, the service definition that is currently active is displayed.</p> <p><b>workload</b> If unspecified, all the workloads in the service definition are displayed.</p>	Displays the specified workload or a list of all the workloads defined in the active service definition.
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_POLICY	None	<p><b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.</p> <p><b>timestamp</b> If unspecified, the service policy that is currently active is displayed.</p>	Displays the service policy that is active in the sysplex.
<b>Workload Management task</b>	Administer and operate WLM, and manage WLM service definitions and policies.	IBM.ZOSMF.VIEW_WLM_STATUS	None	<p><b>sysplex</b> If unspecified, the default value is the z/OSMF host sysplex.</p>	Opens the WLM Status tab.

## Register an event type

You can use this operation to define a new event type to z/OSMF.

### HTTP method and URI path

---

```
POST /zosmf/izual/rest/eventtype
```

---

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **eventtype** identifies the event type component of the application linking process.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

Your request must include a JSON object that describes the event type to be registered, for example:

```
{
  id: "IBM.ZOSMF.EVENT_TYPE_ID",
  displayName: "Default English name",
  desc: "Default English description",
  owner: "ownerId",
  params: {
    "key1": "English description of the parameter.",
    "key2": "English description of the parameter."
  }
}
```

Figure 2. Registering an event type: request content

The following values are supported:

#### id

Specify a unique identifier for the event type. It can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (\_). The event ID is required and must be unique.

It is recommended that IDs have the format *company-name.product-name.event-name* where:

- *company-name* is the name of your company. Use a period as the delimiter within the company name. For example, for Tivoli® products, the company name can be *IBM.TIVOLI*.
- *product-name* is the name of the product for which the event type is being created.
- *event-name* is the action that will be completed by the event handler. The event name should start with a verb that reflects this action. Use an underscore as the delimiter within the event name.

For example, to create an event type that allows a user to view the status of a job listed in z/OSMF, the event name portion of the ID can be *VIEW\_JOB\_STATUS*. The entire ID can be *IBM.ZOSMF.VIEW\_JOB\_STATUS*.

#### displayName

Specify the name of the event type. The name is required and can contain up to 50 characters.

**Example:** *View job status.*

#### desc

Specify a description of the event type. The description is optional and can contain up to 200 characters.

**Example:** *Use this event type to view the status of a job. This event type is invoked when a user selects the View Job Status action.*

#### owner

Specify the ID of the first z/OSMF task or external application that registered the event type. Typically, event types are registered by or on behalf of event handlers. They can also be registered by or on behalf of event requestors. This field is required, and can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (\_).

#### params

Specify the name and description of each parameter that event requestors can supply with an event. Enter each parameter name and description combination on a separate line, and enclose the entry in quotes. Use a colon to separate the parameter name and description, and a comma to end each entry. The final entry is not ended with a delimiter.

This area can contain up to 4,000 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), underscores (\_), and commas (,).

For example, to allow event requestors to provide a job ID, job name, and user ID for an event type that displays the status of a job, you would specify the following parameters:

```
params: {  
  "jobID": "ID assigned to the job.",  
  "jobName": "Name specified for the job.",  
  "userID": "ID of the user who submitted the job."  
}
```

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3.](#)

## Required authorizations

See [“Required authorizations” on page 8.](#)

## IBM-supplied event types

z/OSMF includes a number of predefined event types, requestors, and handlers. For a list, see [“Event types, requestors, and handlers shipped with z/OSMF” on page 9.](#)

## Example of registering an event type

A sample request to register an event type is shown in [Figure 3 on page 24.](#)

```
POST /zosmf/izual/rest/eventtype HTTP/1.1  
Host: zosmf1.yourco.com  
  
Accept: application/json  
Content-Type: application/json  
  
{  
  "id": "IBM.ZOSMF.VIEW_JOB_STATUS",  
  "displayName": "View Job Status",  
  "owner": "SDSF",  
  "params": { "jobName": "Name of the job for which to view status." }  
}
```

*Figure 3. Example: Registering an event type*

## Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 9](#).

The response also includes a JSON object with additional information about the results of the request. If your request is successful, the JSON object contains null data for the error and result fields, as shown in [Figure 4 on page 25](#).

```
{"error":null,"result":null}
```

*Figure 4. Example: Returned results of a successful event registration*

For an unsuccessful request, the JSON object contains an error message in the error fields, msgid and msgtxt. The example in [Figure 5 on page 25](#) shows the results for an attempt to register an already-registered event with different parameters:

```
{
  "error": {
    "msgid": "IZUG690E",
    "msgtxt": "Event type \"IBM.ZOSMF.VIEW_JOB_STATUS\" is already defined, but
              different parameters are specified."},
  "result": null
}
```

*Figure 5. Example: Returned results of an unsuccessful event registration*

## Register an event handler

You can use this operation to define a new event handler to z/OSMF.

### HTTP method and URI path

```
POST /zosmf/izual/rest/handler?eventTypeId=<eventTypeId>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **handler** identifies the event handler component of the application linking process.
- **eventTypeId=<eventTypeId>** is the event type to be associated with the new handler.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

Your request must include a JSON object that describes the event handler to be registered.

The following values are supported:

**type**

Handler type. For a z/OSMF plug-in, specify INTERNAL. For an external application, specify EXTERNAL.

**id**

Unique identifier for a launch point within the handler task or application. It can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (\_). The handler ID is required and must be unique.

For external applications, you can specify any value. To ensure uniqueness, it is recommended that you make the company name and the application name part of the handler ID. For example, *IBM.TIVOLI.OMEGAMON*.

For applications within the ISPF task, it is recommended that you prefix the handler ID with IBM.ISPF. Then, specify an ID for the application that will handle the events. For example, *IBM.ISPF.SDSF.ST*.

**applId**

Identifier assigned to the z/OSMF plug-in that contains the task. It is required for a z/OSMF task (type is set to INTERNAL). Omit this value if the handler is an external application.

Table 6 on page 26 lists the valid applId values for the z/OSMF tasks.

Table 6. Valid applId values for the z/OSMF plug-ins	
z/OSMF Task	applId Value
Network Configuration Assistant task	CAV1R11
Import Manager task	IzuImportManager
ISPF task	com.ibm.zosmf.ispf
Resource Monitoring task	IZUR
System Status task	IZUR
Workflows task	workflow
Workload Management task	IZUW

**taskId**

Identifier assigned to the z/OSMF task. It can contain up to 50 characters, including alphanumeric characters (A-Z, a-z, and 0-9), periods (.), and underscores (\_). The task ID is required when type is set to INTERNAL. Omit this value if the handler is an external application.

Table 7 on page 26 lists the valid task ID values for the z/OSMF tasks.

Table 7. Valid taskId values for the z/OSMF tasks	
z/OSMF Task	taskId Value
Network Configuration Assistant task	Configuration Assistant
Import Manager task	IZUG_TASK_zOSMFImportManager
ISPF task	ISPF
Resource Monitoring task	IZUR_PERFDESKS_TASK_ID
System Status task	IZUR_OVERVIEW_TASK_ID
Workflows task	Workflows
Workload Management task	Workload Management

**displayName**

For the handler name, specify the name of the handler task or application. The name is required and can contain up to 50 characters. For z/OSMF tasks, it is recommended that you use the same name displayed in the z/OSMF desktop. For example, *Workload Management*.

For external applications, it is recommended that you use the name of the product or application. For example, *Omegamon*.

**url**

URL to be used for accessing the handler. The URL can contain up to 4,000 characters, including alphanumeric characters (A-Z, a-z, 0-9), blanks, mathematical symbols (+ - = | ~ ( ) { } \), punctuation marks ( ? , . ! ; : ' " / [ ] ), and the following special characters: %, \$, #, @, ^, \*, and \_. The URL is required and must be URI-encoded as specified in RFC 2396. For more information about RFC 2396, see the [Uniform Resource Identifiers \(URI\): Generic Syntax \(www.ietf.org/rfc/rfc2396.txt\)](http://www.ietf.org/rfc/rfc2396.txt) web page.

For a z/OSMF task, specify a URL that is relative to the z/OSMF instance. That is, the URL must begin with /zosmf/. For an external application, specify the full URL, including the protocol.

**options**

The CONTEXT\_SUPPORT option indicates what the handler will display when it processes events of this type. Specify one of the following values for CONTEXT\_SUPPORT:

**OPT\_CONTEXT\_SUPPORT\_NONE**

Handler is launched without context. That is, its homepage is displayed. If the handler is already open, it receives focus, but the context is not updated.

If the handler is an external application, it opens in a separate window. Otherwise, the handler opens in a new z/OSMF task tab.

This option is selected by default.

**OPT\_CONTEXT\_SUPPORT\_LAUNCH**

Handler is launched with context. If the handler is already open, it receives focus, but the context is not updated.

If the handler is an external application, it opens in a separate window. Otherwise, the handler opens in a new z/OSMF task tab.

**OPT\_CONTEXT\_SUPPORT\_LAUNCH\_AND\_RELOAD**

Handler is launched with context. If the handler is already open, a message is displayed warning the user that the current context will be overwritten. This option is supported only when the event requestor and handler are z/OSMF tasks.

**OPT\_CONTEXT\_SUPPORT\_LAUNCH\_AND\_SWITCH**

Handler is launched with the context it specified when subscribing to the event type. This option is supported only when the event requestor and handler are z/OSMF tasks.

**Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

**Required authorizations**

See [“Required authorizations” on page 8](#).

**Expected response**

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 9](#).

## IBM-supplied event handler registrations

z/OSMF includes a number of predefined event types, requestors, and handlers. For a list, see [“Event types, requestors, and handlers shipped with z/OSMF” on page 9](#).

## Obtain a list of all tasks that are eligible to be handlers

The *handlerEligible* property indicates whether a z/OSMF task can participate in the application linking process as an event handler. To obtain a list of the tasks with the *handlerEligible* property set to *true*, use the GET method.

### HTTP method and URI path

---

```
GET /zosmf/izual/rest/adm/getHandlerEligibleTasks?eventId=<eventId>
```

---

where:

- **/zosmf/izual/rest/adm** identifies the Application Linking Manager interface.
- **getHandlerEligibleTasks** indicates that the service will retrieve a list of tasks that are eligible to be event handlers.
- **eventId=<eventId>** is the event type for which the request is being submitted.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### Required authorizations

See [“Required authorizations” on page 8](#).

### Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 9](#).

The response also includes a JSON object with additional information about the results of the request. If your request is successful, the JSON object contains null data for the error field, and the result field lists the task ID, navigation URL, display name, and plug-in ID for each task that is eligible to be a handler. [Figure 6 on page 29](#) provides a sample response for a successful request.

```

HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"error":null,"result":{"Task":[
{"TaskID":"test2","navigationUri":"\\zosmf\\test2","displayName":"test2","PluginID":"TestPlugin1"},
{"TaskID":"test3","navigationUri":"\\zosmf\\test3","displayName":"test3","PluginID":"TestPlugin1"},
{"TaskID":"test4","navigationUri":"\\zosmf\\test4","displayName":"test4","PluginID":"TestPlugin1"},
{"TaskID":"test1","navigationUri":"\\zosmf\\test1","displayName":"Test1","PluginID":"TestPlugin2"}
]}}
```

Figure 6. Sample response from a successful list tasks request

For an unsuccessful request, the JSON object contains an error message in the error fields, msgid and msgtext.

## Obtain a list of handlers for an event type

You can use this operation to obtain a list of registered handlers for an event type.

### HTTP method and URI path

```
GET /zosmf/izual/rest/handler?eventType=<eventType>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **handler** identifies the event handler component of the application linking process.
- **eventType=<eventType>** is the ID of the event type for which you want to obtain a list of registered handlers.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

### Required authorizations

See [“Required authorizations”](#) on page 8.

### Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling”](#) on page 9.

The response also includes a JSON object with additional information about the results of the request. If your request is successful, the JSON object contains null data for the error field, as shown in [Figure 7 on page 30](#), [Figure 8 on page 30](#), and [Figure 9 on page 30](#).

If the request is successful and one or more handlers are enabled for the event type, information about the registered handlers are returned, as depicted in [Figure 7 on page 30](#).

```
{ "result": [{ "id": "IBM.ZOSMF.IZU_IMPORT_HANDLER", "taskId": "IZUG_TASK_zOSMFImportManager",  
  "enabled": true, "defaultHandler": false, "applId": "IzuImportManager", "type": "INTERNAL",  
  "displayName": "Import Manager", "url": "\/zosmf\/IzuImportUtility\/index.jsp",  
  "eventType": "IBM.ZOSMF.IMPORT_EXTERNAL_APP",  
  "options": { "CONTEXT_SUPPORT": "OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH" } } ], "error": null }
```

*Figure 7. Example: Handlers enabled for the event type*

If the request is successful and no handlers are registered for the event type, the result field contains null data, as depicted in [Figure 8 on page 30](#).

```
{ "result": null, "error": null }
```

*Figure 8. Example: Returned results of a successful list handlers request*

If the request is successful and all the handlers that are registered for the event type are disabled, the result field contains an empty array, as depicted in [Figure 9 on page 30](#).

```
//Result if all the handlers defined for the event type are disabled.  
{ "result": [], "error": null }
```

*Figure 9. Example: Returned results of a successful list handlers request*

For an unsuccessful request, the JSON object contains an error message in the error fields, msgid and msgtext.

## Unregister an event handler

You can use this operation to remove an existing event handler registration from z/OSMF.

### HTTP method and URI path

```
DELETE /zosmf/izual/rest/handler/<handlerId>?eventType=<eventType>
```

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **handler** identifies the event handler component of the application linking process.
- **<handlerId>** is the ID of the event handler to be removed.
- **eventType=<eventType>** is the ID of the event type for which you want to remove the specified handler. The combination of **<handlerId>** and **<eventType>** identifies the handler registration to be removed.

### Standard headers

None.

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3.](#)

## Required authorizations

See [“Required authorizations” on page 8.](#)

## Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 9.](#)

# Unregister an event type

You can use this operation to remove the definition of an event type from z/OSMF.

## HTTP method and URI path

---

```
DELETE /zosmf/izual/rest/eventtype/<eventTypeId>
```

---

where:

- **/zosmf/izual/rest** identifies the Application Linking Manager interface.
- **eventtype** identifies the event type component of the application linking process.
- **<eventTypeId>** is the ID of the event type to be removed.

## Standard headers

None.

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3.](#)

## Required authorizations

See [“Required authorizations” on page 8.](#)

## Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. See [“Error handling” on page 9](#).

## Application server routing services

The application server routing services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. Use these services to route requests and responses between the client-side and server-side code for any z/OSMF plug-ins you created where the server-side code is hosted on an application server other than the z/OSMF server.

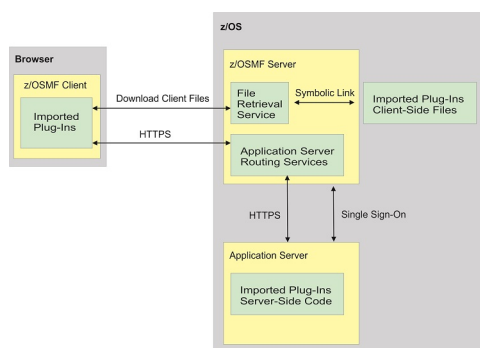


Figure 10. Process for routing requests and responses between application servers

As depicted in Figure 10 on page 32, the process of routing requests and responses for plug-ins where the client and server-side code are on different application servers is as follows:

1. Use the z/OSMF Import Manager task to import your plug-in into z/OSMF, and to associate each task in the plug-in with an application server. For instructions, see [“Adding your applications to z/OSMF” on page 963](#).
2. During the import process, z/OSMF core creates symbolic links to the client-side code for your application and stores those links in the z/OSMF file system on the z/OSMF server.
3. When the import process completes, z/OSMF core adds the tasks included in your plug-in to the z/OSMF desktop.
4. When a user selects your task in the z/OSMF desktop, z/OSMF core submits an HTTPS request to the file retrieval service to retrieve the client-side files and the browser downloads those files.
5. When the user performs an action that requires the task to interact with the server-side code, the client-side code for the task submits an HTTPS request to the application server routing interface, and that interface routes the request to the application server that is associated with the task.
6. The server-side code for your task processes the request and submits an HTTPS response to the application server routing interface, and that interface routes the response to the client.

The z/OSMF server and the application server that hosts the server-side code for your plug-in (referred to as the *target application server*) can reside on the same system or on different systems. To enable single sign-on between the servers, ensure that the servers share the same Lightweight Third Party Authentication (LTPA) key files.

## Operations provided through the application server routing services

[Table 8 on page 33](#) lists the operations that the application server routing services provide.

Table 8. Operations provided through the application server routing services.

Operation	HTTP method and URI path
<b>“Retrieve data from an application server” on page 35</b>	GET /zosmf/externalgateway/system?content=<http-content>
<b>“Update data for an application server” on page 39</b>	POST /zosmf/externalgateway/system PUT /zosmf/externalgateway/system
<b>“Delete data from an application server” on page 42</b>	DELETE /zosmf/externalgateway/system?content=<http-content>

## Required authorizations

The user must be logged into z/OSMF. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for HTTP response data. If the client requests for the application server routing service to add additional information to the response from the target application server, the service wraps the response in the following JSON object. Otherwise, the service returns only the response from the target application server.

```
{
  "primaryAPIVersion": "primary-API-version",
  "systemsOutput": {
    "systemOutput": "system-output",
    "rc": "return-code",
    "error": { "msgid": "message-ID", "msgtxt": "message-text" },
    "systemName": "system-name"
  },
  "numOfSystems": "total-systems"
}
```

where

### primary-API-version

Version of the application server routing services interface on the z/OSMF server.

### systemsOutput

Contains a set of attributes that provide different information about the response from the target application server.

### system-output

Contains the response from the target application server.

### return-code

Contains the return code provided by the target application server. The return code can be one of the following values:

#### OK

Success.

#### HttpConnectionFailed

The HTTPS connection failed. Typically, this error occurs when the target application server is unavailable or a network error has occurred.

**HttpConnectionTimedOut**

The HTTPS request did not complete in the time allotted.

**CertificateError**

The certificate for the target application server is not trusted.

**InvalidLogin**

The login credentials for the target application server are not valid.

**FailedWithMessage**

The request was successful; however, an internal error occurred on the target application server.

**UnexpectedFailure**

An unexpected error occurred.

**error**

If an error occurred with the request, the error attribute contains the message ID (msgid) and message text (msgtxt) for the message that was issued. Otherwise, this attribute is *null*.

**system-name**

Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the target application server.

**total-systems**

Value is set to *1* because the HTTPS request can be sent to only one application server at a time.

**Error handling**

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

**HTTP 200 OK**

Success.

**HTTP 400 Bad request**

Request contained incorrect parameters.

**HTTP 401 Unauthorized**

Submitter of the request is not authorized to use the service or did not authenticate with z/OSMF, or single sign-on is not enabled between the z/OSMF server and the target application server.

**HTTP 404 Bad URL**

Target of the request (a URL) was not found.

**HTTP 500 Internal server error**

Programming error.

**Error logging**

Errors from the application server routing services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Retrieve data from an application server

You can use this operation to request that z/OSMF route a retrieve data request to the application server where the server-side code for your plug-in resides.

### HTTP method and URI path

```
GET /zosmf/externalgateway/system?content=<http-content>
```

where:

- **zosmf/externalgateway** identifies the application server routing services.
- **system** informs the service that the request will be routed to only one application server.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 9 on page 35](#) lists the parameters that are supported for this request.

**Important:** If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *AppServer#1*, specify *AppServer%231*.

Table 9. Supported input parameters for the application server routing services		
Parameter	Required	Description
<b>target</b>	Yes	<p>Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the application server where the server-side code for your plug-in resides. If the specified system entry does not exist, the request will fail.</p> <p>z/OSMF stores the nickname for the target application server in the window object in the Browser Object Model. To retrieve the nickname, issue the following JavaScript command from your task:</p> <pre>window.frameElement.getAttribute("target")</pre> <p>For example:</p> <pre>postCreate: function() {   var target = window.frameElement.getAttribute("target");   var remoteURL = "/zosmf/externalgateway/system? content=   {'target':'" + target + "', 'resourcePath': '/ testApp'}"; }</pre>
<b>resourcePath</b>	Yes	Path to the service that will process the request.
<b>requestProperties</b>	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
<b>timeout</b>	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.

Table 9. Supported input parameters for the application server routing services (continued)

Parameter	Required	Description
<b>wrapped</b>	No	Indicator of whether the application server routing service will wrap the response from the target application server in a JSON object that contains additional information about the response. Set the parameter to <i>N</i> to obtain only the response provided by the target application server. Otherwise, set the parameter to <i>Y</i> or omit it to obtain the response along with additional information. For more details, see <a href="#">“Content type used for HTTP response data” on page 33</a> .
<b>binary</b>	No	Indicator of whether the response from the target application server is in binary format. Set the parameter to <i>N</i> or omit it if the response is not in binary format. Otherwise, set the parameter to <i>Y</i> if the response is in binary format.
<b>content</b>	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the service that will process the request.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Application server routing services” on page 32](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 34](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 33](#).

## Example 1: Retrieve wrapped data from an application server

To retrieve wrapped data from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
GET /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp"} HTTP/1.1
Host: appname.yourco.com
```

*Figure 11. Sample request to retrieve wrapped data from an application server*

A sample response is shown in [Figure 12 on page 37](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":
    {
      "systemOutput":
        {
          "items":[
            {
              "object-ID":"objectA",
              "attribute1":"value1",
              "attribute2":"value2",
              "attribute3":"value3",
              "attribute4":"value4",
              "attribute5":"value5",
            },
            {
              "object-ID":"objectB",
              "attribute1":"value6",
              "attribute2":"value7",
              "attribute3":"value8",
              "attribute4":"value9",
              "attribute5":"value10",
            }
          ],
          "rc":"Ok",
          "error":null,
          "systemName":"appServer1"
        },
      "numOfSystems":1
    }
}
```

*Figure 12. Sample response for retrieving wrapped data from an application server*

## Example 2: Retrieve unwrapped data from an application server

To retrieve unwrapped data from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
GET /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp",
"wrapped":"N"} HTTP/1.1
Host: appname.yourco.com
```

*Figure 13. Sample request to retrieve unwrapped data from an application server*

A sample response is shown in [Figure 14 on page 38](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close
```

```
{
  "items": [
    {
      "object-ID": "objectA",
      "attribute1": "value1",
      "attribute2": "value2",
      "attribute3": "value3",
      "attribute4": "value4",
      "attribute5": "value5",
    },
    {
      "object-ID": "objectB",
      "attribute1": "value6",
      "attribute2": "value7",
      "attribute3": "value8",
      "attribute4": "value9",
      "attribute5": "value10",
    }
  ]
}
```

*Figure 14. Sample response for retrieving unwrapped data from an application server*

### **Example 3: Retrieve binary data from an application server**

To retrieve binary data from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
GET /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp",
"binary":"Y"} HTTP/1.1

Host: appname.yourco.com
```

*Figure 15. Sample request to retrieve binary data from an application server*

A sample response is shown in [Figure 16 on page 39](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close
```

```
01111011 00001101 00001010 00100000 00100000 00100010 01101001 01101000 01100101 01101101
01110011 00100010 00111010 01011011 00001101 00001010 00100000 00100000 01111011 00001101
00001010 00100000 00100000 00100000 00100000 00100010 01101111 01100010 01101010 01100010
01100011 01110100 00101101 01001001 01000100 00100010 00100010 00111010 00100010 01101111 01100010
01101010 01100101 01100011 01110100 01000001 00100010 00001101 00001010 00100000 00100000
00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001 01100010 01110101
01110100 01100101 00110001 00100010 00111010 00100010 01110110 01100001 01101100 01110101
01100101 00110001 00100010 00101100 00001101 00001010 00100000 00100000 00100000 00100000
00100010 01100001 01110100 01110100 01110010 01101001 01100010 01110101 01110100 01100101
00110010 00100010 00111010 00111010 01110110 01100001 01101100 01110101 01100101 00110010
00100010 00101100 00001101 00001010 00100000 00100000 00100000 00100000 00100010 01100001
01110100 01110100 01110010 01101001 01100010 01110101 01110100 01100101 00110011 00100010
00111010 00100010 01110110 01100001 01101100 01110101 01100101 00110011 00100010 00101100
00001101 00001010 00100000 00100000 00100000 00100000 00100010 01100001 01110100 01110100
01110010 01101001 01100010 01110101 01110100 01100101 00110100 00100010 00111010 00100010
01110110 01100001 01101100 01110101 01100101 00110100 00100010 00101100 00001101 00001010
00100000 00100000 00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001
01100010 01110101 01110100 01100101 00110101 00100010 00111010 00100010 01110110 01100001
01101100 01110101 01100101 00110101 00100010 00101100 00001101 00001010 00100000 00100000
01111101 00101100 00001101 00001010 00100000 00100000 01111011 00001101 00001010 00100000
00100000 00100000 00100000 00100010 01101111 01100010 01101010 01100011 01110100
00101101 01001001 01000100 00100010 00111010 00100010 01101111 01100010 01101010 01100010
01100011 01110100 01000010 00001101 00001010 00001101 00100000 00100000 00100000 00100000
00100010 01100001 01110100 01110100 01110010 01101001 01100010 01110101 01110100 01100101
00110001 00100010 00111010 00100010 01110110 01100001 01101100 01110101 01100101 00110110
00100010 00101100 00001101 00001010 00100000 00100000 00100000 00100000 00100000 00100001
01110100 01110100 01110010 01101001 01100010 01110101 01110100 01100101 00110010 00100010
00111010 00100010 01110110 01100001 01101100 01110101 01100101 00110111 00100010 00101100
00001101 00001010 00100000 00100000 00100000 00100010 01100001 01110100 01110100 01110100
01110010 01101001 01100010 01110101 01110100 01100101 00110011 00100010 00111010 00100010
01110110 01100001 01101100 01110101 01100101 00111000 00100010 00101100 00001101 00001010
00100000 00100000 00100000 00100000 00100010 01100001 01110100 01110100 01110010 01101001
01100010 01110101 01110100 01100101 00110100 00100010 00111010 00100010 01110110 01100001
01101100 01110101 00110101 00100010 00111010 00100010 01110110 01100001 01101100 01110101
01100101 00110001 00110001 00100010 00101100 00001101 00001010 00100000 00100000 01111101
01011101 00001101 00001010 01111101
```

Figure 16. Sample response for retrieving binary data from an application server

## Update data for an application server

You can use this operation to request that z/OSMF route an update data request to the application server where the server-side code for your plug-in resides.

### HTTP method and URI path

```
POST /zosmf/externalgateway/system
PUT /zosmf/externalgateway/system
```

where:

- **zosmf/externalgateway** identifies the application server routing services.
- **system** informs the service that the request will be routed to only one application server.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

Your request must include a JSON object that describes the objects to be created or modified on the target application server. [Table 10 on page 40](#) lists the supported parameters.

Table 10. Supported input parameters for the application server routing services		
Parameter	Required	Description
<b>target</b>	Yes	<p>Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the application server where the server-side code for your plug-in resides. If the specified system entry does not exist, the request will fail.</p> <p>z/OSMF stores the nickname for the target application server in the window object in the Browser Object Model. To retrieve the nickname, issue the following JavaScript command from your task:</p> <pre>window.frameElement.getAttribute("target")</pre> <p>For example:</p> <pre>postCreate: function() {   var target = window.frameElement.getAttribute("target");   var remoteURL = "/zosmf/externalgateway/system? content=   {'target':'" + target + "', 'resourcePath': '/ testApp'}"; }</pre>
<b>resourcePath</b>	Yes	Path to the service that will process the request.
<b>requestProperties</b>	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
<b>timeout</b>	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
<b>wrapped</b>	No	Indicator of whether the application server routing service will wrap the response from the target application server in a JSON object that contains additional information about the response. Set the parameter to <i>N</i> to obtain only the response provided by the target application server. Otherwise, set the parameter to <i>Y</i> or omit it to obtain the response along with additional information. For more details, see <a href="#">“Content type used for HTTP response data” on page 33</a> .
<b>binary</b>	No	Indicator of whether the response from the target application server is in binary format. Set the parameter to <i>N</i> or omit it if the response is not in binary format. Otherwise, set the parameter to <i>Y</i> if the response is in binary format.

Table 10. Supported input parameters for the application server routing services (continued)

Parameter	Required	Description
<b>content</b>	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the service that will process the request.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Application server routing services”](#) on page 32.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 34.

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data”](#) on page 33.

## Example

To add *objectC* on the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
POST /zosmf/externalgateway/system HTTP/1.1
Host: appname.yourco.com
{"target":"appServer1","resourcePath":"/testApp/objectC","content":{"attribute1":"value11","attribute2":"value12","attribute3":"value13","attribute4":"value14","attribute5":"value15"}}
```

Figure 17. Sample request to update data on an application server

A sample response is shown in [Figure 18](#) on page 41.

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":
  {
    "systemOutput":
    {
      "result":"success"
    },
    "rc":"Ok",
    "error":null,
    "systemName":"appServer1"
  },
  "numOfSystems":1
}
```

Figure 18. Sample response for updating data on an application server

## Delete data from an application server

You can use this operation to request that z/OSMF route a delete data request to the application server where the server-side code for your plug-in resides.

### HTTP method and URI path

```
DELETE /zosmf/externalgateway/system?content=<http-content>
```

where:

- **zosmf/externalgateway** identifies the application server routing services.
- **system** informs the service that the request will be routed to only one application server.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 11 on page 42](#) lists the parameters that are supported for this request.

**Important:** If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *AppServer#1*, specify *AppServer%231*.

Table 11. Supported input parameters for the application server routing services

Parameter	Required	Description
<b>target</b>	Yes	<p>Nickname assigned to the system entry in the z/OSMF Systems task that describes the settings required to access the application server where the server-side code for your plug-in resides. If the specified system entry does not exist, the request will fail.</p> <p>z/OSMF stores the nickname for the target application server in the window object in the Browser Object Model. To retrieve the nickname, issue the following JavaScript command from your task:</p> <pre>window.frameElement.getAttribute("target")</pre> <p>For example:</p> <pre>postCreate: function() {   var target = window.frameElement.getAttribute("target");   var remoteURL = "/zosmf/externalgateway/system? content=   {'target':'" + target + "', 'resourcePath': '/ testApp'}"; }</pre>
<b>resourcePath</b>	Yes	Path to the service that will process the request.
<b>requestProperties</b>	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
<b>timeout</b>	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.

Table 11. Supported input parameters for the application server routing services (continued)		
Parameter	Required	Description
<b>wrapped</b>	No	Indicator of whether the application server routing service will wrap the response from the target application server in a JSON object that contains additional information about the response. Set the parameter to <i>N</i> to obtain only the response provided by the target application server. Otherwise, set the parameter to <i>Y</i> or omit it to obtain the response along with additional information. For more details, see <a href="#">“Content type used for HTTP response data”</a> on page 33.
<b>binary</b>	No	Indicator of whether the response from the target application server is in binary format. Set the parameter to <i>N</i> or omit it if the response is not in binary format. Otherwise, set the parameter to <i>Y</i> if the response is in binary format.
<b>content</b>	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the service that will process the request.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Application server routing services”](#) on page 32.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 34.

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data”](#) on page 33.

## Example

To remove *objectA* from the application server identified in system entry *appServer1*, which is defined in the z/OSMF Systems task, submit the following request:

```
DELETE /zosmf/externalgateway/system?content={"target":"appServer1","resourcePath":"/testApp/objectA",
"timeout":"30000"} HTTP/1.1
```

Host: appname.yourco.com

*Figure 19. Sample request to delete data from an application server*

A sample response is shown in [Figure 20 on page 44](#).

```
HTTP/1.1 200 OK
Date: Tue, 28 Apr 2015 05:39:28 +0000GMT
Connection: close
```

```
{
  "primaryAPIVersion":1.0,
  "systemsOutput":
  {
    "systemOutput":
    {
      "result":"success"
    },
    "rc":"Ok",
    "error":null,
    "systemName":"appServer1"
  },
  "numOfSystems":1
}
```

*Figure 20. Sample response for deleting data from an application server*

## Cloud provisioning services

The cloud provisioning services are a set of application programming interfaces (APIs), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to perform software provisioning for IBM Cloud Provisioning and Management for z/OS. This includes creating instances of IBM middleware, such as IBM Customer Information Control System (CICS), IBM Db2, IBM Information Management System (IMS), IBM MQ, and IBM WebSphere Application Server (WAS), and creating middleware resources, such as MQ queues, CICS regions, and Db2 databases. This makes it possible for consumers to quickly provision and deprovision an environment as needed.

### Getting started

The security administrator defines the various roles that are required, such as the domain administrator, network administrator, approvers, and consumers. For more information about setting up security for cloud provisioning, see [Cloud provisioning services](#) in *IBM z/OS Management Facility Configuration Guide*.

Using the Cloud Provisioning tasks, your system programmers and application programmers can perform the following actions:

- System programmers:
  - Define the cloud domain (systems), administrators for the domain, and classes of users (tenants) for the domain.
  - Prepare software services templates, which are used to provision z/OS software. Service providers add templates, associate tenants with the templates, create resource pools for the templates, test the templates, then publish them to make them available for consumers.
- System programmers or application programmers:
  - Provision software from templates, creating software services instances.
  - Manage software services instances.

For an illustration of cloud provisioning, see [Figure 21 on page 45](#).

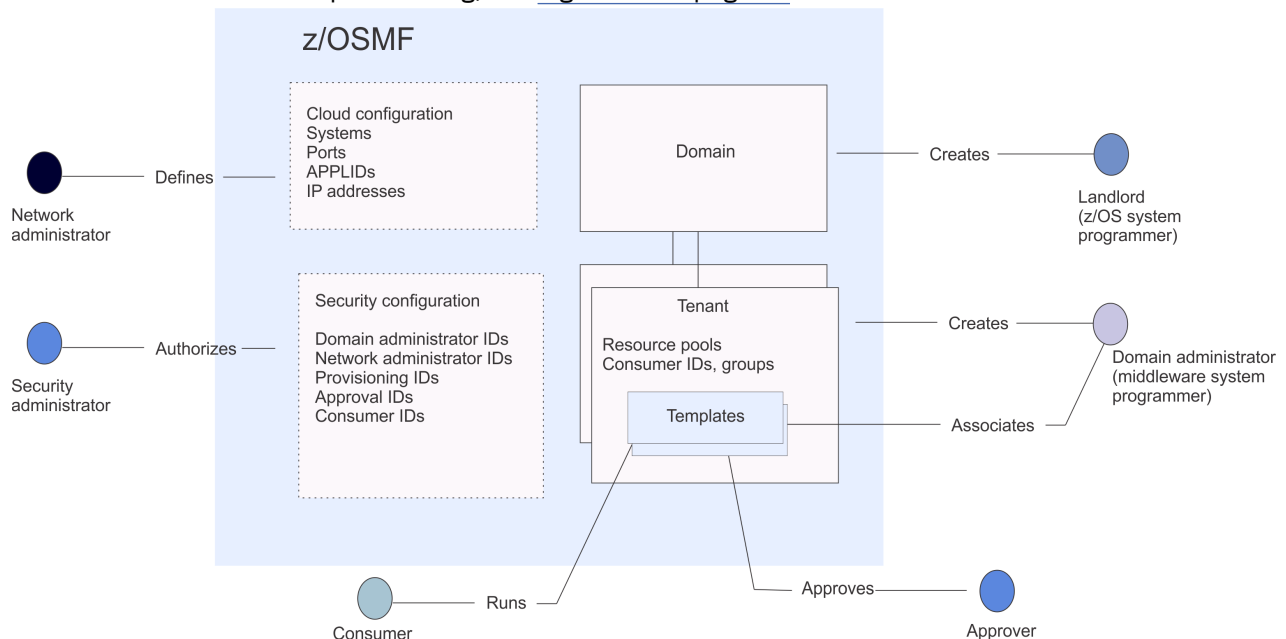


Figure 21. Cloud Provisioning Summary

## Domains define the scope of cloud provisioning

A cloud provisioning domain defines a system or set of systems in the sysplex. The systems in the domain must be included in the group of systems named IYUCLOUD in the **Systems** task of the **z/OSMF Settings** category.

A landlord, typically a z/OS system programmer, decides which system or systems (LPARs) are used for provisioning and creates a domain. If the domain extends beyond one sysplex, the landlord configures a primary z/OSMF system for communicating with secondary z/OSMF systems.

To help you get started quickly, a default domain is provided. The default domain is fully operational without any further configuration, and is accessible to any z/OSMF administrator. A default tenant is associated with the default domain.

When a domain includes more than one system, the domain administrator can specify:

- The systems that are to be used as potential targets for provisioning
- How the target system should be selected when the software service is provisioned: either automatically, by z/OSMF, or manually, by the consumer
- That the instance can be relocated to a system in the domain other than the system it was originally provisioned on. The instance can run on only one system at a time.

## Templates guide provisioning

To make an environment available to consumers as a software service, a domain administrator creates and configures a software services template. The template describes what is provisioned. For example, a template might request that a Db2 subsystem be deployed onto a z/OS system with three databases, or might create a set of CICS regions.

To provision the middleware, templates start and run z/OSMF workflows. A template includes a workflow definition file, along with other files, including a file that defines input variables for the workflow, and a file that defines actions that can be used against the provisioned software.

The template might need to be customized for the installation – for example, to conform with naming standards in your company. You might modify variables that are input to the workflow, or use a properties file that is provided with the template to configure the provisioned software. For information about customization, you typically refer to documentation that is included with the template by the software provider. In addition, the domain administrator:

- Adds the software services template to a tenant.
- Connects the template to network, storage, and WLM resource pools. Resource pools are sets of z/OS resources that are required by the z/OS software service, for example, ports, IP addresses, or APPLIDs.

When a template requires resource pools, for example, when you want to dynamically allocate ports to provisioned software instances, the network and WLM resource pool administrators (typically z/OS system programmers) use the appropriate z/OSMF tasks to complete the resource pools.

Offering self-service provisioning to a development team might require that some steps in the template, or certain actions, run under automation IDs. Any use of these user IDs in a template must be approved. Approval records are created for a template when a workflow or action definition file contains an element that identifies a user ID under which a workflow step or action is to be performed. (The workflow element is `runAsUser` ID, and the ID is sometimes referred to as a `runAsUser` ID). Approval records can also be defined for the template in general, and for a domain. Approval records must be approved by the approvers (typically identified by user ID) before the template can be tested or published.

The domain administrator tests the template to ensure that it successfully provisions the software, that is, creates the environment. Software that is provisioned from a template is known as a software services instance. (Note that this is different than a software instance that you manage with the Software Management task. A *software instance* is a collection of data sets containing installed software, and other data sets that may be associated with that installed software.) You manage a software services instance by using actions such as **Remove** and **deprovision**.

Publishing the template makes it available to consumers in the tenant – the application developers who require the new environment.

## Summary

The terms that you need to understand for provisioning and managing provisioned software are defined here.

## Resources

The following are the key resources in the Cloud Provisioning tasks.

Table 12. Resources for Cloud Provisioning	
Resource	Description
<i>Domain</i>	<p>Defines the management scope for tenants, services, and resource pools.</p> <p>A domain consists of one or more z/OS systems. A domain can include z/OS systems from more than one sysplex.</p> <p>A z/OS system can be in a single domain or in multiple domains that are managed by a single instance of z/OSMF. A cloud domain is defined by a z/OS system programmer who acts as the <i>landlord</i>. Each cloud domain is assigned one or more middleware system programmers who act as domain administrators.</p> <p>A base z/OSMF configuration includes one domain by default – the default domain.</p>
<i>Resource pool</i>	<p>Identifies the z/OS resources that are required by a z/OS software service. In a cloud domain with multiple tenants, the resource pool defines the scope of resource sharing and resource isolation. For example, a resource pool can define a range of dedicated IP addresses or ports for each tenant.</p>
<i>Tenant</i>	<p>Defines the group of users who have the authority to provision software instances.</p> <p>A tenant consists of a user or group of users that have contracted for the use of specified services and pooled z/OS resources that are associated with the services in a domain.</p> <p>A base z/OSMF configuration includes one tenant by default – the default tenant.</p>

## User roles

The following are the key roles in the Cloud Provisioning tasks.

Table 13. User roles for Cloud Provisioning		
Role	Performer	Description
<i>Landlord</i>	z/OS system programmer	Defines the cloud domains and the associated system resources for the cloud. The landlord also designates one or more users as domain administrators.
<i>Domain administrator</i>	Middleware system programmer	Manages a domain. The domain administrator is responsible for defining services, tenants, and resource pools for the domain, and managing the relationship across tenants, services, and resource pools.
<i>Resource pool networking administrator</i>	Network administrator	Manages the resource pool for the networking resources in the cloud, such as network configuration policies.

Table 13. User roles for Cloud Provisioning (continued)		
Role	Performer	Description
<i>Resource pool WLM administrator</i>	Performance administrator	Manages the resource pool for the WLM resources in the cloud, such as WLM policies.
<i>Security administrator</i>	Security administrator	Maintains the installation's security manager, such as RACF.
<i>Template approver</i>	System programmer or security administrator	Responsible for approving the pending approval records that are associated with the template.
<i>Consumer</i>	Application programmer	Has access to the software services and resource pools for a tenant. This user can provision a software services instance by using a software services template, and can manage the lifecycle of a software services instance.

## Objects

The following are some basic objects that you work with in the Cloud Provisioning tasks.

Table 14. Objects for Cloud Provisioning	
Object	Description
<i>Instance, or software services instance</i>	Represents software that is provisioned by using templates.
<i>Template, or software services template</i>	Represents a z/OS middleware or a z/OS middleware resource service. A template consists of workflows and input variables that can be used to provision z/OS software, actions that can be used with the provisioned software (the instance), and documentation.

## Authorization requirements

Use of the cloud provisioning services APIs requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

In addition, the user's z/OS user ID may need access to other resources, including those that define roles. The specific requirements for each cloud provisioning service are described in the topic for that service. For a summary of resources related to roles, see [Table 15 on page 48](#).

Table 15. SAF resources for Cloud Provisioning Roles			
Role	Class	Resources	Access
Landlord	ZMFCLOUD	<SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT.saf_cloud_groupID_prefix	READ
Domain administer	ZMFCLOUD	<SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT.domainGroupID	READ
Domain approver	ZMFCLOUD	<SAF-prefix>.ZOSMF.TEMPLATE.APPROVERS.domainGroupID	READ
Template runAsUser ID	ZMFCLOUD	<SAF-prefix>.ZOSMF.TEMPLATE.RUNASUSERS.domainGroupID.templateID	READ <sup>1</sup>

Table 15. SAF resources for Cloud Provisioning Roles (continued)			
Role	Class	Resources	Access
Template approver	ZMFCLOUD	<SAF-prefix>.ZOSMF.TEMPLATE. APPROVERS.domainGroupID.templateID	READ
Tenant	ZMFCLOUD	<SAF-prefix>.ZOSMF.PROVISIONING. RESOURCE_MANAGEMENT.tenantGroupID	READ
Resource pool network administrator	ZMFCLOUD	<SAF-prefix>.ZOSMF. RESOURCE_POOL.NETWORK.domainGroupID	READ
Resource pool WLM administrator	ZMFCLOUD	<SAF-prefix>.ZOSMF. RESOURCE_POOL.WLM.domainGroupID	READ

<sup>1</sup> Successful READ attempts for the <SAF-prefix>.ZOSMF.TEMPLATE.RUNASUSERS.domainGroupID.templateID resource are audited. Prior to switching identities to the runAsUser user ID, z/OSMF does an authorization check for access to this resource. If the authorization is successful, the runAsUser ID has access and an audit record is generated. If the authorization check fails, no audit record is generated and switching to the runAsUser user ID does not occur. The workflow fails.

For details about security for the cloud provisioning roles, see [Cloud provisioning services in IBM z/OS Management Facility Configuration Guide](#).

For information about how to prepare software for provisioning through the REST APIs or the z/OSMF Cloud Provisioning tasks, including the format of the file for defining actions, see [Preparing software to exploit cloud provisioning in IBM z/OS Management Facility Programming Guide](#).

## Using the Swagger interface

You can use the Swagger interface to display information about the IBM Cloud Provisioning and Management for z/OS REST APIs.

For more information, see [Using the z/OSMF REST services in IBM z/OS Management Facility Programming Guide](#).

## Resource pool services

The resource pool services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to obtain and release network or WLM resources from the network or WLM resource pool that was defined in support of IBM Cloud Provisioning and Management for z/OS. These REST services are intended to be invoked from a workflow step during cloud provisioning and are not available for general use outside of the scope of cloud provisioning.

Table 16 on page 50 lists the operations that the resource pool services provide.

See also:

- [“Get a resource pool” on page 132](#)
- [“Get a domain resource pool” on page 138](#)
- [“List the resource pools” on page 147](#)
- [“List domain resource pools” on page 152](#)

## Resource pool services

Table 16. z/OSMF resource pool services: operations summary	
Operation name	HTTP method and URI path
<a href="#">“Obtain an IP address” on page 51</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/obtain
<a href="#">“Release an IP address” on page 56</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/release
<a href="#">“Obtain a port” on page 59</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/obtain
<a href="#">“Release a port” on page 63</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/release
<a href="#">“Obtain a SNA application name” on page 65</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/obtain
<a href="#">“Release a SNA application name” on page 68</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/release
<a href="#">“Add a classification rule” on page 70</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/add
<a href="#">“Remove a classification rule” on page 72</a>	POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/remove
<a href="#">“Get data set attributes” on page 74</a>	GET /zosmf/resource-mgmt/rest/<version>/rdp/storage/dataset-attr/<tenant-id>/<template-name>/<registry-uuid>

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, a network administrator for APIs related to network resources, or a WLM administrator for APIs related to WLM resources, in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 204 No Content

The request succeeded.

### HTTP 500 Server error

The server encountered an error when it processed the request.

## Obtain an IP address

Use this operation to obtain an IP address from a resource pool that has a configured network resource pool.

### HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/obtain
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation obtains an IP address from a resource pool with a configured network resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in an IP address being obtained.

### Request content

The request content is expected to contain a JSON object that describes the IP address to be obtained. See [Table 17 on page 51](#).

Table 17. Request content for the obtain IP address request			
Field name	Type	Required or optional	Description
registry-uuid	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"><li>Provisioning workflows: \${_workflow-registryID}</li><li>Action workflows: \${_workflow-parentRegistryID}</li></ul> Required when provisioning network resources as part of a composite cluster template.
template-uuid	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
template-name	String	Required	Name of the template that is associated with the resource pool. Derived from workflow internal variable, \${_workflow-templateName}.
tenant-id	String	Required	ID of the tenant that is associated with the resource pool. Derived from workflow internal variable, \${_workflow-tenantID}.

Table 17. Request content for the obtain IP address request (continued)

Field name	Type	Required or optional	Description
<b>network-parms</b>	JSON object	Required	Network. parameters for the request. See <a href="#">Table 18</a> on page 52.

Table 18. Network parameters fields

Field	Type	Required or optional	Description
name	String	Optional	Name used in a panel for the Network Configuration Assistant task to identify who is using the network resource. It is recommended to use workflow internal variable <code>\${_workflow-softwareServiceInstanceName}</code> for this parameter.
usage-type	String	Optional	Used as a filter. If not specified, only network resource pools without a usage type match. If specified, must match the usage type in the network resource pool definition in the Network Configuration Assistant task.
ipaddr	String	Required	IP address. The value can be:  <b>A specific IP address</b> Provision this IP address. The IP address must fit within the available IP address allocation range. Available ranges are associated with the targeted network resource pool, and match the provided usage type.  <b>any4</b> Provision any available Ipv4 address, from the available range.  <b>any6</b> Provision any available IPv6 address, from the available range.
system-name	String	See description	Specifies the target system that the resource will be provisioned on. Required if there is more than one system in the network resource pool. Derived from a workflow internal variable, <code>\${_workflow-systemName}</code> .
deployment-id	String	Optional	Workflow-defined string token, used to group all provisioned resources with a server instance.
host-name	String	Optional	Indicates that the domain name server that is configured to the associated IP address allocation range is to be updated with the IP address and the concatenation of the host name and the zone name from the IP address allocation range. Requires a domain name server object to be configured by the network administrator.

Table 18. Network parameters fields (continued)													
Field	Type	Required or optional	Description										
system-list[]	String	Optional	A list of target systems for provisioning. If you omit this field and the system-name field, all systems in the network resource pool are targets. See <a href="#">Table 20 on page 54</a> for details on the values.										
recovery-method	String	Optional	<p>Defines the availability characteristics of an IP address under high availability recovery situations. The following values are valid:</p> <p><b>MANUAL_DISRUPTIVE</b> Does not allow the IP address to be moved non-disruptively to another TCP/IP stack. Connections to this VIPA are broken if the VIPA is moved. Only one application can simultaneously use this IP address.</p> <p><b>MANUAL_NONDISRUPTIVE</b> Allows the IP address to be moved non-disruptively to another TCP/IP stack. Only one application can simultaneously use this IP address, unless the application-owned field has a value of <code>false</code>.</p> <p><b>DYNAMIC</b> The IP address is distributed and therefore available to multiple equivalent applications simultaneously. There is no need to move the IP address for recovery situations.</p> <p>The default value is dependent upon the deployment type of resource pool, represented by the Workload Deployment Type attribute of the Network Resource Pool in Network Configuration Assistant.</p> <table><tr><th colspan="2">Table 19. Default values for recovery-method</th></tr><tr><th>Workload Deployment Type</th><th>recovery-method Default</th></tr><tr><td>SINGLE_SYSTEM</td><td>MANUAL_DISRUPTIVE</td></tr><tr><td>MOVABLE</td><td>MANUAL_NONDISRUPTIVE. This is the only valid value for the Workload Deployment Type of MOVABLE.</td></tr><tr><td>CLUSTER</td><td>DYNAMIC</td></tr></table>	Table 19. Default values for recovery-method		Workload Deployment Type	recovery-method Default	SINGLE_SYSTEM	MANUAL_DISRUPTIVE	MOVABLE	MANUAL_NONDISRUPTIVE. This is the only valid value for the Workload Deployment Type of MOVABLE.	CLUSTER	DYNAMIC
Table 19. Default values for recovery-method													
Workload Deployment Type	recovery-method Default												
SINGLE_SYSTEM	MANUAL_DISRUPTIVE												
MOVABLE	MANUAL_NONDISRUPTIVE. This is the only valid value for the Workload Deployment Type of MOVABLE.												
CLUSTER	DYNAMIC												

Table 18. Network parameters fields (continued)

Field	Type	Required or optional	Description
application-owned	String	Optional	<p>Defines the ownership characteristics of an IP address with regard to applications. The following values are valid:</p> <p><b>"true"</b> The IP address is owned by the single application that is currently bound to the IP address. Only one application can simultaneously use this IP address. This is the default value. Requests containing the same deployment-id from the same template instance (same job-name) will provision different IP addresses each time. Requests from a different template instance (different job-name) containing the same deployment-id as previous requests will return the same IP address returned to those previous requests and increment a use count. When deprovisioning, the use count must reach zero before the IP address is returned to the pool.</p> <p><b>"false"</b> The IP address is not owned by any single application. Multiple applications can use the IP address simultaneously. A value of false is only valid when the recovery-method value is MANUAL_NONDISRUPTIVE.</p> <p><b>Note:</b> This property is a string value (not Boolean).</p>
job-name	String	Required	Job name associated with the provisioned instance.
requires-zcx-addr	boolean	Optional	Indicates whether the resource pool needs to support z/OS Container Extensions (zCX).

Table 20. Systems field

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, with a response body. See [“Response content” on page 55](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 21. HTTP error response codes for an obtain IP address request</i>	
HTTP error status code	Description
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 22 on page 55](#).

<i>Table 22. Response from an obtain IP address request</i>		
Field	Type	Description
id	String	Identifier of the IP address.
ipaddr	String	IP address returned from the Network Configuration Assistant task.

## Example HTTP interaction

In [Figure 22 on page 56](#), a request is submitted to obtain an IP address.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/ip/obtain
```

```
{
  "registry-uuid": "E1E2A1C4",
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IZU$0AA",
  "network-params": {
    {
      "name": "CICSA IP",
      "usage-type": "internal",
      "ipaddr": "any4",
      "system-list": [
        {
          "sysplex-name": "LOCAL",
          "sysplex-node-name": "MVSW",
          "system-nickname": "MVSWNIK"
        }
      ]
    },
    "deployment-id": "CICSBasic",
    "host-name": "myHostName",
    "recovery-method": "DYNAMIC",
    "application-owned": true,
    "requires-zcx-addr": true
  }
}
```

Figure 22. Sample request to obtain an IP address, with the request body

The following is the response body for the example obtain IP address request.

```
{
  "id": "101",
  "ipaddr": "192.168.1.1"
}
```

## Release an IP address

Use this operation to release an IP address from a network resource pool.

### HTTP method and URI path

---

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/ip/actions/release
```

---

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation releases an IP address from a network resource pool, calling through the tenant's resource pool.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in an IP address being released.

### Request content

The request content is expected to contain a JSON object that describes the IP address to be released. See [Table 23 on page 57](#).

Table 23. Request content for the release IP address request

Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> <li>Provisioning workflows: \${_workflow-registryID}</li> <li>Action workflows: \${_workflow-parentRegistryID}</li> </ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-uuid</b>	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
<b>template-name</b>	String	Required	Name of the template that is associated with the resource pool. Derived from workflow internal variable, \${_workflow-templateName}.
<b>tenant-id</b>	String	Required	ID of the tenant that is associated with the resource pool. Derived from workflow internal variable, \${_workflow-tenantID}.
<b>network-parms</b>	JSON object	Required	Network parameters for the request. See <a href="#">Table 24 on page 57</a> .

Table 24. Network parameters fields

Field	Type	Required or optional	Description
ip-id	String	Required	Identifier of the IP address. This is returned as the id property in an Obtain an IP address request.
system-list[]	String	Optional	A list of target systems for deprovisioning, When specified, system-list must contain only one system or all systems in the network resource pool. When system-list is omitted and system-name is not present, the resource will be deprovisioned on each system in the network resource pool. See <a href="#">Table 25 on page 57</a> for details on the values.

Table 25. Systems field

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 26. HTTP error response codes for a release IP address request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 403	The request cannot be processed because the client is not authorized.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

None.

## Example HTTP interaction

In [Figure 23 on page 58](#), a request is submitted to release an IP address.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/ip/release
```

```
{
  "registry-uuid": "E1E2A1C4",
  "tenant-id": "IYU0AA",
  "network-params": {
    "ip-id": "1001",
    "system-list": [
      {
        "sysplex-name": "LOCAL",
        "sysplex-node-name": "MVS",
        "system-nickname": "MVS"
      }
    ]
  }
}
```

Figure 23. Sample request to release an IP address, with the request body

## Obtain a port

Use this operation to obtain a port from a resource pool that has a configured network resource pool.

### HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/obtain
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation obtains a port from a resource pool with a configured network resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a port being obtained.

### Request content

The request content is expected to contain a JSON object that describes the port to be obtained. See [Table 27 on page 59](#).

Table 27. Request content for the obtain port request			
Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"><li>Provisioning workflows: \${_workflow-registryID}</li><li>Action workflows: \${_workflow-parentRegistryID}</li></ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-uuid</b>	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
<b>template-name</b>	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
<b>tenant-id</b>	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
<b>network-parms</b>	JSON object	Required	Network parameters for the request. See <a href="#">Table 28 on page 60</a> .

Table 28. Network parameters fields

Field	Type	Required or optional	Description
name	String	Optional	Name used in a panel for the Network Configuration Assistant task to identify who is using the network resource. It is recommended to use workflow internal variable \${_workflow-softwareServiceInstanceName} for this parameter.
usage-type	String	Optional	Used as a filter. If not specified, only network resource pools without a usage type match. If specified, must match the usage type in the network resource pool definition in the Network Configuration Assistant task.
port	String	Optional	Request port number. If port is not specified, a port is provisioned from available port allocation ranges of the specified transport. If port is specified, it must fit within an available range. Available ranges are those which are associated with the targeted network resource pool and match the provided usage type.
job-name	String	Required	Job name associated with the provisioned instance.
system-name	String	Optional	System name. Derived from a workflow internal variable, \${_workflow-systemName}.
deployment-id	String	Optional	Workflow-defined string token, used to group all provisioned resources with a server instance.
host-name	String	Optional	Host-name for the Network Configuration Assistant task.
system-list[]	String	Optional	A list of target systems for provisioning. If you omit this field and the system-name field, all systems in the network resource pool are targets. See <a href="#">Table 29 on page 61</a> .
is-port-shared	String	Optional	Indicates whether the port should be capable of being shared with other servers on the same system that are listening on the same IP address and port. Used to enable same-system coexistence or when multiple equivalent servers normally run simultaneously on the same system bound to the same IP address and port.
is-port-distributed	String	Optional	Indicates whether the port is used for workload distribution, as follows:  <b>true</b> The port will be used in conjunction with provisioned distributed or group DVIPAs (IP addresses provisioned with a value of DYNAMIC for the recovery-method field) when creating the TCP/IP VIPADISTRIBUTE statement for workload distribution among the clustered instances. These are ports that the server application will use when binding its listening sockets to the distributed or group DVIPA.  <b>false</b> The port is not used for workload distribution.

Table 28. Network parameters fields (continued)

Field	Type	Required or optional	Description
intent	String	Optional	<p>Indicates the intent, as follows:</p> <p><b>shared</b> The port number can be reused for multiple port provisioning requests on the same TCP/IP stack. This is the default value.</p> <p><b>exclusive</b> The port provisioning request is given a port number that has not been previously provisioned on this TCP/IP stack. Moreover, subsequent port provisioning requests do not return the same port number on the same TCP/IP stack, as long as this port remains provisioned.</p> <p><b>workloadExclusive</b> The port provisioning request is given port numbers that are unique for each port provisioning request within the same cluster, regardless of the system or stack it is provisioned on.</p>

Table 29. Systems field

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, with a response body. See [“Response content” on page 62](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 30. HTTP error response codes for an obtain port request

HTTP error status code	Description
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.

Table 30. HTTP error response codes for an obtain port request (continued)	
HTTP error status code	Description
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 31 on page 62](#).

Table 31. Response from an obtain port request		
Field	Type	Description
id	String	Identifier of the port.
port	String	Port number.

## Example HTTP interaction

In [Figure 24 on page 62](#), a request is submitted to obtain a port.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/port/actions/obtain
```

```
{
  "registry-uuid": "E1E2A1C4",
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IZU$0AA",
  "network-parms": {
    "name": "PortForCics1",
    "port": "80",
    "usage-type": "Internal",
    "job-name": "WLP001",
    "deployment-id": "CICSBasic",
    "host-name": "myHostName",
    "is-port-shared": true,
    "is-port-distributed": true,
    "intent": "workloadExclusive",
    "system-list": [
      {
        "sysplex-name": "LOCAL",
        "sysplex-node-name": "MVS",
        "system-nickname": "MVS"
      }
    ]
  }
}
```

Figure 24. Sample request to obtain a port, with request body

The following is the response body for the example obtain port request.

```
{
  "id": "82346",
  "port": "80",
}
```

## Release a port

Use this operation to release a port from a network resource pool.

### HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/port/actions/release
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation releases a port from a network resource pool, calling through the tenant's resource pool.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in a port being released.

### Request content

The request content is expected to contain a JSON object that describes the port to be released. See [Table 32 on page 63](#).

Table 32. Request content for the release port request			
Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"><li>Provisioning workflows: \${_workflow-registryID}</li><li>Action workflows: \${_workflow-parentRegistryID}</li></ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-uuid</b>	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
<b>template-name</b>	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
<b>tenant-id</b>	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
<b>network-parms</b>	JSON object	Required	Network parameters for the request. See <a href="#">Table 33 on page 64</a> .

*Table 33. Network parameters fields*

Field	Type	Required or optional	Description
port-id	String	Required	Identifier of the port. This is returned as the id property in an Obtain a port request.
system-list[]	String	Optional	A list of target systems for deprovisioning. When specified, system-list must contain only one system or all systems in the network resource pool. When system-list is omitted and system-name is not present, the resource will be deprovisioned on each system in the network resource pool. See Table 34 on page 64 for details on the values.
system-name	String	Optional	Specifies the target system that the resource will be deprovisioned from.

*Table 34. Systems field*

Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex.
sysplex-node-name	String	Required	Node name of the system.
system-nickname	String	Required	Nickname of the system.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

*Table 35. HTTP error response codes for a release port request*

HTTP error status code	Description
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

None.

## Example HTTP interaction

In Figure 25 on page 65, a request is submitted to release a port.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/port/actions/release
```

```
{
  "registry-uuid": "E1E2A1C4",
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IYU0AA",
  "network-params" :
  {
    "port-id" : "1001",
    "system-name": "SY1"
  }
}
```

Figure 25. Sample request to release a port, with the request body

## Obtain a SNA application name

Use this operation to obtain a SNA application name from a resource pool that has a configured network resource pool.

## HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/obtain
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

## Query parameters

None.

## Description

This operation obtains a SNA application name from a resource pool with a configured network resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a SNA application name being obtained.

## Request content

The request content is expected to contain a JSON object that describes the SNA application name to be obtained. See [Table 36 on page 66](#).

Table 36. Request content for the obtain SNA application name request

Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> <li>Provisioning workflows: \${_workflow-registryID}</li> <li>Action workflows: \${_workflow-parentRegistryID}</li> </ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-uuid</b>	String	Optional	This field is deprecated. It contains a unique identifier for the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateID}.
<b>template-name</b>	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
<b>tenant-id</b>	String	Required	Name of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
<b>network-parms</b>	JSON object	Required	Network parameters for the request. See <a href="#">Table 37 on page 66</a> .

Table 37. Network parameters fields

Field	Type	Optional/ Required	Description
name	String	Optional	Name used in a panel for the Network Configuration Assistant task to identify who is using the network resource. It is recommended to use workflow internal variable workflow internal variable, \${_workflow-softwareServiceInstanceName} for this parameter.
deployment-id	String	Optional	Workflow-defined string token.
sna-application-name	String	Required	A name for the SNA application. Derived from the workflow internal variable \${_workflow-softwareServiceInstanceName}.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, with a response body. See [“Response content” on page 67](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 38. HTTP error response codes for an obtain SNA application name request</i>	
HTTP error status code	Description
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 39 on page 67](#).

<i>Table 39. Response from an obtain SNA application name request</i>		
Field	Type	Description
id	String	Identifier of the SNA application name. Needed for the release call, as the value for the appl-name-id property.
appl-name	String	Required. Application name from the network resource pool in the Network Configuration Assistant task.

## Example HTTP interaction

In [Figure 26 on page 67](#), a request is submitted to obtain a SNA application name.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/snaapplname/actions/obtain
{
  "template-uuid": "F0F1A1C2",
  "template-name": "CICSBasic",
  "tenant-id": "IZU$0AA",
  "network-parms" :
  {
    "name": "CICSA APPLID",
    "deployment-id": "CICSBasic",
    "sna-appl-name": "CICSA001"
  }
}
```

*Figure 26. Sample request to obtain a SNA application name, with request body*

The following is the response body for the example obtain SNA application name request.

```
{
  "id": 82346,
  "appl-name": "CICSC10"
}
```

## Release a SNA application name

Use this operation to release a SNA application name from a network resource pool.

### HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/network/snaapplname/actions/release
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation releases a SNA application name from a network resource pool, calling through the tenant's resource pool.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in a SNA application name being released.

### Request content

The request content is expected to contain a JSON object that describes the SNA application name to be released. See [Table 40 on page 68](#).

Table 40. Request content for the release SNA application name request			
Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"><li>Provisioning workflows: \${_workflow-registryID}</li><li>Action workflows: \${_workflow-parentRegistryID}</li></ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-name</b>	String	Required	Name of the template that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-templateName}.
<b>network-parms</b>	JSON object	Required	Network parameters for the request. See <a href="#">Table 41 on page 69</a> .

Table 41. Network parameters fields

Field	Type	Optional/ Required	Description
appl-name-id	String	Required	Identifier of the SNA application name. This is returned as the id property in an Obtain a SNA application name request.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a network administrator for APIs related to network resources in the domain that the tenant is associated with.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 42. HTTP error response codes for a release SNA application name request

HTTP error status code	Description
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

None.

## Example HTTP interaction

In [Figure 27 on page 69](#), a request is submitted to release a SNA application name.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/network/snaapplname/release
{
  "tenant-id": "IYU0AA",
  "network-parms":
  {
    "appl-name-id": "82346"
  }
}
```

Figure 27. Sample request to release a SNA application name

## Add a classification rule

Use this operation to add a classification rule in a WLM Policy with service level agreement specified in a resource pool.

### HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/add
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation adds a classification rule in a WLM policy for a middleware instance with a service level agreement specified in the tenant's resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a classification rule being added.

### Request content

The request content is expected to contain a JSON object. See [Table 43 on page 70](#).

Table 43. Request content for the add classification rule request			
Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"><li>Provisioning workflows: \${_workflow-registryID}</li><li>Action workflows: \${_workflow-parentRegistryID}</li></ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-name</b>	String	Required	Name of the template that is associated with the tenant. Derived from a workflow internal variable, \${_workflow-templateName}.
<b>tenant-id</b>	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
<b>wlm-parms</b>	JSON object	Required	WLM parameters for the request. See <a href="#">Table 44 on page 71</a> .

Table 44. WLM parameters fields			
Field	Type	Optional/Required	Description
qualifier	String	Required	The started task name. In most cases it can be derived from workflow internal variable \${_workflow-softwareServiceInstanceName}

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a WLM administrator in the domain that the tenant is associated with.

The user must also be the owner of the software services registry entry for the software services instance.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 45. HTTP error response codes for a release SNA application name request	
HTTP error status code	Description
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. See [Table 46 on page 71](#).

Table 46. Response from an add classification rule request		
Field	Type	Description
cl-rule-id	String	Required. Identifier of the classification rule.
report-class-name	String	Report class that is associated with the resource pool.
service-class-name	String	Service class that is associated with the SLA that is defined in the resource pool.

## Example HTTP interaction

In [Figure 28 on page 72](#), a request is submitted to add a classification rule.

```

POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/wlm/clrule/actions/add
{
  "registry-uuid": "E1E2A1C4",
  "template-name": "CICSBasic",
  "tenant-id": "IYU102",
  "resistry-id": "0d375584-305d-4bd5-b26e-88ac74c8171a",
  "wlm-parms":
  {
    "qualifier": "CICSA001"
  }
}

```

Figure 28. Sample request to add a classification rule

The following is the response body for the request:

```

{
  "cl-rule-id" : "82346",
  "report-class-name": "RPTCLASS",
  "service-class-name": "SCGOLD",
}

```

## Remove a classification rule

Use this operation to remove a classification rule from a WLM Policy.

### HTTP method and URI path

---

```
POST /zosmf/resource-mgmt/rest/<version>/rdp/wlm/clrule/actions/remove
```

---

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### Query parameters

None.

### Description

This operation removes a classification rule from a WLM policy.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the request resulted in a classification rule being removed.

### Request content

The request content is expected to contain a JSON object. See [Table 47 on page 73](#).

Table 47. Request content for the remove classification rule request

Field name	Type	Required or optional	Description
<b>registry-uuid</b>	String	See description	Contains a unique identifier for the registry object that is associated with the resource pool. Derived from a workflow internal variable as follows: <ul style="list-style-type: none"> <li>Provisioning workflows: \${_workflow-registryID}</li> <li>Action workflows: \${_workflow-parentRegistryID}</li> </ul> Required when provisioning network resources as part of a composite cluster template.
<b>template-name</b>	String	Required	Name of the template that is associated with the tenant. Derived from a workflow internal variable, \${_workflow-templateName}.
<b>tenant-id</b>	String	Required	ID of the tenant that is associated with the resource pool. Derived from a workflow internal variable, \${_workflow-tenantID}.
<b>wlm-parms</b>	JSON object	Required	WLM parameters for the request. See <a href="#">Table 48 on page 73</a> .

Table 48. WLM parameters fields

Field	Type	Optional/ Required	Description
cl-rule-id	String	Required	Returned by an Add Classification Rule request as the id property.

## Authorization requirements

The user must be a consumer in the tenant, a domain administrator in the domain that the tenant is associated with, or a WLM administrator in the domain that the tenant is associated with.

The user must also be the owner of the software services registry entry for the software services instance.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (No Content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 49. HTTP error response codes for a release SNA application name request	
HTTP error status code	Description
<b>HTTP 403</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

Table 49. HTTP error response codes for a release SNA application name request (continued)	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

None.

## Example HTTP interaction

In Figure 29 on page 74, a request is submitted to add a classification rule.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/wlm/classification-rule/actions/remove
{
  "registry-uuid": "E1E2A1C4",
  "template-name": "CICSBasic",
  "tenant-id": "IYU102",
  "registry-id": "0d375584-305d-4bd5-b26e-88ac74c8171a",
  "wlm-parms":
  {
    "cl-rule-id" : 82346,
  }
}
```

Figure 29. Sample request to remove a classification rule

## Get data set attributes

Use this operation to retrieve data set attributes.

## HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/rdp/storage/dataset-attr/<tenant-id>/<template-name>/<registry-uuid>
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### <tenant-id>

Identifies the tenant that the data set attributes are associated with.

### <template-name>

Identifies the template that the data set attributes are associated with.

### <registry-uuid>

The UUID of the registry instance. This value is required only when using a composite cluster template.

## Query parameters

### dsn-type (optional)

The type of data set that will be allocated on the storage resource.

### size (optional)

The size of the storage resource requested.

- If both *dsn-type* and *size* are omitted, all of the data set attribute information is returned for the storage pool.
- If only *dsn-type* is specified, all of the entries that have a matching *dsn-type* are returned for all *sizes* found.
- If only *size* is specified, all of the entries that have a matching *size* are returned for all *dsn-type* found.

## Description

This operation retrieves data set attributes.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a resource pool being retrieved, and a response body is returned. See [“Response content” on page 75](#).

## Request content

None.

## Authorization requirements

The user must be a landlord, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 50. HTTP error response codes for a get data set attributes request</i>	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request body is missing a field.
<b>HTTP 404 Not found</b>	The requested resource pool does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains an array of data set attributes.

```

{
  "dataset-attributes-list" : [
    {
      "id": "RDPIDSPDS",
      "size": "SIZE",
      "type": "TYPE",
      "data-class": "DATACLAS",
      "storage-class": "STORCLAS",
      "management-class": "DATACLAS",
      "volser": "VOLSER",
      "description": "DESCRIPTION"
    }
  ]
}
Success: 200 (OK)

Error: 404 Not Found

```

Figure 30. Response from a get data set attributes request.

## Example HTTP interaction

In Figure 31 on page 76, a request is submitted to retrieve a resource pool.

```

<rest>
  <httpMethod>GET</httpMethod>
  <uriPath substitution="true">/zosmf/resource-mgmt/rest/1.0/rdp/storage/dataset-attr/${_workflow-tenantID}/${_workflow-
templateName}/${_workflow-registryID}?size="MEDIUM"&type="VSAM"
  <expectedStatusCode>200</expectedStatusCode>
  <propertyMapping mapTo="DFH_ZOS_VSAM_VOLUME">["dataset-attributes-list"][0]["volser"]</propertyMapping>
</rest>

```

Figure 31. Sample request to get data set attributes

## Resource management services

The resource management services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to get and list domains, tenants, and resource pools that were defined in support of IBM Cloud Provisioning and Management for z/OS.

Table 51 on page 76 lists the operations that the resource management services provide.

### Resource management services

Table 51. z/OSMF resource management services: operations summary	
Operation name	HTTP method and URI path
<b>“Get a domain” on page 79</b>	GET /zosmf/resource-mgmt/rest/<version>/domains/<object-id>
<b>“Get a domain history” on page 84</b>	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/history
<b>“List the domains” on page 86</b>	GET /zosmf/resource-mgmt/rest/<version>/domains/
<b>“Create a tenant” on page 91</b>	PUT /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/tenants
<b>“Get a tenant” on page 94</b>	GET /zosmf/resource-mgmt/rest/<version>/tenants/<object-id>

Table 51. z/OSMF resource management services: operations summary (continued)	
Operation name	HTTP method and URI path
<b><a href="#">“Get a tenant history” on page 100</a></b>	GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/history
<b><a href="#">“List the tenants” on page 103</a></b>	GET /zosmf/resource-mgmt/rest/<version>/tenants/
<b><a href="#">“Delete a tenant” on page 109</a></b>	DELETE /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>
<b><a href="#">“Assign CPU properties to a tenant” on page 110</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-cpu-capping-properties
<b><a href="#">“Assign memory capping properties to a tenant” on page 112</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-memory-capping-properties
<b><a href="#">“Assign a solution ID” on page 114</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-solution-id
<b><a href="#">“Disable CPU capping” on page 115</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-cpu-capping
<b><a href="#">“Disable memory capping” on page 117</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-memory-capping
<b><a href="#">“Disable metering” on page 118</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-metering
<b><a href="#">“Enable CPU capping” on page 120</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-cpu-capping
<b><a href="#">“Enable memory capping” on page 121</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-memory-capping
<b><a href="#">“Enable metering” on page 123</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-metering
<b><a href="#">“Add tenant consumer” on page 124</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/actions/add
<b><a href="#">“Remove tenant consumer” on page 126</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/actions/remove
<b><a href="#">“Add tenant description” on page 127</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/description/actions/add
<b><a href="#">“Add tenant groups” on page 129</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/add

Table 51. z/OSMF resource management services: operations summary (continued)	
Operation name	HTTP method and URI path
<b><a href="#">“Remove tenant groups” on page 130</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/remove
<b><a href="#">“Get a resource pool” on page 132</a></b>	GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/<rdp-id>
<b><a href="#">“Get a domain resource pool” on page 138</a></b>	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/<rdp-id>
<b><a href="#">“Get a resource pool history” on page 145</a></b>	GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/history
<b><a href="#">“List the resource pools” on page 147</a></b>	GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/
<b><a href="#">“List domain resource pools” on page 152</a></b>	GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/
<b><a href="#">“Update the security state for a tenant” on page 159</a></b>	POST /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/state/actions/update
<b><a href="#">“Get security resources” on page 160</a></b>	GET /zosmf/resource-mgmt/rest/<version>/security-resources

## Authorization requirements

Use of the Resource Management services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

In addition, the user’s z/OS user ID may need access to other resources, including those that define roles such as the landlord and domain administrator. The specific requirements for each resource management service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements” on page 48](#). For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 400 Bad request

There is a missing field in the request body.

### HTTP 401 Not authorized

The request cannot be processed because the client is not authorized.

### HTTP 403 Cannot access

The client does not have access rights to the content (they are not authorized). As a result, the server is not returning the expected.

### HTTP 404 Not found

The requested resource does not exist.

### HTTP 409 Conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

### HTTP 500 Server error

The server encountered an error when it processed the request.

## Get a domain

Use this operation to retrieve a domain.

### HTTP method and URI path

---

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<object-id>
```

---

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### <object-id>

Identifies the domain to be retrieved.

### Query parameters

None.

### Description

This operation retrieves a domain.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a domain being retrieved.

### Request content

None.

### Authorization requirements

The user must be a landlord, domain administrator, or consumer in the domain.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content” on page 80](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 52. HTTP error response codes for a get domain request	
HTTP error status code	Description
HTTP 400 Bad request	The request contains incorrect parameters.
HTTP 404 Not found	The requested domain does not exist.

Table 52. HTTP error response codes for a get domain request (continued)

HTTP error status code	Description
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the domain. See [Table 53 on page 80](#).

Table 53. Response from a get domain request

Field	Type	Description
<b>domain-id</b>	String	The generated ID for the domain.
<b>domain-name</b>	String	Descriptive name for the domain.
<b>domain-system-list</b>	Array	Array describing the systems in the domain. See <a href="#">Table 54 on page 82</a> .
<b>domain-administrator-list</b>	Array of Strings	List of user IDs for the domain administrators.
<b>domain-administrator-group-list</b>	Array of Strings	List of SAF groups for the domain administrators.
<b>network-administrator-list</b>	Array of Strings	List of user IDs for the network administrators.
<b>network-administrator-group-list</b>	Array of Strings	List of SAF groups for the network administrators.
<b>wlm-administrator-list</b>	Array of Strings	List of user IDs for the WLM administrators.
<b>wlm-administrator-group-list</b>	Array of Strings	List of SAF groups for the WLM administrators.
<b>security-administrator</b>	String	User ID of the security administrator.
<b>security-job-statement</b>	String	JOB statement JCL used in security jobs for the domain.
<b>domain-approver-list</b>	Array of Strings	List of user IDs for the domain approvers.
<b>domain-approver-group-list</b>	Array of Strings	List of SAF groups for the domain approvers.
<b>object-uri</b>	String	URI of the newly created object.
<b>domain-description</b>	String	Description of the domain.

Table 53. Response from a get domain request (continued)

Field	Type	Description
<b>automatic-security</b>	boolean	Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization: <ul style="list-style-type: none"> <li>• true if the domain is set up for automatic authorization</li> <li>• false if the domain is setup for manual authorization.</li> </ul>
<b>SAF-resources</b>	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 55 on page 83</a> .
<b>local-system</b>	JSON object	System object for the local system. See <a href="#">Table 54 on page 82</a> .
<b>domain-state</b>	String	State of the domain: <p><b>network_update_failed</b> Indicates that an attempt to modify the network cloud domain that is associated with the domain failed. The network cloud domain is modified when you modify the domain's network administrators or the domain's systems.</p> <p><b>security_update_failed</b> Indicates that the security workflow that provides automatic security failed.</p> <p><b>pending_security_update</b> Indicates one of the following: <ul style="list-style-type: none"> <li>• Manual Security definition was selected for the domain, and security setup is required.</li> <li>• Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running.</li> </ul> </p> <p><b>Operational</b> Indicates that the domain is ready for use.</p>
<b>security-job-disposition</b>	String	Disposition of security jobs after they complete successfully: <p><b>manual</b> The domain is configured for manual security.</p> <p><b>keep</b> Keep jobs on completion. Jobs that are dynamically submitted for security are kept after they complete. This is the default. You can manually delete the jobs.</p> <p><b>delete</b> Delete jobs on completion. Jobs that are dynamically submitted for security are deleted automatically after they complete.</p>

Table 53. Response from a get domain request (continued)

Field	Type	Description
<b>security-workflow-disposition</b>	String	Disposition of security workflows after they complete successfully:  <b>manual</b> The domain is configured for manual security.  <b>delete</b> Delete successful workflows on completion.  Workflows that are used for security are deleted automatically after they complete successfully. This is the default.  <b>keep</b> Keep successful workflows on completion.  Workflows that are used for security are kept after they complete successfully.  You can manually delete the workflows using the Workflows table in the Workflows task.
<b>create-time</b>	String	Date and time that the domain was created.
<b>created-by-user</b>	String	User who created the domain.
<b>last-modified-time</b>	String	The date and time of the last modification to the domain.
<b>last-modified-by-user</b>	String	User who last modified the domain.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"><li>• true if the operations are allowed</li><li>• false if the operations are not allowed.</li></ul>
<b>domain-state-error-details</b>	String	If present, contains details about the domain's error state.
<b>domain-shared-rdp-id</b>	String	The ID of the shared resource pool in this domain. It is empty until a shared resource pool in the domain is created.

Table 54. Response from a get request: Systems

Field	Type	Description
<b>sysplex-name</b>	String	Name of the sysplex. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
<b>sysplex-node-name</b>	String	Sysplex node name.
<b>system-nickname</b>	String	Unique name that is assigned to the system definition.

Table 55. Response from a create request: SAF-resource object

Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

## Example HTTP interaction

In Figure 32 on page 83, a request is submitted to retrieve a domain.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/<object-id>
```

Figure 32. Sample request to get a domain

The following is the response body for the example get domain request.

```
{
  "domain-id": "IYU1",
  "domain-name": "Domain1",
  "domain-state": "operational",
  "domain-system-list": [{
    "sysplex-name": "PLEX1",
    "sysplex-node-name": "SYS1",
    "system-nickname": "SYS1"
  }],
  "domain-administrator-list": ["landlord"],
  "domain-administrator-group-list": ["DAGRP1", "DAGRP2", ...],
  "network-administrator-list": ["netadmin"],
  "network-administrator-group-list": ["NAGRP1", ...],
  "security-administrator": "secadmin",
  "security-job-statement": "//JOB CARD JOB (ACCTINFO)",
  "automatic-security": true,
  "wlm-administrator-list": ["wlmadmin"],
  "wlm-administrator-group-list": ["WAGRP1", ...],
  "domain-approver-list": ["approver"],
  "domain-approver-group-list": ["AAGRP1", ...],
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1",
  "domain-description": "domain description",
  "create-time": "2017-11-15T19:12:45.723Z",
  "created-by-user": "landlord",
}
```

```

"last-modified-time": "2017-11-15T19:13:17.451Z",
"last-modified-by-user": "landlord",
"local-system": {
  "sysplex-name": "PLEX1",
  "sysplex-node-name": "SY1",
  "system-nickname": "SY1"
},
"SAF-resources": [{
  "description": "Designates the user as a z/OSMF user with authorization to log in.",
  "ids": ["landlord"],
  "groups": [],
  "role": "Domain Administrator",
  "resource-class": "ZMFAPLA",
  "resource-name": "IZUDFLT.ZOSMF",
  "required-access": "SAF_READ",
  "other-required-ids": [],
  "audit-requirements": ""
}, ... ],
"security-job-disposition": "delete",
"security-workflow-disposition": "delete",
"provisioning-version": "1200",
"provisioning-version-supported": true,
"domain-shared-rdp-id": "IYU1ZZZZ"
}

```

## Get a domain history

Use this operation to retrieve a domain history.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/history
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### <domain-id>

Identifies the domain for which history is to be retrieved.

### Query parameters

None.

### Description

This operation retrieves the history for a domain.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 85](#).

### Request content

None.

### Authorization requirements

The user must be a landlord, domain administrator, or consumer in the domain.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content”](#) on page 85.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 56. HTTP error response codes for a get domain history request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contains incorrect parameters.
<b>HTTP 404 Not found</b>	The requested domain does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the domain. Table 57 on page 85 lists the fields in the history object.

Table 57. Response from a get request: History object		
Field	Type	Description
<b>action-type</b>	String	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"><li>• Create</li><li>• Add administrator</li><li>• Add approver</li><li>• Add system</li><li>• Create pool</li><li>• Create tenant</li><li>• Delete pool</li><li>• Delete tenant</li><li>• Remove administrator</li><li>• Remove approver</li><li>• Remove system</li><li>• Update description</li><li>• Update state</li></ul>
<b>user</b>	String	The user who performed the action.
<b>action-time</b>	String	The time that the action was taken.
<b>action-details</b>	String	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

## Example HTTP interaction

In [Figure 33](#) on page 86, a request is submitted to retrieve the history for a domain.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/IYU1/history
```

Figure 33. Sample request to get a domain history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:28:38.133Z",
      "action-details": "Created domain"
    },
    {
      "action-type": "Update description",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:28:55.880Z",
      "action-details": "Updated domain description"
    }
  ]
}
```

## List the domains

Use this operation to list the domains that are defined for IBM Cloud Provisioning and Management for z/OS.

### HTTP method and URI path

---

```
GET /zosmf/resource-mgmt/rest/<version>/domains/
```

---

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation lists the domains for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 87](#).

### Request content

None.

### Authorization requirements

The user must be a landlord, domain administrator, or consumer in the domain.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 58. HTTP error response codes for a list domains request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contains incorrect parameters.
<b>HTTP 404 Not found</b>	The requested domain does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the domains. See [Table 59 on page 87](#).

Table 59. Response from a list domains request		
Field	Type	Description
<b>domain-list</b>	Array	Domains. See <a href="#">Table 60 on page 87</a> .
<b>local-system</b>	JSON object	System object for the local system. See <a href="#">Table 54 on page 82</a> .

Table 60. Properties of domains		
Field	Type	Description
domain-id	String	The generated ID for the domain.
domain-name	String	Descriptive name for the domain.
domain-state	String	State of the domain: <b>network_update_failed</b> Indicates that an attempt to modify the network cloud domain that is associated with the domain failed. The network cloud domain is modified when you modify the domain's network administrators or the domain's systems. <b>security_update_failed</b> Indicates that the security workflow that provides automatic security failed. <b>pending_security_update</b> Indicates one of the following: <ul style="list-style-type: none"><li>• Manual Security definition was selected for the domain, and security setup is required.</li><li>• Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running.</li></ul> <b>Operational</b> Indicates that the domain is ready for use.

Table 60. Properties of domains (continued)

Field	Type	Description
domain-system-list	Array	Array describing the systems in the domain. See <a href="#">Table 54 on page 82</a> .
domain-administrator-list	Array of Strings	List of user IDs for the domain administrators.
network-administrator-list	Array of Strings	List of user IDs for the network administrators.
security-administrator	String	User ID of the security administrator.
security-job-statement	String	JOB statement JCL used in security jobs for the domain.
automatic-security	boolean	Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization: <ul style="list-style-type: none"> <li>• true if the domain is set up for automatic authorization</li> <li>• false if the domain is setup for manual authorization.</li> </ul>
wlm-administrator-list	Array of Strings	List of user IDs for the WLM administrators.
domain-approver-list	Array of Strings	List of user IDs for the domain approvers.
object-uri	String	URI of the newly created object.
domain-description	String	Description of the domain.
create-time	String	Date and time that the domain was created.
created-by-user	String	User who created the domain.
last-modified-time	String	The date and time of the last modification to the domain.
last-modified-by-user	String	User who last modified the domain.
security-job-disposition	String	Disposition of security jobs after they complete successfully: <p><b>manual</b> The domain is configured for manual security.</p> <p><b>keep</b> Keep jobs on completion.</p> <p>Jobs that are dynamically submitted for security are kept after they complete. This is the default.</p> <p>You can manually delete the jobs.</p> <p><b>delete</b> Delete jobs on completion.</p> <p>Jobs that are dynamically submitted for security are deleted automatically after they complete.</p>

Table 60. Properties of domains (continued)

Field	Type	Description
security-workflow-disposition	String	Disposition of security workflows after they complete successfully:  <b>manual</b> The domain is configured for manual security.  <b>delete</b> Delete successful workflows on completion. Workflows that are used for security are deleted automatically after they complete successfully. This is the default.  <b>keep</b> Keep successful workflows on completion. Workflows that are used for security are kept after they complete successfully.  You can manually delete the workflows using the Workflows table in the Workflows task.
SAF-resources	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See Table 55 on page 83.
provisioning-version	String	Identifies the provisioning version of the persistent data object for the entry.
provisioning-version-supported	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"><li>• true if the operations are allowed</li><li>• false if the operations are not allowed.</li></ul>
domain-state-error-details	String	If present, contains details about the domain's error state.
domain-shared-rdp-id	String	The ID of the shared resource pool in this domain. It is empty until a shared resource pool in the domain is created.

## Example HTTP interaction

In Figure 34 on page 89, a request is submitted to list the domains.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/
```

Figure 34. Sample request to list domains

The following is the response body for the example list domains request.

```
{
  "domain-list": [{
    "domain-id": "izu$0",
    "domain-name": "default",
    "domain-system-list": [{
      "sysplex-name": "DUMBPLEX",
      "sysplex-node-name": "DUMBNODE",
      "system-nickname": "DUMBNODE_001"}],
    ...
  ]},
  "domain-administrator-list": ["ZOSMFT1", ... ],
  "network-administrator-list": ["ZOSMFT1", ... ],
```

```

        "wlm-administrator-list": ["ZOSMFT1", ... ],
        "security-administrator": "ZOSMFT1",
        "security-job-statement" : "//JOB CARD JOB(acct-info)",
        "domain-approver-list": ["ZOSMFT1", ... ],
        "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/izu$0",
        "domain-description": "default domain"
    },
    ..
}

```

```

{
    "domain-list": [{
        "domain-id": "IYU0",
        "domain-name": "default",
        "domain-state": "operational",
        "domain-system-list": [{
            "sysplex-name": "PLEX1",
            "sysplex-node-name": "SYS1",
            "system-nickname": "SYS1"
        }],
        "domain-administrator-list": [],
        "network-administrator-list": [],
        "security-administrator": null,
        "security-job-statement": "",
        "automatic-security": true,
        "wlm-administrator-list": [],
        "domain-approver-list": [],
        "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU0",
        "domain-description": "default domain",
        "create-time": "2016-10-19T08:09:08.648Z",
        "created-by-user": "izusvr",
        "last-modified-time": "2016-10-19T08:09:08.648Z",
        "last-modified-by-user": "izusvr",
        "security-job-disposition": "keep",
        "security-workflow-disposition": "delete",
        "provisioning-version": "1200",
        "provisioning-version-supported": true,
        "domain-shared-rdp-id": "IYU0ZZZ"
    }, {
        "domain-id": "IYU1",
        "domain-name": "Domain1",
        "domain-state": "operational",
        "domain-system-list": [{
            "sysplex-name": "PLEX1",
            "sysplex-node-name": "SYS1",
            "system-nickname": "SYS1"
        }],
        "domain-administrator-list": ["landlord"],
        "network-administrator-list": ["netadmin"],
        "security-administrator": "secadmin",
        "security-job-statement": "//JOB CARD JOB (ACCTINFO)",
        "automatic-security": true,
        "wlm-administrator-list": ["wlmadmin"],
        "domain-approver-list": ["approver"],
        "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1",
        "domain-description": "domain description",
        "create-time": "2017-10-15T19:12:45.723Z",
        "created-by-user": "landlord",
        "last-modified-time": "2017-10-15T19:13:17.451Z",
        "last-modified-by-user": "landlord",
        "SAF-resources": [{
            "description": "Designates the user as a z/OSMF user with authorization to log in.",
            "ids": ["landlord"],
            "groups": [],
            "role": "Domain Administrator",
            "resource-class": "ZMFAPLA",
            "resource-name": "IZUDFLT.ZOSMF",
            "required-access": "SAF_READ",
            "other-required-ids": [],
            "audit-requirements": ""
        }, ... ],
        "security-job-disposition": "delete",
        "security-workflow-disposition": "delete",
        "provisioning-version": "1200",
        "provisioning-version-supported": true,
        "domain-shared-rdp-id": "IYU0ZZZ"
    }],
    "local-system": {
        "sysplex-name": "PLEX1",
        "sysplex-node-name": "SYS1",

```

```
}
  "system-nickname": "SYS1"
}
```

## Create a tenant

Use this operation to create a tenant.

### HTTP method and URI path

```
PUT /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/tenants
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### **<domain-id>**

Identifies the domain in which to create the tenant.

### Query parameters

None.

### Description

This operation creates a tenant on the specified domain.

On successful completion, HTTP status code 201 is returned, indicating that the request resulted in a tenant being created.

### Request content

The request content is expected to contain a JSON object that describes the tenant to be created. See [Table 61 on page 91](#).

Table 61. Request content for the create tenant request			
Field name	Type	Required or optional	Description
tenant-name	String	Required	Name of the tenant.
tenant-description	String	Optional	Description of the tenant.
tenant-consumer-list	Array	Optional	List of consumer user IDs for the tenant.
tenant-group-list	Array	Optional	List of groups in the tenant.
tenant-metering-capping-properties	JSON object	Optional	Object that describes the metering and capping properties to set for the tenant. See <a href="#">Table 62 on page 92</a> .
tenant-managed-by	JSON object	Optional	Object that describes the tenant managed by information. See <a href="#">Table 63 on page 93</a> .

Table 62. Tenant metering and capping properties object

Field	Type	Required or optional	Description
tenant-capping-enabled	Boolean	Optional	If true, enables CPU capping on the specified tenant.
tenant-cpu-cap-limit	Integer	Optional	Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type: <b>lpar-share-percentage</b> 1-100. <b>service-unit</b> 1-999999999 <b>cp</b> 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP. <b>msu</b> 1-999999999.
tenant-cpu-cap-type	String	Optional	Indicates the type of capping for the tenant. Values are: <b>lpar-share-percentage</b> Percentage of the LPAR share in the general purpose processor pool. <b>service-unit</b> Unweighted CPU service units per second. <b>cp</b> A number of general purpose processors (CPs), including numbers with up to two decimal places. <b>msu</b> Millions of service units per hour. <b>none</b> Removes all of the capping properties.
tenant-memory-cap-limit	Integer	Optional	The limit in gigabytes of the memory cap.
tenant-memory-capping-enabled	Boolean	Optional	If true, enables memory capping on the specified tenant.
tenant-metering-enabled	Boolean	Optional	If true, enables metering on the specified tenant.
tenant-solution-id	String	Optional	Is the tenant solution ID. It corresponds to the Container Pricing for IBM Z solution as defined in the License Management Support (LMS) web portal. This is up to 64 characters, and optional.

<i>Table 63. Tenant managed by request object</i>			
Field	Type	Required or optional	Description
system	JSON Object	Optional	Object that describes the managing system. See <a href="#">Table 64 on page 93</a> .
tenant-id	String	Required	The generated ID for the managing tenant.
tenant-name	String	Required	Descriptive name for the managing tenant.
zosmf-url	String	Required	The URL used to access the managing z/OSMF instance.

<i>Table 64. Systems request object</i>			
Field	Type	Required or optional	Description
sysplex-name	String	Required	Name of the sysplex. The name is the value that is specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
sysplex-node-name	String	Required	Sysplex node name.
system-nickname	String	Required	Unique name that is assigned to the system definition.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 201 is returned, and with a response body. Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 65. HTTP error response codes for a get domain request</i>	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained incorrect parameters.
<b>HTTP 401 Not authorized</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 403 Cannot access</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404 Not found</b>	The requested resource does not exist.
<b>HTTP 409 Conflict</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the tenant. See [Table 66 on page 94](#).

Table 66. Response from a create tenant request		
Field	Type	Description
tenant-id	String	The generated ID for the tenant.
object-uri	String	URI of the tenant.
tenant-domain-id	String	The generated ID for the domain to which the tenant belongs.

## Example HTTP interaction

In [Figure 35 on page 94](#), a request is submitted to retrieve a domain.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/tenants
```

```
{
  "tenant-name": "string",
  "tenant-description": "string",
  "tenant-consumer-list": [
    "string"
  ],
  "tenant-group-list": [
    "string"
  ],
  "tenant-metering-capping-properties": {
    "tenant-capping-enabled": true,
    "tenant-cpu-cap-limit": 0,
    "tenant-cpu-cap-type": "lpar-share-percentage",
    "tenant-memory-cap-limit": 0,
    "tenant-memory-capping-enabled": true,
    "tenant-metering-enabled": true,
    "tenant-solution-id": "string"
  },
  "tenant-managed-by": {
    "system": {
      "sysplex-name": "string",
      "sysplex-node-name": "string",
      "system-nickname": "string"
    },
    "tenant-id": "string",
    "tenant-name": "string",
    "zosmf-url": "string"
  }
}
```

Figure 35. Sample request to create a tenant

The following is the response body for the example create tenant request.

```
{
  "tenant-id": "string",
  "object-uri": "string",
  "tenant-domain-id": "string"
}
```

## Get a tenant

Use this operation to retrieve a tenant.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<object-id>
```

In this request:

**<version>**

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

**<object-id>**

Identifies the tenant to be retrieved.

**Query parameters**

None.

**Description**

This operation retrieves a tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a tenant being retrieved, and a response body is returned. See [“Response content” on page 95](#).

**Request content**

None.

**Authorization requirements**

The user must be a domain administrator, or a consumer in the tenant.

For more information, see [“Resource management services” on page 76](#).

**HTTP status codes**

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 67. HTTP error response codes for a get tenant request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested tenant does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

**Response content**

On successful completion, the service returns a response body, which contains a JSON object with details about the tenant. See [Table 68 on page 95](#).

Table 68. Response from a get tenant request		
Field	Type	Description
tenant-id	String	Generated ID for the tenant.
tenant-name	String	Descriptive name for the tenant.

Table 68. Response from a get tenant request (continued)

Field	Type	Description
<b>tenant-shared-rdp-id</b>	String	The ID of the shared resource pool that is associated with this tenant. It is empty until a shared resource pool is created.  The resource pool ID suffix ZZ is reserved for the shared resource pool of the tenant. Dedicated resource pools cannot have ZZ as an ID suffix.
<b>tenant-domain-id</b>	String	Generated ID for the domain to which the tenant belongs.
<b>tenant-domain-name</b>	String	Descriptive name of the domain to which the tenant belongs.
<b>tenant-templates</b>	Array of Strings	Array that describes the templates that are associated with dedicated resource pools. See <a href="#">Table 69 on page 99</a> . For templates that are associated with shared resource pools, see the rdp-shared-template-name-list field in a Get resource pool request.
<b>tenant-consumer-list</b>	Array of Strings	Consumer user IDs for the tenant.
<b>object-uri</b>	String	URI of the newly created object.
<b>tenant-description</b>	String	Description of the tenant.

Table 68. Response from a get tenant request (continued)

Field	Type	Description
<b>tenant-metering-capping-properties</b>	JSON Object	<p>Properties of tenant capping:</p> <p><b>tenant-capping-enabled</b> Indicates if capping for the tenant is enabled. Values are true or false.</p> <p><b>tenant-metering-enabled</b> Indicates if metering for the tenant is enabled. Values are true or false.</p> <p><b>tenant-cpu-cap-type</b> Indicates the type of capping for the tenant. Values are:</p> <p><b>lpar-share-percentage</b> Percentage of the LPAR share in the general purpose processor pool.</p> <p><b>service-unit</b> Unweighted CPU service units per second.</p> <p><b>cp</b> A number of general purpose processors (CPs), including numbers with up to two decimal places.</p> <p><b>msu</b> Millions of service units per hour.</p> <p><b>none</b> Removes all of the capping properties.</p> <p><b>tenant-cpu-cap-limit</b> Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type:</p> <p><b>lpar-share-percentage</b> 1-100.</p> <p><b>service-unit</b> 1-999999999.</p> <p><b>cp</b> 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP.</p> <p><b>msu</b> 1-999999999.</p> <p><b>tenant-solution-id</b> Is the tenant solution ID. It corresponds to the Container Pricing for IBM Z solution as defined in the License Management Support (LMS) web portal. This is up to 64 characters, and optional.</p>
<b>tenant-resource-group-name</b>	String	Name of the tenant resource group, which can be used for processor capping or container pricing.
<b>tenant-group-list</b>	Array of Strings	List of groups in the tenant.

Table 68. Response from a get tenant request (continued)

Field	Type	Description
<b>tenant-state</b>	String	<p>State of the tenant.</p> <p><b>security_update_failed</b> Indicates that the security workflow that provides automatic security failed. The accompanying error message indicates the workflow name and workflow key. To understand why the security workflow failed, use the z/OSMF Workflows task to review the failed workflow step status and the workflow history. Make corrections as necessary, then use the <b>Set Security Complete</b> action for the domain.</p> <p>Pending Security Update indicates one of the following:</p> <p><b>pending_security_update</b> indicates one of the following:</p> <ul style="list-style-type: none"> <li>• Manual Security definition was selected for the domain, and security setup is required.</li> <li>• Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running. Make corrections as necessary, then use the <b>Set Security Complete</b> action for the tenant.</li> </ul> <p><b>wlm-update-failed</b> Indicates that an attempt to modify the Workload Management (WLM) service definition that is associated with the tenant failed. The attempted modification included one of these:</p> <ul style="list-style-type: none"> <li>• Specifying a Solution ID, enabling metering, or enabling capping</li> <li>• Modifying existing Workload Management resource pools.</li> </ul> <p>Review the accompanying error messages, make corrections as necessary and use the <b>Set Security Complete</b> action to try the Workload Management modification for the tenant and accompanying Workload Management resource pools again. Or, reverse the modification (for example, disable metering) and, if necessary, use the <b>Set Security Complete</b> action to return the state to Operational.</p> <p><b>operational</b> Indicates that the tenant is ready for use.</p>
<b>create-time</b>	String	Date and time that the tenant was created.
<b>created-by-user</b>	String	User who created the tenant.
<b>last-modified-time</b>	String	The date and time of the last modification to the tenant.
<b>last-modified-by-user</b>	String	User who last modified the tenant.
<b>SAF-resources</b>	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 70 on page 99</a> .
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.

Table 68. Response from a get tenant request (continued)

Field	Type	Description
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> <li>• true if the operations are allowed</li> <li>• false if the operations are not allowed.</li> </ul>

Table 69. Response from a get request: tenant-templates

Field	Type	Description
<b>rdp-id</b>	String	Identifier of the resource deployment pool.
<b>template-available</b>	boolean	Deprecated and does not have accurate information.
<b>template-name</b>	String	Name of the template.

Table 70. Response from a create request: SAF-resource object

Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

## Example HTTP interaction

In Figure 36 on page 99, a request is submitted to retrieve a tenant.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/<object-id>
```

Figure 36. Sample request to get a tenant

The following is the response body for the example get tenant request.

```
{
  "tenant-id": "IYU100",
  "tenant-name": "Tenant",
  "tenant-shared-rdp-id": "IYU100ZZ",
  "tenant-domain-id": "IYU1",
  "tenant-domain-name": "Domain1",
  "tenant-state": "operational",
  "tenant-metering-capping-properties": {
    "tenant-metering-enabled": false,
    "tenant-capping-enabled": false,
    "tenant-cpu-cap-type": "",
    "tenant-cpu-cap-limit": 0.0
  },
  "tenant-templates": [{
    "template-name": "Template",
    "rdp-id": "IYU10000",
    "template-available": false
  }],
  "tenant-consumer-list": ["consumer"],
  "tenant-group-list": ["group"],
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100",
  "tenant-description": "Tenant description",
  "create-time": "2017-10-18T20:27:49.723Z",
  "created-by-user": "landlord",
  "last-modified-time": "2017-10-18T20:33:00.676Z",
  "last-modified-by-user": "landlord",
  "SAF-resources": [{
    "description": "Designates the user as a z/OSMF user with authorization to log in.",
    "ids": ["consumer"],
    "groups": ["group"],
    "role": "Tenant Consumer",
    "resource-class": "ZMFAPLA",
    "resource-name": "IZUDFLT.ZOSMF",
    "required-access": "SAF_READ",
    "other-required-ids": [],
    "audit-requirements": ""
  },
  {
    "ids": ["consumer"],
    "groups": ["group"],
    "role": "Tenant Consumer",
    "resource-class": "ZMFAPLA",
    "resource-name": "IZUDFLT.ZOSMF",
    "required-access": "SAF_READ",
    "other-required-ids": [],
    "audit-requirements": ""
  },
  ...],
  "provisioning-version": "1400",
  "provisioning-version-supported": true
}
```

## Get a tenant history

Use this operation to retrieve a tenant history.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/history
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### <tenant-id>

Identifies the tenant for which history is to be retrieved.

### Query parameters

None.

### Description

This operation retrieves a tenant history.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content”](#) on page 101.

## Request content

None.

## Authorization requirements

The user must be a domain administrator, or a consumer in the tenant.

For more information, see [“Resource management services”](#) on page 76.

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content”](#) on page 101.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 71. HTTP error response codes for a get tenant history request</i>	
HTTP error status code	Description
<b>HTTP 404 Not found</b>	The requested tenant does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the tenant. [Table 72 on page 102](#) lists the fields in the history object.

Table 72. Response from a get request: History object

Field	Type	Description
<b>action-type</b>	String	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"> <li>• Create</li> <li>• Add consumer</li> <li>• Add template</li> <li>• Assign CPU capping</li> <li>• Assign memory capping</li> <li>• Assign solution ID</li> <li>• Create pool</li> <li>• Delete pool</li> <li>• Disable CPU capping</li> <li>• Disable memory capping</li> <li>• Disable metering</li> <li>• Enable CPU capping</li> <li>• Enable memory capping</li> <li>• Enable metering</li> <li>• Remove consumer</li> <li>• Remove template</li> <li>• Update description</li> <li>• Update state</li> </ul>
<b>user</b>	String	The user who performed the action.
<b>action-time</b>	String	The time that the action was taken.
<b>action-details</b>	String	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

## Example HTTP interaction

In Figure 37 on page 102, a request is submitted to retrieve the history for the tenant IYU100.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/history
```

Figure 37. Sample request to get a tenant history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:33:49.010Z",
      "action-details": "Created tenant"
    },
    {
      "action-type": "Create pool",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:34:26.514Z",
```

```
    "action-details": "Created tenant resource pool: d1.t1.*"
  }
]
}
```

## List the tenants

Use this operation to list the tenants that are defined for IBM Cloud Provisioning and Management for z/OS.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation lists the tenants that are defined for IBM Cloud Provisioning and Management for z/OS.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 104](#).

### Request content

None.

### Authorization requirements

The user must be the domain administrator, or a consumer in the tenant.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 73. HTTP error response codes for a list tenants request	
HTTP error status code	Description
HTTP 404 Not found	The domain does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the tenants. See [Table 74 on page 104](#).

Table 74. Response from a list tenants request		
Field	Type	Description
<b>tenant-list</b>	Array	Information about the tenants that are defined. See <a href="#">Table 75 on page 104</a> .

Table 75. Tenant list		
Field	Type	Description
<b>tenant-id</b>	String	Generated ID for the tenant.
<b>tenant-name</b>	String	Descriptive name for the tenant.
<b>tenant-shared-rdp-id</b>	String	The ID of the shared resource pool that is associated with this tenant. It is empty until a shared resource pool is created.  The resource pool ID suffix ZZ is reserved for the shared resource pool of the tenant. Dedicated resource pools cannot have ZZ as an ID suffix.
<b>tenant-domain-id</b>	String	Generated ID for the domain to which the tenant belongs.
<b>tenant-domain-name</b>	String	Descriptive name of the domain to which the tenant belongs.
<b>tenant-templates</b>	Array of Strings	Array that describes the templates that are associated with dedicated resource pools. See <a href="#">Table 69 on page 99</a> . For templates that are associated with shared resource pools, see the rdp-shared-template-name-list field in a Get resource pool request.
<b>tenant-consumer-list</b>	Array of Strings	Consumer user IDs for the tenant.
<b>object-uri</b>	String	URI of the newly created object.
<b>tenant-description</b>	String	Description of the tenant.

Table 75. Tenant list (continued)

Field	Type	Description
<b>tenant-metering-capping-properties</b>	JSON Object	<p>Properties of tenant capping:</p> <p><b>tenant-capping-enabled</b> Indicates if capping for the tenant is enabled. Values are true or false.</p> <p><b>tenant-metering-enabled</b> Indicates if metering for the tenant is enabled. Values are true or false.</p> <p><b>tenant-cpu-cap-type</b> Indicates the type of capping for the tenant. Values are:</p> <p><b>lpar-share-percentage</b> Percentage of the LPAR share in the general purpose processor pool.</p> <p><b>service-unit</b> Unweighted CPU service units per second.</p> <p><b>cp</b> A number of general purpose processors (CPs), including numbers with up to two decimal places.</p> <p><b>msu</b> Millions of service units per hour.</p> <p><b>none</b> Removes all of the capping properties.</p> <p><b>tenant-cpu-cap-limit</b> Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type:</p> <p><b>lpar-share-percentage</b> 1-100.</p> <p><b>service-unit</b> 1-999999999.</p> <p><b>cp</b> 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP.</p> <p><b>msu</b> 1-999999999.</p> <p><b>tenant-solution-id</b> Is the tenant solution ID. It corresponds to the Container Pricing for IBM Z solution as defined in the License Management Support (LMS) web portal. This is up to 64 characters, and optional.</p>
<b>tenant-resource-group-name</b>	String	Name of the tenant resource group, which can be used for processor capping or container pricing.

<i>Table 75. Tenant list (continued)</i>		
<b>Field</b>	<b>Type</b>	<b>Description</b>
<b>tenant-group-list</b>	Array of Strings	List of groups in the tenant.

Table 75. Tenant list (continued)

Field	Type	Description
<b>tenant-state</b>	String	<p>State of the tenant.</p> <p><b>security_update_failed</b> Indicates that the security workflow that provides automatic security failed. The accompanying error message indicates the workflow name and workflow key. To understand why the security workflow failed, use the z/OSMF Workflows task to review the failed workflow step status and the workflow history. Make corrections as necessary, then use the <b>Set Security Complete</b> action for the domain.</p> <p>Pending Security Update indicates one of the following:</p> <p><b>pending_security_update</b> indicates one of the following:</p> <ul style="list-style-type: none"> <li>• Manual Security definition was selected for the domain, and security setup is required.</li> <li>• Automatic Security workflow did not complete within 60 seconds. Use the z/OSMF Workflows task to see if the workflow for the domain completed successfully, failed, or is still running. Make corrections as necessary, then use the <b>Set Security Complete</b> action for the tenant.</li> </ul> <p><b>wlm-update-failed</b> Indicates that an attempt to modify the Workload Management (WLM) service definition that is associated with the tenant failed. The attempted modification included one of these:</p> <ul style="list-style-type: none"> <li>• Specifying a Solution ID, enabling metering, or enabling capping</li> <li>• Modifying existing Workload Management resource pools.</li> </ul> <p>Review the accompanying error messages, make corrections as necessary and use the <b>Set Security Complete</b> action to try the Workload Management modification for the tenant and accompanying Workload Management resource pools again. Or, reverse the modification (for example, disable metering) and, if necessary, use the <b>Set Security Complete</b> action to return the state to Operational.</p> <p><b>operational</b> Indicates that the tenant is ready for use.</p>

Table 75. Tenant list (continued)		
Field	Type	Description
<b>create-time</b>	String	Date and time that the tenant was created.
<b>created-by-user</b>	String	User who created the tenant.
<b>last-modified-time</b>	String	The date and time of the last modification to the tenant.
<b>last-modified-by-user</b>	String	User who last modified the tenant.
<b>SAF-resources</b>	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 70 on page 99</a> .
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> <li>• true if the operations are allowed</li> <li>• false if the operations are not allowed.</li> </ul>

## Example HTTP interaction

In [Figure 38 on page 108](#), a request is submitted to list the tenants.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/
```

*Figure 38. Sample request to list tenants*

The following is the response body for the example list tenants request.

```
{
  "tenant-list": [{
    "tenant-id": "IYU100",
    "tenant-name": "Tenant",
    "tenant-shared-rdp-id": "IYU100ZZ",
    "tenant-domain-id": "IYU1",
    "tenant-domain-name": "Domain1",
    "tenant-state": "operational",
    "tenant-metering-capping-properties": {
      "tenant-metering-enabled": false,
      "tenant-capping-enabled": false,
      "tenant-cpu-cap-type": "",
      "tenant-cpu-cap-limit": 0.0
    },
    "tenant-templates": [{
      "template-name": "Template",
      "rdp-id": "IYU10000",
      "template-available": false
    }],
    "tenant-consumer-list": ["consumer"],
    "tenant-group-list": ["group"],
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100",
    "tenant-description": "",
    "create-time": "2017-10-18T20:29:58.963Z",
    "created-by-user": "landlord",
    "last-modified-time": "2017-10-18T20:37:23.046Z",
    "last-modified-by-user": "landlord",
    "SAF-resources": [{
      "description": "Designates the user as a z/OSMF user with authorization to log in.",

```

```

        "ids": ["consumer"],
        "groups": ["group"],
        "role": "Tenant Consumer",
        "resource-class": "ZMFAPLA",
        "resource-name": "IZUDFLT.ZOSMF",
        "required-access": "SAF_READ",
        "other-required-ids": [],
        "audit-requirements": "",
    }, ...],
    "provisioning-version": "1400",
    "provisioning-version-supported": true
  }
}

```

## Delete a tenant

Use this operation to delete the specified tenant.

## HTTP method and URI path

---

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>
```

---

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation deletes the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the deletion of the specified tenant.

## Request content

None.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 76. HTTP error response codes for a delete tenant request

HTTP error status code	Description
<b>HTTP 204</b>	The request was processed successfully. However, no content was returned.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 401</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 403</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 39 on page 110, a request is submitted to delete the tenant IYU100.

```
DELETE https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100
```

Figure 39. Sample request to delete a tenant

## Assign CPU properties to a tenant

Use this operation to assign a CPU capping type and limit to the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-cpu-capping-properties
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation assigns the specified tenant a CPU capping type and CPU capping limit.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in CPU properties being assigned to the specified tenant.

## Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 77 on page 111](#).

Table 77. Request content for the assign CPU properties to a tenant request			
Field name	Type	Required or optional	Description
<b>tenant-cpu-cap-limit</b>	Integer	Required	Indicates the capacity limit for the tenant. Values vary with tenant-cpu-cap-type:  <b>lpar-share-percentage</b> 1-100.  <b>service-unit</b> 1-999999999.  <b>cp</b> 0-999999. This is the number of CPs (general purpose processors) times 100. For example, 100 represents the capacity of 1 CP.  <b>msu</b> 1-999999999.
<b>tenant-cpu-cap-type</b>	String	Required	Indicates the type of capping for the tenant. Values are:  <b>lpar-share-percentage</b> Percentage of the LPAR share in the general purpose processor pool.  <b>service-unit</b> Unweighted CPU service units per second.  <b>cp</b> A number of general purpose processors (CPs), including numbers with up to two decimal places.  <b>msu</b> Millions of service units per hour.  <b>none</b> Removes all of the capping properties.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 78. HTTP error response codes for an assign CPU properties to a tenant request	
HTTP error status code	Description
<b>HTTP 204</b>	The request was processed successfully, however, no content was returned.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 401</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 403</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 40 on page 112, a request is submitted to assign CPU properties to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/assign-cpu-capping-properties
{
  "tenant-cpu-cap-limit": 0,
  "tenant-cpu-cap-type": "lpar-share-percentage"
}
```

Figure 40. Sample request to assign CPU properties to a tenant, with the request body

## Assign memory capping properties to a tenant

Use this operation to assign memory capping properties to the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-memory-capping-properties
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation assigns memory capping properties to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in memory capping properties being assigned to the specified tenant.

## Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 79 on page 113](#).

Table 79. Request content for the assign memory capping properties to a tenant request			
Field name	Type	Required or optional	Description
tenant-memory-cap-limit	Integer	Required	The limit in gigabytes of the memory cap.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 80. HTTP error response codes for an assign CPU properties to a tenant request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 41 on page 114, a request is submitted to assign a memory capping limit of 0 to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/assign-memory-capping-properties
{
  "tenant-memory-cap-limit": 0
}
```

Figure 41. Sample request to assign memory capping properties to a tenant, with the request body

## Assign a solution ID

Use this operation to assign a solution ID to the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/assign-solution-id
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### <tenant-id>

Identifies the tenant.

### Query parameters

None.

### Description

This operation assigns a solution ID to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the assignment of a solution ID to the specified tenant.

### Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See Table 81 on page 114.

Table 81. Request content for the assign solution ID request

Field name	Type	Required or optional	Description
tenant-solution-id	String	Required	The solution ID that corresponds to your Container Pricing for IBM Z solution. This value is defined in the License Management Support (LMS) web portal and must be exactly 64 characters.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services”](#) on page 76.

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 82. HTTP error response codes for an assign solution ID request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 42 on page 115, a request is submitted to assign a solution ID "string" to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/assign-solution-id
{
  "tenant-solution-id": "string"
}
```

Figure 42. Sample request to assign a solution ID, with the request body

## Disable CPU capping

Use this operation to disable CPU capping for the specified tenant.

## HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-cpu-capping
```

In this request:

**<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

**<tenant-id>**

Identifies the tenant.

**Query parameters**

None.

**Description**

This operation disables CPU capping for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the disabling of CPU capping for the specified tenant.

**Request content**

None.

**Authorization requirements**

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

**HTTP status codes**

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 83. HTTP error response codes for a disable CPU capping request</i>	
<b>HTTP error status code</b>	<b>Description</b>
<b>HTTP 204</b>	The request was processed successfully, however, no content was returned.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 401</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 403</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

**Response content**

None.

## Example HTTP interaction

In Figure 43 on page 117, a request is submitted to disable CPU capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/disable-cpu-capping
```

Figure 43. Sample request to disable CPU capping

## Disable memory capping

Use this operation to disable memory capping for the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-memory-capping
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### <tenant-id>

Identifies the tenant.

### Query parameters

None.

### Description

This operation disables memory capping for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the disabling of memory capping for the specified tenant.

### Request content

None.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services”](#) on page 76.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 84. HTTP error response codes for a disable memory capping request	
HTTP error status code	Description
<b>HTTP 204</b>	The request was processed successfully, however, no content was returned.
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 401</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 403</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 44 on page 118, a request is submitted to disable memory capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/disable-memory-capping
```

Figure 44. Sample request to disable memory capping

## Disable metering

Use this operation to disable metering for the specified tenant.

## HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/disable-metering
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation disables metering for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the disabling of metering for the specified tenant.

## Request content

None.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 85. HTTP error response codes for a disable metering request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In [Figure 45 on page 119](#), a request is submitted to disable metering for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/disable-metering
```

*Figure 45. Sample request to disable metering*

## Enable CPU capping

Use this operation to enable CPU capping for the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-cpu-capping
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### **<tenant-id>**

Identifies the tenant.

### Query parameters

None.

### Description

This operation enables CPU capping for the specified tenant.

**Important:** To enable CPU capping for a tenant, you must first perform the 'assign-cpu-capping-properties' action to assign a CPU capping type and CPU capping limit to the tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the enabling of CPU capping for the specified tenant.

### Request content

None.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services”](#) on page 76.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 86. HTTP error response codes for an enable CPU capping request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.

Table 86. HTTP error response codes for an enable CPU capping request (continued)	
HTTP error status code	Description
<b>HTTP 403</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 46 on page 121, a request is submitted to enable CPU capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/enable-cpu-capping
```

Figure 46. Sample request to enable CPU capping

## Enable memory capping

Use this operation to enable memory capping for the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-memory-capping
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation enables memory capping for the specified tenant.

**Important:** To enable memory capping for a tenant, you must first perform the 'assign-memory-capping-properties' action to assign a memory capping limit to the tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the enabling of memory capping for the specified tenant.

## Request content

None.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 87. HTTP error response codes for an enable memory capping request</i>	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In [Figure 47 on page 122](#), a request is submitted to enable memory capping for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/enable-memory-capping
```

*Figure 47. Sample request to enable memory capping*

## Enable metering

Use this operation to enable metering for the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/actions/enable-metering
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### **<tenant-id>**

Identifies the tenant.

### Query parameters

None.

### Description

This operation enables metering for the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the enabling of metering for the specified tenant.

### Request content

None.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services”](#) on page 76.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 88. HTTP error response codes for an enable metering request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.

Table 88. HTTP error response codes for an enable metering request (continued)	
HTTP error status code	Description
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 48 on page 124, a request is submitted to enable metering for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/actions/enable-metering
```

Figure 48. Sample request to enable metering

## Add tenant consumer

Use this operation to add a list of one or more consumers to the specified tenant.

## HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/  
actions/add
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation adds a list of one or more consumers to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the addition of the specified consumers to the specified tenant.

## Request content

The request content is expected to contain a JSON object that lists the consumer user IDs to be added to the tenant. See [Table 89 on page 125](#).

Table 89. Request content for the add tenant consumer request

Field name	Type	Required or optional	Description
<b>tenant-consumer-list</b>	String	Required	Consumer user IDs to add to the tenant.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services”](#) on page 76.

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 90. HTTP error response codes for an add tenant consumer request

HTTP error status code	Description
<b>HTTP 400</b>	The request contained incorrect parameters.
<b>HTTP 401</b>	The request cannot be processed because the client is not authorized.
<b>HTTP 403</b>	The client does not have access rights to the content. As a result, the server did not return the expected response.
<b>HTTP 404</b>	The requested resource does not exist.
<b>HTTP 409</b>	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
<b>HTTP 500 Internal server error</b>	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 49 on page 125, a request is submitted to add the user IDs consumer1 and consumer2 to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/consumers/actions/add
{
  "tenant-consumer-list": ["consumer1", "consumer2"]
}
```

Figure 49. Sample request to add consumers to a tenant, with the request body

## Remove tenant consumer

Use this operation to remove a list of one or more consumers from the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/consumers/actions/remove
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### **<tenant-id>**

Identifies the tenant.

### Query parameters

None.

### Description

This operation removes a list of one or more consumers from the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the removal of the one or more listed tenant consumers from the specified tenant.

### Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 91 on page 126](#).

Table 91. Request content for the remove tenant consumer request			
Field name	Type	Required or optional	Description
tenant-consumer-list	String	Required	Consumer user IDs to remove from the tenant.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 92. HTTP error response codes for a remove tenant consumer request	
HTTP error status code	Description
HTTP 204	The request was processed successfully, however, no content was returned.

Table 92. HTTP error response codes for a remove tenant consumer request (continued)	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 50 on page 127, a request is submitted to remove consumer1 and consumer2 from the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/consumers/actions/remove
{
  "tenant-consumer-list": ["consumer1", "consumer2"]
}
```

Figure 50. Sample request to remove tenant consumer, with the request body

## Add tenant description

Use this operation to add a description to the specified tenant.

## HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/description/actions/add
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation adds a description to the specified tenant. If the tenant already has a description, this operation overwrites the existing description.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in addition of a tenant description to the specified tenant.

## Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 93 on page 128](#).

Table 93. Request content for the add tenant description request			
Field name	Type	Required or optional	Description
tenant-description	String	Required	The description to add to the tenant.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 94. HTTP error response codes for an add tenant description request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In [Figure 51 on page 129](#), a request is submitted to add the description "This group of users has the authority to provision software instances." to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/description/actions/add
{
  "tenant-description": "This group of users has the authority to provision software instances."
}
```

Figure 51. Sample request to add tenant description, with the request body

## Add tenant groups

Use this operation to add tenant groups to the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/add
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### <tenant-id>

Identifies the tenant.

### Query parameters

None.

### Description

This operation adds tenant groups to the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the addition of the tenant groups to the specified tenant.

### Request content

The request content is expected to contain a JSON object that is a list of groups to add to the tenant. See [Table 95 on page 129](#).

Table 95. Request content for the add tenant groups request			
Field name	Type	Required or optional	Description
tenant-group-list	String	Required	List of groups to add to the tenant.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 96. HTTP error response codes for an add tenant groups request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In Figure 52 on page 130, a request is submitted to add group1 and group2 to the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/groups/actions/add
{
  "tenant-group-list": [ "group1", "group2" ]
}
```

Figure 52. Sample request to add tenant groups, with the request body

## Remove tenant groups

Use this operation to remove one or more groups from the specified tenant.

## HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/groups/actions/remove
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

### <tenant-id>

Identifies the tenant.

## Query parameters

None.

## Description

This operation removes one or more groups from the specified tenant.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the removal of the tenant groups from the specified tenant.

## Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 97 on page 131](#).

Table 97. Request content for the remove tenant groups request			
Field name	Type	Required or optional	Description
tenant-group-list	String	Required	The list of groups to remove from the tenant.

## Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 98. HTTP error response codes for a remove tenant groups request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In [Figure 53 on page 132](#), a request is submitted to remove group1 and group2 from the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/groups/actions/remove
{
  "tenant-group-list": [ "group1", "group2" ]
}
```

Figure 53. Sample request to remove tenant groups, with the request body

## Get a resource pool

Use this operation to retrieve a resource pool.

### HTTP method and URI path

---

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/<rdp-id>
```

---

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### **<tenant-id>**

Identifies the tenant that the resource pool is associated with.

#### **<rdp-id>**

Identifies the resource pool to be retrieved.

### Query parameters

None.

### Description

This operation retrieves a resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a resource pool being retrieved, and a response body is returned. See [“Response content” on page 133](#).

### Request content

None.

### Authorization requirements

The user must be a landlord, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 99. HTTP error response codes for a get resource pool request

HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request body is missing a field.
<b>HTTP 404 Not found</b>	The requested resource pool does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 100 on page 133](#).

Table 100. Response from a get resource pool request

Field	Type	Description
<b>rdp-id</b>	String	The generated ID for the resource pool.
<b>rdp-name</b>	String	Descriptive name for the resource pool, in the form <i>domain-name.tenant-name</i> . For shared resource pools, the name ends with an asterisk (*).
<b>rdp-pool-type</b>	String	Type of resource pool: <b>rdp-dedicated</b> Dedicated to a single software services template <b>rdp-shared</b> Shared amongst software services templates
<b>rdp-shared-service-instance-snaapplid-name-prefix</b>	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool.  This property is returned only for shared resource pools.
<b>rdp-shared-service-instance-subsystem-name-prefix</b>	String	The software services instance name subsystem prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rdp-shared-service-instance-general-name-prefix</b>	String	The software services instance name general prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rdp-domain-id</b>	String	The generated ID for the domain to which the resource pool belongs.
<b>rdp-tenant-id</b>	String	The generated ID for the tenant to which the resource pool belongs.
<b>rdp-template-name</b>	String	Name of the software services template that the dedicated resource pool is associated with.
<b>rdp-shared-template-name-list</b>	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.

Table 100. Response from a get resource pool request (continued)

Field	Type	Description
<b>rdp-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for the template.
<b>rdp-instance-actual</b>	Integer	Actual number of software services instances for the software services template that exist.
<b>rdp-system-instance</b>	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
<b>rdp-user-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for a single user.
<b>rdp-instance-expiration-limit</b>	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
<b>rdp-system-pool</b>	Array of JSON objects	Array that describes the systems selected for provisioning. See <a href="#">Table 101 on page 137</a> .
<b>rdp-system-selection</b>	String	Type of system selection for the resource pool: <b>single</b> Use a specific system. <b>auto</b> Assign a system automatically. <b>prompt</b> Prompt the user for the system.
<b>rdp-ready</b>	boolean	Indicates if the resource pool is ready for use. <b>true</b> The resource pool is ready for use. <b>false</b> The resource pool is not ready for use.
<b>rdp-network-resources-needed</b>	boolean	Indicates if a network resource pool is required. <b>true</b> A network resource pool is required. <b>false</b> A network resource pool is not required.
<b>rdp-network-pool-id</b>	String	Generated identifier of the network resource pool.
<b>rdp-network-pool-ready</b>	String	Indicates if the network resource pool is ready for use. <b>complete</b> The network resource pool is ready for use. <b>incomplete</b> The network resource pool is not yet ready for use. <b>attention</b> The network resource pool requires attention.

Table 100. Response from a get resource pool request (continued)

Field	Type	Description
<b>rdp-relocatable-resources</b>	String	Optional, indicates if the resource pool can be relocated to a different system.  <b>movable</b> The resource pool can be relocated to a different system.  <b>none</b> Not specified.
<b>rdp-sna-applid-prefixed-instance-name</b>	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID.  <b>true</b> The prefix should be derived from the SNA application ID.  <b>false</b> The prefix should not be derived from the SNA application ID.
<b>rdp-service-instance-name-prefix</b>	String	The explicit prefix of the software services instance names.
<b>rdp-wlm-resources-needed</b>	boolean	Indicates if a workload management resource pool is required.  <b>true</b> A workload management resource pool is required.  <b>false</b> A workload management resource pool is not required.
<b>rdp-wlm-sla</b>	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
<b>rdp-wlm-pool-id</b>	String	Generated identifier of the workload management resource pool.
<b>rdp-wlm-pool-ready</b>	boolean	Indicates if the workload management resource pool is ready for use.  <b>complete</b> The workload management resource pool is ready for use.  <b>incomplete</b> The workload management resource pool is not yet ready for use.
<b>rdp-report-class-name</b>	String	The name of the workload management report class.
<b>rdp-job-statement</b>	String	JOB statement JCL that is used in provisioning jobs.

Table 100. Response from a get resource pool request (continued)

Field	Type	Description
<b>rdp-share-tenant-instances</b>	boolean	<p>Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool.</p> <p><b>true</b> Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool.</p> <p><b>false</b> Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.</p>
<b>rdp-account-modify</b>	boolean	<p>Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action.</p> <p><b>true</b> The account information can be modified.</p> <p><b>false</b> The account information cannot be modified .</p>
<b>create-time</b>	String	Date and time that the resource pool was created.
<b>created-by-user</b>	String	User ID of the user that created the resource pool.
<b>last-modified-time</b>	String	Date and time that the resource pool was most recently modified.
<b>last-modified-by-user</b>	String	User ID of the user who last modified the resource pool.
<b>object-uri</b>	String	URI of the newly resource pool object.
<b>local-system</b>	Array	Array that describes the local system. See <a href="#">Table 101 on page 137</a> .
<b>rdp-quieted</b>	boolean	<p>Indicates if the resource pool is quieted:</p> <p><b>true</b> The resource pool is quieted. You cannot provision any resources for this pool.</p> <p><b>false</b> The resource pool is not quieted. You can provision any resources for this pool.</p>
<b>rdp-tenant-report-class-name</b>	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	<p>Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry:</p> <ul style="list-style-type: none"> <li>• true if the operations are allowed</li> <li>• false if the operations are not allowed.</li> </ul>

Table 100. Response from a get resource pool request (continued)

Field	Type	Description
<b>rdp-composite-cluster</b>	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows:  <b>true</b> The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools.  <b>false</b> The resource pool is not to be used in a composite cluster.
<b>rdp-cluster-name-prefix</b>	String	The prefix used for cluster names.

Table 101. Response from a get request: Systems

Field	Type	Description
<b>sysplex-name</b>	String	Name of the sysplex. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
<b>sysplex-node-name</b>	String	Sysplex node name.
<b>system-nickname</b>	String	Unique name that is assigned to the system definition.

## Example HTTP interaction

In Figure 54 on page 137, a request is submitted to retrieve a resource pool.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/IYU10000
```

Figure 54. Sample request to get a resource pool

The following is the response body for the example get resource pool request.

```
Updated Response Body:
{
  "rdp-id": "IYU10000",
  "rdp-name": "Domain1.Tenant.Template",
  "rdp-pool-type": "rdp-dedicated",
  "rdp-quiesced": false,
  "rdp-domain-id": "IYU1",
  "rdp-tenant-id": "IYU100",
  "rdp-template-name": "Template",
  "rdp-instance-limit": 300,
  "rdp-instance-actual": 0,
  "rdp-system-instance": {
    "SY1": 0
  },
  "rdp-user-instance-limit": 12,
  "rdp-instance-expiration-limit": 0,
  "rdp-system-pool": [
    {
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SY1",
      "system-nickname": "SY1"
    }
  ],
  "rdp-system-selection": "single",
}
```

```

"rdp-ready": true,
"rdp-network-resources-needed": false,
"rdp-network-pool-id": "",
"rdp-network-pool-ready": "incomplete",
"rdp-sna-applid-prefixed-instance-name": false,
"rdp-service-instance-name-prefix": "TEMPL",
"rdp-wlm-resources-needed": false,
"rdp-wlm-sla": [],
"rdp-wlm-pool-id": "",
"rdp-wlm-pool-ready": "incomplete",
"rdp-job-statement": "",
"rdp-account-modify": true,
"rdp-share-tenant-instances": false,
"create-time": "2017-10-19T20:37:23.044Z",
"created-by-user": "landlord",
"last-modified-time": "2017-12-19T20:37:23.055Z",
"last-modified-by-user": "landlord",
"object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/IYU100000",
"local-system": {
  "sysplex-name": "PLEX1",
  "sysplex-node-name": "SY1",
  "system-nickname": "SY1"
},
"rdp-composite-cluster": true,
"rdp-cluster-name-prefix": "Y",
"provisioning-version": "1200",
"provisioning-version-supported": true
}

```

## Get a domain resource pool

Use this operation to retrieve a domain-shared resource pool.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/<rdp-id>
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### <domain-id>

Identifies the domain that the resource pool is associated with.

#### <rdp-id>

Identifies the resource pool to be retrieved.

### Query parameters

None.

### Description

This operation retrieves a domain-shared resource pool.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a resource pool being retrieved, and a response body is returned. See [“Response content” on page 139](#).

### Request content

None.

## Authorization requirements

The user must be a landlord or a domain administrator for the domain that the resource pool is in.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 102. HTTP error response codes for a get domain resource pool request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request body is missing a field.
<b>HTTP 404 Not found</b>	The requested resource pool does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 103 on page 139](#).

Table 103. Response from a get domain resource pool request		
Field	Type	Description
<b>rdp-id</b>	String	The generated ID for the resource pool.
<b>rdp-name</b>	String	Description name of the domain-shared resource pool. The resource pool name is in the form <i>domain-name.*.*</i> , where asterisks (*.*) are used to indicate that the resource pool is shared across the domain.
<b>rdp-pool-type</b>	String	Type of resource pool: <b>rdp-dedicated</b> Dedicated to a single software services template. <b>rdp-shared</b> Shared among software services templates.
<b>rdp-shared-service-instance-snaapplid-name-prefix</b>	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool.  This property is returned only for shared resource pools.
<b>rdp-shared-service-instance-subsystem-name-prefix</b>	String	The software services instance name subsystem prefix for a shared resource pool.  This property is returned only for shared resource pools.

Table 103. Response from a get domain resource pool request (continued)

Field	Type	Description
<b>rdp-shared-service-instance-general-name-prefix</b>	String	The software services instance name general prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rdp-domain-id</b>	String	The generated ID for the domain to which the resource pool belongs.
<b>rdp-tenant-id</b>	String	The generated ID for the tenant to which the resource pool belongs.
<b>rdp-template-name</b>	String	Name of the software services template that the dedicated resource pool is associated with.
<b>rdp-shared-template-name-list</b>	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
<b>rdp-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for the template.
<b>rdp-instance-actual</b>	Integer	Actual number of software services instances for the software services template that exist.
<b>rdp-system-instance</b>	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
<b>rdp-user-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for a single user.
<b>rdp-instance-expiration-limit</b>	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
<b>rdp-system-pool</b>	Array of JSON objects	Array that describes the systems that are selected for provisioning. See <a href="#">Table 104 on page 143</a> .
<b>rdp-system-selection</b>	String	Type of system selection for the resource pool:  <b>single</b> Use a specific system.  <b>auto</b> Assign a system automatically.  <b>prompt</b> Prompt the user for the system.
<b>rdp-ready</b>	Boolean	Indicates whether the resource pool is ready for use.  <b>true</b> The resource pool is ready for use.  <b>false</b> The resource pool is not ready for use.

Table 103. Response from a get domain resource pool request (continued)

Field	Type	Description
<b>rdp-network-resources-needed</b>	Boolean	Indicates whether a network resource pool is required. <b>true</b> A network resource pool is required. <b>false</b> A network resource pool is not required.
<b>rdp-network-pool-id</b>	String	Generated identifier of the network resource pool.
<b>rdp-network-pool-ready</b>	String	Indicates whether the network resource pool is ready for use. <b>complete</b> The network resource pool is ready for use. <b>incomplete</b> The network resource pool is not yet ready for use. <b>attention</b> The network resource pool requires attention.
<b>rdp-relocatable-resources</b>	String	Optional, indicates whether the resource pool can be relocated to a different system. <b>movable</b> The resource pool can be relocated to a different system. <b>none</b> Not specified.
<b>rdp-sna-applid-prefixed-instance-name</b>	Boolean	Indicates whether the prefix of the software services instance names should be derived from the SNA application ID. <b>true</b> The prefix should be derived from the SNA application ID. <b>false</b> The prefix should not be derived from the SNA application ID.
<b>rdp-service-instance-name-prefix</b>	String	The explicit prefix of the software services instance names.
<b>rdp-wlm-resources-needed</b>	Boolean	Indicates whether a workload management resource pool is required. <b>true</b> A workload management resource pool is required. <b>false</b> A workload management resource pool is not required.
<b>rdp-wlm-sla</b>	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
<b>rdp-wlm-pool-id</b>	String	Generated identifier of the workload management resource pool.

Table 103. Response from a get domain resource pool request (continued)

Field	Type	Description
<b>rdp-wlm-pool-ready</b>	Boolean	Indicates whether the workload management resource pool is ready for use.  <b>complete</b> The workload management resource pool is ready for use.  <b>incomplete</b> The workload management resource pool is not yet ready for use.
<b>rdp-report-class-name</b>	String	The name of the workload management report class.
<b>rdp-job-statement</b>	String	JOB statement JCL that is used in provisioning jobs.
<b>rdp-share-tenant-instances</b>	Boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool.  <b>true</b> Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool.  <b>false</b> Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.
<b>rdp-account-modify</b>	Boolean	Indicates whether the account information can be modified when a template is provisioned, with a Test Run or Run action.  <b>true</b> The account information can be modified.  <b>false</b> The account information cannot be modified.
<b>create-time</b>	String	Date and time that the resource pool was created.
<b>created-by-user</b>	String	User ID of the user that created the resource pool.
<b>last-modified-time</b>	String	Date and time that the resource pool was most recently modified.
<b>last-modified-by-user</b>	String	User ID of the user who last modified the resource pool.
<b>object-uri</b>	String	URI of the newly created resource pool object.
<b>local-system</b>	Array	Array that describes the local system. See <a href="#">Table 101 on page 137</a> .

Table 103. Response from a get domain resource pool request (continued)

Field	Type	Description
<b>rdp-quietesced</b>	Boolean	Indicates whether the resource pool is quietesced:  <b>true</b> The resource pool is quietesced. You cannot provision any resources for this pool.  <b>false</b> The resource pool is not quietesced. You can provision any resources for this pool.
<b>rdp-tenant-report-class-name</b>	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"><li>• true if the operations are allowed</li><li>• false if the operations are not allowed.</li></ul>
<b>rdp-composite-cluster</b>	Boolean	Indicates whether the resource pool is to be used in a composite cluster, as follows:  <b>true</b> The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools.  <b>false</b> The resource pool is not to be used in a composite cluster.
<b>rdp-cluster-name-prefix</b>	String	The prefix used for cluster names.

Table 104. Response from a get request: Systems

Field	Type	Description
<b>sysplex-name</b>	String	Name of the sysplex. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.
<b>sysplex-node-name</b>	String	Sysplex node name.
<b>system-nickname</b>	String	Unique name that is assigned to the system definition.

## Example HTTP interaction

In “Get a domain resource pool” on [page 138](#), a request is submitted to retrieve the resource pool IYU2ZZZZ from domain domain2.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp/IYU1ZZZZ
```

Figure 55. Sample request to get a domain resource pool

The following figure shows the response body for the get domain resource pool request in the previous example.

```
Updated Response Body:{
  "rdp-id": "IYU1ZZZZ",
  "rdp-name": "d1.*.*",
  "rdp-pool-type": "rdp-shared",
  "rdp-quiesced": false,
  "rdp-domain-id": "IYU1",
  "rdp-tenant-id": "IYU1ZZ",
  "rdp-template-name": "*",
  "rdp-instance-limit": 123,
  "rdp-instance-actual": 0,
  "rdp-system-instance": {
    "SY1": 0
  },
  "rdp-user-instance-limit": 123,
  "rdp-system-pool": [
    {
      "sysplex-name": "PLEX1",
      "sysplex-node-name": "SY1",
      "system-nickname": "SY1"
    }
  ],
  "rdp-system-selection": "single",
  "rdp-ready": true,
  "rdp-network-resources-needed": false,
  "rdp-network-pool-id": "",
  "rdp-network-pool-ready": "incomplete",
  "rdp-network-pool-local-ready": "incomplete",
  "rdp-sna-applid-prefixed-instance-name": false,
  "rdp-service-instance-name-prefix": "",
  "rdp-wlm-resources-needed": false,
  "rdp-wlm-sla": [],
  "rdp-wlm-pool-id": "",
  "rdp-wlm-pool-ready": "incomplete",
  "rdp-wlm-pool-local-ready": "incomplete",
  "rdp-job-statement": "",
  "rdp-account-modify": true,
  "rdp-shared-service-instance-snaapplid-name-prefix": "",
  "rdp-shared-service-instance-subsystem-name-prefix": "SP",
  "rdp-shared-service-instance-general-name-prefix": "SG",
  "rdp-share-tenant-instances": false,
  "rdp-composite-cluster": false,
  "rdp-cluster-name-prefix": null,
  "rdp-storage-resources-needed": false,
  "rdp-storage-pool": {
    "dataset-attributes-list": []
  },
  "rdp-instance-expiration-limit": 23,
  "create-time": "2020-12-14T16:11:27.135Z",
  "created-by-user": "ibmuser",
  "last-modified-time": "2020-12-14T16:11:27.167Z",
  "last-modified-by-user": "ibmuser",
  "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp/IYU1ZZZZ",
  "local-system": {
    "sysplex-name": "PLEX1",
    "sysplex-node-name": "SY1",
    "system-nickname": "SY1"
  },
  "provisioning-version": "1600",
  "provisioning-version-supported": true
}
```

## Get a resource pool history

Use this operation to retrieve a resource pool history.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/rdp/<rdp-id>/history
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### **<rdp-id>**

Identifies the resource pool for which history is to be retrieved.

### Query parameters

None.

### Description

This operation retrieves the history for a resource pool.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 146](#).

### Request content

None.

### Authorization requirements

The user must be a landlord, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned, and with a response body. See [“Response content” on page 146](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 105. HTTP error response codes for a get resource pool request	
HTTP error status code	Description
<b>HTTP 404 Not found</b>	The requested resource pool does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the resource pool. [Table 106 on page 146](#) lists the fields in the history object.

*Table 106. Response from a get request: History object*

Field	Type	Description
<b>action-type</b>	String	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"><li>• Create</li><li>• Add template</li><li>• Modify</li><li>• Quiesce</li><li>• Remove template</li><li>• Unquiesce</li></ul>
<b>user</b>	String	The user who performed the action.
<b>action-time</b>	String	The time that the action was taken.
<b>action-details</b>	String	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

## Example HTTP interaction

In [Figure 56 on page 146](#), a request is submitted to retrieve the history for the resource pool IYU10000.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/rdp/IYU10000/history
```

*Figure 56. Sample request to get a resource pool history*

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:34:26.502Z",
      "action-details": "Created resource pool"
    },
    {
      "action-type": "Add template",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:39.285Z",
      "action-details": "Added template template1, tenant: t1"
    }
  ]
}
```

## List the resource pools

Use this operation to list the resource pools.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/rdp/
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

#### **<tenant-id>**

Identifies the tenant that the resource pool is associated with.

#### **<rdp-id>**

Identifies the resource pool to be retrieved.

### Query parameters

None.

### Description

This operation lists the resource pools for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 148](#).

### Request content

None.

### Authorization requirements

The user must be a landlord, domain administrator, or a consumer for the tenant that the resource pool is in.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 107. HTTP error response codes for a get resource pool request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [“Response content” on page 148](#).

Table 108. Response from a list resource pool request

Field	Type	Description
<b>rdp-list</b>	Array	Array describing the resource pools. See <a href="#">Table 109 on page 148</a> .

Table 109. Resource pool list

Field	Type	Description
<b>rdp-id</b>	String	The generated ID for the resource pool.
<b>rdp-name</b>	String	Descriptive name for the resource pool, in the form <i>domain-name.tenant-name</i> . For shared resource pools, the name ends with an asterisk (*).
<b>rdp-pool-type</b>	String	Type of resource pool: <b>rdp-dedicated</b> Dedicated to a single software services template <b>rdp-shared</b> Shared amongst software services templates
<b>rpd-shared-service-instance-snaapplid-name-prefix</b>	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool.  This property is returned only for shared resource pools.
<b>rpd-shared-service-instance-subsystem-name-prefix</b>	String	The software services instance name subsystem prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rpd-shared-service-instance-general-name-prefix</b>	String	The software services instance name general prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rdp-domain-id</b>	String	The generated ID for the domain to which the resource pool belongs.
<b>rdp-tenant-id</b>	String	The generated ID for the tenant to which the resource pool belongs.
<b>rdp-template-name</b>	String	Name of the software services template that the dedicated resource pool is associated with.
<b>rdp-shared-template-name-list</b>	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
<b>rdp-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for the template.
<b>rdp-instance-actual</b>	Integer	Actual number of software services instances for the software services template that exist.

Table 109. Resource pool list (continued)

Field	Type	Description
<b>rdp-system-instance</b>	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
<b>rdp-user-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for a single user.
<b>rdp-system-pool</b>	Array of JSON objects	Array that describes the systems selected for provisioning. See <a href="#">Table 101 on page 137</a> .
<b>rdp-system-selection</b>	String	Type of system selection for the resource pool: <b>single</b> Use a specific system. <b>auto</b> Assign a system automatically. <b>prompt</b> Prompt the user for the system.
<b>rdp-ready</b>	boolean	Indicates if the resource pool is ready for use. <b>true</b> The resource pool is ready for use. <b>false</b> The resource pool is not ready for use.
<b>rdp-network-resources-needed</b>	boolean	Indicates if a network resource pool is required. <b>true</b> A network resource pool is required. <b>false</b> A network resource pool is not required.
<b>rdp-network-pool-id</b>	String	Generated identifier of the network resource pool.
<b>rdp-network-pool-ready</b>	String	Indicates if the network resource pool is ready for use. <b>complete</b> The network resource pool is ready for use. <b>incomplete</b> The network resource pool is not yet ready for use. <b>attention</b> The network resource pool requires attention.
<b>rdp-relocatable-resources</b>	String	Optional, indicates if the resource pool can be relocated to a different system. <b>movable</b> The resource pool can be relocated to a different system. <b>none</b> Not specified.

Table 109. Resource pool list (continued)

Field	Type	Description
<b>rdp-sna-applid-prefixed-instance-name</b>	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID.  <b>true</b> The prefix should be derived from the SNA application ID.  <b>false</b> The prefix should not be derived from the SNA application ID.
<b>rdp-service-instance-name-prefix</b>	String	The explicit prefix of the software services instance names.
<b>rdp-wlm-resources-needed</b>	boolean	Indicates if a workload management resource pool is required.  <b>true</b> A workload management resource pool is required.  <b>false</b> A workload management resource pool is not required.
<b>rdp-wlm-sla</b>	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
<b>rdp-wlm-pool-id</b>	String	Generated identifier of the workload management resource pool.
<b>rdp-wlm-pool-ready</b>	boolean	Indicates if the workload management resource pool is ready for use.  <b>complete</b> The workload management resource pool is ready for use.  <b>incomplete</b> The workload management resource pool is not yet ready for use.
<b>rdp-report-class-name</b>	String	The name of the workload management report class.
<b>rdp-job-statement</b>	String	JOB statement JCL that is used in provisioning jobs.
<b>rdp-share-tenant-instances</b>	boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool.  <b>true</b> Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool.  <b>false</b> Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.

Table 109. Resource pool list (continued)

Field	Type	Description
<b>rdp-account-modify</b>	boolean	Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action. <b>true</b> The account information can be modified. <b>false</b> The account information cannot be modified .
<b>create-time</b>	String	Date and time that the resource pool was created.
<b>created-by-user</b>	String	User ID of the user that created the resource pool.
<b>last-modified-time</b>	String	Date and time that the resource pool was most recently modified.
<b>last-modified-by-user</b>	String	User ID of the user who last modified the resource pool.
<b>object-uri</b>	String	URI of the newly resource pool object.
<b>local-system</b>	Array	Array that describes the local system. See <a href="#">Table 101 on page 137</a> .
<b>rdp-quiesced</b>	boolean	Indicates if the resource pool is quiesced: <b>true</b> The resource pool is quiesced. You cannot provision any resources for this pool. <b>false</b> The resource pool is not quiesced. You can provision any resources for this pool.
<b>rdp-tenant-report-class-name</b>	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: • true if the operations are allowed • false if the operations are not allowed.
<b>rdp-composite-cluster</b>	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows: <b>true</b> The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools. <b>false</b> The resource pool is not to be used in a composite cluster.
<b>rdp-cluster-name-prefix</b>	String	The prefix used for cluster names.

## Example HTTP interaction

In Figure 57 on page 152, a request is submitted to retrieve a resource pool.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/rdp/
```

Figure 57. Sample request to list resource pools

The following is the response body for the example get resource pool request.

```
{
  "rdp-list": [{
    "rdp-id": "IYU10100",
    "rdp-name": "domain1.tenant2.template1",
    "rdp-pool-type": "rdp-dedicated",
    "rdp-domain-id": "IYU1",
    "rdp-tenant-id": "IYU101",
    "rdp-template-name": "template1",
    "rdp-instance-limit": 101,
    "rdp-instance-actual": 0,
    "rdp-system-instance": {
      "DUMBNODE": 0
    },
    "rdp-user-instance-limit": 99,
    "rdp-system-pool": [{
      "sysplex-name": "DUMBPlex",
      "sysplex-node-name": "DUMBNODE",
      "system-nickname": "DUMBNODE"
    }],
    "rdp-system-selection": "single",
    "rdp-ready": true,
    "rdp-network-resources-needed": false,
    "rdp-network-pool-id": "",
    "rdp-network-pool-ready": "incomplete",
    "rdp-relocatable-resources": "MOVABLE",
    "rdp-sna-applid-prefixed-instance-name": false,
    "rdp-service-instance-name-prefix": "TEMP",
    "rdp-wlm-resources-needed": false,
    "rdp-wlm-sla": [],
    "rdp-wlm-pool-id": "",
    "rdp-wlm-pool-ready": "incomplete",
    "rdp-report-class-name": "",
    "rdp-job-statement": "",
    "rdp-account-modify": true,
    "rdp-share-tenant-instances": false,
    "create-time": "2017-06-19T01:58:35.919Z",
    "created-by-user": "landlord",
    "last-modified-time": "2017-06-19T02:01:34.928Z",
    "last-modified-by-user": "landlord",
    "object-uri": "/zosmf/resource-mgmt/rest/1.0/tenants/IYU101/rdp/IYU10100",
    "local-system": {
      "sysplex-name": "DUMBPlex",
      "sysplex-node-name": "DUMBNODE",
      "system-nickname": "DUMBNODE"
    },
    "rdp-composite-cluster": true,
    "rdp-cluster-name-prefix": "Y",
    "provisioning-version": "1400",
    "provisioning-version-supported": true
  }]
}
```

## List domain resource pools

Use this operation to list the domain-shared resource pools.

### HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/domains/<domain-id>/rdp/
```

In this request:

**<version>**

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

**<domain-id>**

Identifies the domain that the resource pool is associated with.

**<rdp-id>**

Identifies the resource pool to be retrieved.

## Query parameters

None.

## Description

This operation lists the domain-shared resource pools for cloud provisioning.

On successful completion, HTTP status code 200 (OK) is returned, and a response body is returned. See [“Response content” on page 153](#).

## Request content

None.

## Authorization requirements

The user must be a landlord or a domain administrator for the domain that the resource pool is in.

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 110. HTTP error response codes for a get domain resource pool request	
HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested resource pool does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the resource pool. See [Table 111 on page 153](#).

Table 111. Response from a list domain resource pool request		
Field	Type	Description
rdp-list	Array	Array describing the resource pools. See <a href="#">“List domain resource pools” on page 152</a> .

Table 112. Domain resource pool list

Field	Type	Description
<b>rdp-id</b>	String	The generated ID for the resource pool.
<b>rdp-name</b>	String	Description name of the domain-shared resource pool. The resource pool name is in the form <i>domain-name.**</i> , where asterisks (**) are used to indicate that the resource pool is shared across the domain.
<b>rdp-pool-type</b>	String	Type of resource pool: <b>rdp-dedicated</b> Dedicated to a single software services template <b>rdp-shared</b> Shared amongst software services templates
<b>rdp-shared-service-instance-snaapplid-name-prefix</b>	String	The software services instance name SNA APPLID prefix for a shared resource pool. The value is obtained from the network resource pool.  This property is returned only for shared resource pools.
<b>rdp-shared-service-instance-subsystem-name-prefix</b>	String	The software services instance name subsystem prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rdp-shared-service-instance-general-name-prefix</b>	String	The software services instance name general prefix for a shared resource pool.  This property is returned only for shared resource pools.
<b>rdp-domain-id</b>	String	The generated ID for the domain to which the resource pool belongs.
<b>rdp-tenant-id</b>	String	The generated ID for the tenant to which the resource pool belongs.
<b>rdp-template-name</b>	String	Name of the software services template that the dedicated resource pool is associated with.
<b>rdp-shared-template-name-list</b>	Array of Strings	Array of strings, where each string is the name of a template that is associated with the shared resource pool.
<b>rdp-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for the template.
<b>rdp-instance-actual</b>	Integer	Actual number of software services instances for the software services template that exist.
<b>rdp-system-instance</b>	JSON Object	The property field name is the system on which the software services template was provisioned, and the value is the number of software services instances on that system.
<b>rdp-user-instance-limit</b>	Integer	Maximum number of software services instances that are allowed for a single user.

Table 112. Domain resource pool list (continued)

Field	Type	Description
<b>rdp-instance-expiration-limit</b>	Integer	Maximum expiration time limit that can be set to software services instances under this resource pool. The value is expressed as the number of days. A value of zero (0) indicates that no limit exists. The default value is 0. This field is optional
<b>rdp-system-pool</b>	Array of JSON objects	Array that describes the systems selected for provisioning. See <a href="#">Table 101 on page 137</a> .
<b>rdp-system-selection</b>	String	Type of system selection for the resource pool: <b>single</b> Use a specific system. <b>auto</b> Assign a system automatically. <b>prompt</b> Prompt the user for the system.
<b>rdp-ready</b>	boolean	Indicates if the resource pool is ready for use. <b>true</b> The resource pool is ready for use. <b>false</b> The resource pool is not ready for use.
<b>rdp-network-resources-needed</b>	boolean	Indicates if a network resource pool is required. <b>true</b> A network resource pool is required. <b>false</b> A network resource pool is not required.
<b>rdp-network-pool-id</b>	String	Generated identifier of the network resource pool.
<b>rdp-network-pool-ready</b>	String	Indicates if the network resource pool is ready for use. <b>complete</b> The network resource pool is ready for use. <b>incomplete</b> The network resource pool is not yet ready for use. <b>attention</b> The network resource pool requires attention.
<b>rdp-relocatable-resources</b>	String	Optional, indicates if the resource pool can be relocated to a different system. <b>movable</b> The resource pool can be relocated to a different system. <b>none</b> Not specified.

Table 112. Domain resource pool list (continued)

Field	Type	Description
<b>rdp-sna-applid-prefixed-instance-name</b>	boolean	Indicates if the prefix of the software services instance names should be derived from the SNA application ID.  <b>true</b> The prefix should be derived from the SNA application ID.  <b>false</b> The prefix should not be derived from the SNA application ID.
<b>rdp-service-instance-name-prefix</b>	String	The explicit prefix of the software services instance names.
<b>rdp-wlm-resources-needed</b>	boolean	Indicates if a workload management resource pool is required.  <b>true</b> A workload management resource pool is required.  <b>false</b> A workload management resource pool is not required.
<b>rdp-wlm-sla</b>	Array of Strings	Service level agreement for the workload management pool. Indicates the level of performance that the software services instance requires. (PLATINUM, GOLD, SILVER, or BRONZE). Only one value is supported in the array.
<b>rdp-wlm-pool-id</b>	String	Generated identifier of the workload management resource pool.
<b>rdp-wlm-pool-ready</b>	boolean	Indicates if the workload management resource pool is ready for use.  <b>complete</b> The workload management resource pool is ready for use.  <b>incomplete</b> The workload management resource pool is not yet ready for use.
<b>rdp-report-class-name</b>	String	The name of the workload management report class.
<b>rdp-job-statement</b>	String	JOB statement JCL that is used in provisioning jobs.
<b>rdp-share-tenant-instances</b>	boolean	Indicates whether being a member of the tenant allows a user to view and perform actions against provisioned instances that are associated with the resource pool.  <b>true</b> Membership in the tenant gives the user access to view and perform actions against provisioned instances that are associated with the resource pool.  <b>false</b> Membership in the tenant does not give the user access to view and perform actions against provisioned instances that are associated with the resource pool. Only users who are owners of the instance or domain administrators have that access.

Table 112. Domain resource pool list (continued)

Field	Type	Description
<b>rdp-account-modify</b>	boolean	Indicates if the account information can be modified when a template is provisioned, with a Test Run or Run action. <b>true</b> The account information can be modified. <b>false</b> The account information cannot be modified .
<b>create-time</b>	String	Date and time that the resource pool was created.
<b>created-by-user</b>	String	User ID of the user that created the resource pool.
<b>last-modified-time</b>	String	Date and time that the resource pool was most recently modified.
<b>last-modified-by-user</b>	String	User ID of the user who last modified the resource pool.
<b>object-uri</b>	String	URI of the newly resource pool object.
<b>local-system</b>	Array	Array that describes the local system. See <a href="#">Table 101 on page 137</a> .
<b>rdp-quiesced</b>	boolean	Indicates if the resource pool is quiesced: <b>true</b> The resource pool is quiesced. You cannot provision any resources for this pool. <b>false</b> The resource pool is not quiesced. You can provision any resources for this pool.
<b>rdp-tenant-report-class-name</b>	String	Is the name of the WLM report class of the tenant. If present, the resource pool is able to participate in tenant-based metering and capping.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: • true if the operations are allowed • false if the operations are not allowed.
<b>rdp-composite-cluster</b>	boolean	Indicates if the resource pool is to be used in a composite cluster, as follows: <b>true</b> The resource pool is to be used in a composite cluster. rdp-composite-cluster can be true only for dedicated resource pools. <b>false</b> The resource pool is not to be used in a composite cluster.
<b>rdp-cluster-name-prefix</b>	String	The prefix used for cluster names.

## Example HTTP interaction

In “List domain resource pools” on page 152, a request is submitted to retrieve a resource pool.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp
```

Figure 58. Sample request to list domain resource pools

The following is the response body for the request to list domain resource pools in the previous example.

```
{
  "rdp-list": [
    {
      "rdp-id": "IYU1ZZZZ",
      "rdp-name": "d1.*.*",
      "rdp-pool-type": "rdp-shared",
      "rdp-quiesced": false,
      "rdp-domain-id": "IYU1",
      "rdp-tenant-id": "IYU1ZZ",
      "rdp-template-name": "*",
      "rdp-instance-limit": 123,
      "rdp-instance-actual": 0,
      "rdp-system-instance": {
        "SY1": 0
      },
      "rdp-user-instance-limit": 123,
      "rdp-system-pool": [
        {
          "sysplex-name": "PLEX1",
          "sysplex-node-name": "SY1",
          "system-nickname": "SY1"
        }
      ],
      "rdp-system-selection": "single",
      "rdp-ready": true,
      "rdp-network-resources-needed": false,
      "rdp-network-pool-id": "",
      "rdp-network-pool-ready": "incomplete",
      "rdp-network-pool-local-ready": "incomplete",
      "rdp-sna-applid-prefixed-instance-name": false,
      "rdp-service-instance-name-prefix": "",
      "rdp-wlm-resources-needed": false,
      "rdp-wlm-sla": [],
      "rdp-wlm-pool-id": "",
      "rdp-wlm-pool-ready": "incomplete",
      "rdp-wlm-pool-local-ready": "incomplete",
      "rdp-job-statement": "",
      "rdp-account-modify": true,
      "rdp-shared-service-instance-snaapplid-name-prefix": "",
      "rdp-shared-service-instance-subsystem-name-prefix": "SP",
      "rdp-shared-service-instance-general-name-prefix": "SG",
      "rdp-share-tenant-instances": false,
      "rdp-composite-cluster": false,
      "rdp-cluster-name-prefix": null,
      "rdp-storage-resources-needed": false,
      "rdp-storage-pool": {
        "dataset-attributes-list": []
      },
      "rdp-instance-expiration-limit": 23,
      "create-time": "2020-12-14T16:11:27.135Z",
      "created-by-user": "ibmuser",
      "last-modified-time": "2020-12-14T16:11:27.167Z",
      "last-modified-by-user": "ibmuser",
      "object-uri": "/zosmf/resource-mgmt/rest/1.0/domains/IYU1/rdp/IYU1ZZZZ",
      "local-system": {
        "sysplex-name": "PLEX1",
        "sysplex-node-name": "SY1",
        "system-nickname": "SY1"
      },
      "provisioning-version": "1600",
      "provisioning-version-supported": true
    }
  ]
}
```

## Update the security state for a tenant

Use this operation to update the tenant-state security field to a specified value for the specified tenant.

### HTTP method and URI path

```
/zosmf/resource-mgmt/rest/<version>/tenants/<tenant-id>/state/actions/update
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF service. The following value is valid:  
1.0.

#### **<tenant-id>**

Identifies the tenant.

### Query parameters

None.

### Description

This operation updates the tenant-state security field to the value that you specify in the request body. If the security definition is "Manual Security", use this API to set the state of the tenant to "Operational".

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the updating of the security state for the specified tenant.

### Request content

The request content is expected to contain a JSON object that describes the properties to be assigned. See [Table 113 on page 159](#).

Table 113. Request content for the update security state request			
Field name	Type	Required or optional	Description
tenant-state	String	Required	The security state to assign to the tenant.

### Authorization requirements

The user must be a landlord or a domain administrator.

For more information, see [“Resource management services” on page 76](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 114. HTTP error response codes for an update security state request	
HTTP error status code	Description
HTTP 400	The request contained incorrect parameters.

Table 114. HTTP error response codes for an update security state request (continued)	
HTTP error status code	Description
HTTP 401	The request cannot be processed because the client is not authorized.
HTTP 403	The client does not have access rights to the content. As a result, the server did not return the expected response.
HTTP 404	The requested resource does not exist.
HTTP 409	The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.
HTTP 500 Internal server error	The server encountered an error that prevented it from completing the request.

## Response content

None.

## Example HTTP interaction

In [Figure 59](#) on [page 160](#), a request is submitted to update the security state to "Operational" for the tenant IYU100.

```
POST https://localhost:4444/zosmf/resource-mgmt/rest/1.0/tenants/IYU100/state/actions/update
{
  "tenant-state": "Operational"
}
```

Figure 59. Sample request to update security state, with the request body

## Get security resources

Use this operation to retrieve security profile information.

## HTTP method and URI path

```
GET /zosmf/resource-mgmt/rest/<version>/security-resources
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF resource management service. The following value is valid: 1.0.

## Description

This operation retrieves security profile information.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in security profile information being retrieved.

## Request content

None.

## Authorization requirements

For more information, see [“Resource management services” on page 76](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned with a response body. See [“Response content” on page 161](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and the associated error message.

Table 115. HTTP error response codes for a get security resources request	
HTTP error status code	Description
HTTP 401 Unauthorized	The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the security profile. See [Table 116 on page 161](#).

Table 116. Response from a get security resources request		
Field	Type	Description
ServiceId	String	The ID of the service. For Cloud Provisioning, 5655S28PR00.
ServiceName	String	Descriptive name for the service. For Cloud Provisioning, IBM Cloud Provisioning and Management for z/OS.
MetaValidationItemVersion	Number	The meta validation item version. For Cloud Provisioning, 1.0.
Vendor	String	The vendor of the service. For Cloud Provisioning, IBM.
SecurityValidationItems	Array of Strings	The array of security validation items. There is 1 array entry for each security validation item.  See <a href="#">Table 117 on page 161</a> .

Table 117. SecurityValidationItem structure		
Field	Type	Description
ItemId	String	The ID of the security validation item.  For Cloud Provisioning, the first item is "5655S28PR00I001000000". Each subsequent item has the rightmost digit incremented by 1.
ItemType	String	The type of the item, PROGRAMMABLE, MANUAL or SEMI-PROGRAMMABLE.

Table 117. <i>SecurityValidationItem</i> structure (continued)		
Field	Type	Description
ItemCategory	String	The category of the item.
ResourceProfile	String	The SAF resource profile. For Cloud Provisioning, this is the resource profile associated with the resource.
ResourceClass	String	The SAF resource class. For Cloud Provisioning, this is the resource class associated with the resource.
WhoNeedsAccess	String	The ID of the user/group needing access. When there are multiple IDs, they are separated by a space. For Cloud Provisioning, this is constructed based on the details of the resource.
LevelOfAccessRequired	String	The required access level. For Cloud Provisioning, this is the required access for the resource profile associated with the resource.
ItemDescription	String	The description of the item. For Cloud Provisioning, this is constructed based on details of the resource.

## Example HTTP interaction

In Figure 60 on page 162, a request is submitted to retrieve security profile information.

```
GET https://localhost:4444/zosmf/resource-mgmt/rest/1.0/security-resources
```

Figure 60. Sample request to get security profile information

The following is the response body for the example get security profile information request.

```
{
  "ServiceId": "5655S28PR00",
  "ServiceName": "IBM Cloud Provisioning and Management for z/OS",
  "MetaValidationItemVersion": 1,
  "Vendor": "IBM",
  "SecurityValidationItems": [
    {
      "ItemId": "5655S28PR00I00100012",
      "ItemType": "PROGRAMMABLE",
      "ItemCategory": "Cloud Provisioning Security Administrators",
      "ResourceProfile": "IZUDFLT.ZOSMF.SECURITY.ADMIN",
      "ResourceClass": "ZMFCLLOUD",
      "WhoNeedsAccess": "IZUSECAD",
      "LevelOfAccessRequired": "READ",
      "ItemDescription": "Grants the user the security administrator role."
    },
    {
      "ItemId": "5655S28PR00I00100006",
      "ItemType": "PROGRAMMABLE",
      "ItemCategory": "Cloud Provisioning z/OSMF",
      "ResourceProfile": "IZUDFLT.ZOSMF",
      "ResourceClass": "ZMFAPLA",
      "WhoNeedsAccess": "IYU0RPAN IYU IYU000 IYU0RPAW IYU0 LANDLORD IZUADMIN ZOSMFAD"
    }
  ]
}
```

```

IZUUSER",
    "LevelOfAccessRequired": "READ",
    "ItemDescription": "Allows the user access to z/OSMF."
  },
}

```

## Software services template services

The software services template services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software services templates, which can be used to provision z/OS software in support of IBM Cloud Provisioning and Management for z/OS.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 45.

The basic procedure for provisioning software is:

1. Define domains and tenants. See [“Resource management services”](#) on page 76.
2. Create a template, specifying the workflow, action and variables files that were provided by the software vendor.

The template is added to the software services catalog.

3. Add the template to a tenant.
4. Modify the template as needed.
5. Approve any approval records. Approval records are created when a workflow or action definition file contains an element that identifies a user ID under which a workflow step or action is to be performed (a runAsUser ID). They can also be defined for the template in general, and for a domain.
6. Test the template and ensure that it successfully creates an instance, that is, that it provisions the software and that the actions defined for the instance perform as expected. Optionally, clean up the results of your test, that is, deprovision and remove the instance that you created by testing the template.
7. Publish the template to make it available to consumers.
8. Run the template to create a software instance.

There are these types of templates:

### Standard

Use these to provision a single software service. The preceding procedure assumes the use of standard templates.

### Composite

Use these to provision more than one type of software service with a single Run operation. For more information, see [“Composite templates”](#) on page 165.

[Table 118 on page 163](#) lists the operations that the software services template services provide.

[“Published software service template services”](#) on page 250 describes the REST APIs for working with published software services templates, for example, for running a template to create an instance.

[“Software services instance services”](#) on page 286 describes the REST APIs for working with software services instances.

## Software services template

Table 118. z/OSMF software services template services: operations summary	
Operation name	HTTP method and URI path
<a href="#">“Create a software services template”</a> on page 170	POST /zosmf/provisioning/rest/<version>/scc

Table 118. z/OSMF software services template services: operations summary (continued)

Operation name	HTTP method and URI path
<a href="#">“Create a new version of a software services template” on page 178</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_new_version
<a href="#">“Create a new software services template based on an existing one” on page 185</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_based_on
<a href="#">“Modify a software services template” on page 188</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>
<a href="#">“Delete a software services template” on page 194</a>	DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>
<a href="#">“List the software services templates” on page 220</a>	GET /zosmf/provisioning/rest/<version>/scc
<a href="#">“Get a software services template” on page 195</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>
<a href="#">“Get a software services template history” on page 211</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/history
<a href="#">“Get software services template documentation” on page 213</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/admin GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/consumer
<a href="#">“Get prompt variables for a software services template” on page 214</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/prompt-variables
<a href="#">“Get source information for a software services template” on page 218</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/sources
<a href="#">“Publish a software services template” on page 226</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/publish
<a href="#">“Test a software services template” on page 228</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/test

Table 118. z/OSMF software services template services: operations summary (continued)	
Operation name	HTTP method and URI path
<a href="#">“Refresh a software services template” on page 232</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/refresh
<a href="#">“Archive a software services template” on page 233</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/archive
<a href="#">“Add an approval for a software services template” on page 235</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
<a href="#">“Get an approval for a software services template” on page 236</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>
<a href="#">“List the approvals for a software services template” on page 239</a>	GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
<a href="#">“Approve an approval record for a software services template” on page 242</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>/actions/approve
<a href="#">“Batch approve approval records for a software services template” on page 243</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/actions/update
<a href="#">“Reject the use of a user ID with a software services template” on page 245</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>/actions/reject
<a href="#">“Delete an approval for a software services template” on page 246</a>	DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/<approval-object-id>
<a href="#">“Set security complete for a software services template” on page 248</a>	POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/security_complete

## Composite templates

Use a composite template to provision multiple related software services with a single Run operation. For example, you might use a composite template to provision CICS and z/OS Connect. A composite template contains other templates that are:

- Published
- Standard type. A composite template cannot contain other composite templates.

A composite template is associated with a specific domain. The published standard templates that it contains must be in that domain.

The standard templates that are members of a composite template dictate the sequence that they are provisioned in.

**Variables:** A provider can satisfy prompt variables that are associated with the standard template using the connectors field. If a prompt variable is also specified as a connector variable, the prompting of that variable is automatically disabled, because it is satisfied through the connectors field.

The composite template can also take in an optional variable input file, the composite properties file. This file contains atCreate variable values that are associated with the member standard templates. It is an alternative to providing the atCreate values with the Run action. The atCreate variable names are in this format: `<standard-template>.<atcreate-variable-name>`. If the composite properties file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables associated with the member template. No validation is done on the values that are associated with the atCreate variables.

The precedence of values for the provisioning workflow is as follows. Values that are earlier in the list override values that are later in the list.

1. Connector and prompt values.
2. Values in the composite properties file.

The precedence of values for the action workflow is as follows:

1. Prompt values.
2. wfVar values that are specified in the actions definition.
3. Values in the composite properties file.

**Resource pools:** Like standard templates, composite templates must be associated with a tenant prior to being test run and run. The following describes values for the resource pools of a composite template:

**instance name prefix**

Specified by the resource pool for the composite template.

**maximum number of instances**

Specified by the resource pool for the composite template. It cannot exceed the smallest maximum of all of the standard template resource pools.

**system selection**

Specified by the resource pool for the composite template. The system selection is limited to the common systems that are referenced by the resource pools of standard templates that are associated with the composite template. All of the standard templates that are associated with the composite template are provisioned on the same system.

**account information**

Obtained from the resource pool that is associated with the standard template.

**network resource pool**

Not specified by the resource pool for the composite template.

**workload management resource pool**

Not specified by the resource pool for the composite template.

The resource pools that are associated with the standard templates that are referenced by the composite template must exist in the same tenant as the composite template.

**Software services instances:** When you use the Run operation for a composite template, multiple catalog type registry instances are created, one parent and a child for each standard template in the sequence.

The composite resource pool prefix is applied to the parent software services instance only. The standard template resource pool prefix is applied to each child software services instance.

An instance count is updated for both the composite resource pool and for each of the standard template resource pools.

The parent software services instance contains an array of composite registry objects, and each child includes the parent registry instance object ID.

Once all of the child software services instances are provisioned, the parent software services instance moves to the provisioned state, and you can use the child software services instances, that is, you can perform actions against them. The deprovisioning action is allowed only against the parent instance. The deprovisioning sequence is the opposite of the provisioning sequence.

If any of the children fail provisioning, you can either:

- Deprovision the failed provisioning child along with any child instances that have already been provisioned. Any child in the being-initialized state will remain as is – no deprovision action is run against it.
- Restart the failed child instance. If the restart is successful, it resumes the provisioning of the remaining children instances.

Once you have deprovisioned the parent instance (by using the **Perform deprovision** action against it), you can delete the parent instance, which also deletes all of the child instances.

**Template Versions:** When a new version of a standard template that is included in a composite template is published, any composite template that includes the standard template as a member is archived. The user then has the option to either re-publish one or more of the affected composite templates or create a new version of them.

When a standard template that is a member of one or more composite templates is moved out of published state (with the Archive or Delete actions) and a new standard template is not provided simultaneously, all affected composite templates are put into missing\_required\_member state. The composite templates remain in that state until a version of the missing member is published. The new version must be a version of the original member that was included in the composite definition. Once the missing member template is in publish state, the composite template is put into archive state if only that member template was missing. Otherwise, the composite template remains in missing\_required\_member state until all of the member templates are present. From the archive state, the provider or user can choose to re-publish the archived composite templates if the content of the standard templates and the connector information is still valid. If the content of the standard templates and the connector information is no longer valid, the user can create a new version of the archived composite template. The user should delete the previous version if it is no longer needed.

When all versions of a member template are deleted and a new unrelated standard template is published, all affected composite templates are put into missing\_required\_member state. The composite templates remain in that state indefinitely because there are no versions of the missing member template, and so the requirement that the member must be a version of the original member of the composite definition cannot be satisfied. The user can either delete the composite template or create a new version of it.

**Usage scenario:** Two published templates, template1 and template2, are located in the same domain, and are associated with the same tenant, with at least one system in common.

1. A provider creates a composite template from the published standard templates, specifying template1 as sequence 1, and template2 as sequence 2, with a connector value, TEMP2\_VAR1 = TEMP1\_VAR1 from template1.
2. The provider associates the composite template with the tenant, creates the resource pool, and then test runs the template.
3. The provider displays the instances table in the Software Services task. After the parent instance is in a provisioned state, the provider performs actions against the child instance for template1.
4. When the instance is no longer needed, the provider uses an action to deprovision the parent instance.

5. Once the parent instance is in a deprovisioned state, the provider removes it. This also removes all of the child instances.

## Clustered composite templates

Clustered composite templates allow you to leverage sysplex capabilities to provision a continuously available middleware environment. With a single provisioning action, you provision network-clustered instances of a specific middleware in a sysplex. Similarly, a single deprovision action releases all of the member instances that are associated with the clustered composite template instance.

You create a clustered composite template from a single published template or from multiple published templates that use the **Use the composite template to cluster instances on systems in a sysplex** option when adding a template. The published templates must all be of the same software type (that is, they provision the same middleware).

Provisioning a clustered composite template results in each instance of the member templates being provisioned on a separate system. As a result, the total number of instances defined in a clustered composite template is limited, based on several factors, including whether the composite template resides in a single sysplex domain or a multiple sysplex domain. In a single sysplex domain, the total number of instances cannot exceed the number of systems in the domain or the number of systems in any of the resource pools that are associated with the clustered composite template definition. In a multiple-sysplex domain, the maximum number is based on the sysplex that contains the most systems in the domain; the instances will be created in this sysplex. As an example, assume that a domain encompasses systems on two sysplexes: System 1 on Sysplex A and Systems 2 and 3 on Sysplex B. Here, the maximum number of clustered instances that can be created is two because Sysplex B has two systems in the domain.

Clustered composite templates have their own resource pools. z/OS resources for all of the member instances are obtained from the same resource pool when the clustered composite template is provisioned. All of the systems in the resource pool must be a member of the same sysplex.

## Authorization requirements

Use of the software services template services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 2.

In addition, the user’s z/OS user ID may need access to other resources, including those that define roles such as the landlord and domain administrator. The specific requirements for each software services template service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements”](#) on page 48. For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

## Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 119. Response from a software services template request failure		
Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.

Table 119. Response from a software services template request failure (continued)

Field	Type	Description
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

## Error logging

Errors from the software services template services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 201 Created

The request succeeded and resulted in the creation of an object.

### HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

### HTTP 204 No content

The request succeeded, but no content is available to be returned.

### HTTP 400 Bad request

The request contained incorrect parameters.

### HTTP 403 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

### HTTP 404 Not found

The requested resource does not exist.

### HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

## Related information

The publish operation locks the template, preventing any further modification, and exposes the template to consumers. To work with a published software services template, use the REST APIs that are described in [“Published software service template services”](#) on page 250.

The run operation for a published template creates a workflow, starts the workflow, and creates a corresponding software services instance in the software services registry. To work with a software services instance, use the REST APIs described in [“Software services instance services”](#) on page 286.

## Create a software services template

Use this operation to create a software services template in the catalog. The template is a private entry until it is published.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc
```

In this request, the URI path variable <version> identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation creates a software services template in the catalog, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 170.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new software services template. A response body is provided, as described in [“Response content”](#) on page 175.

### Request content

The request content is expected to contain a JSON object that describes the software services template to be created. See [Table 120 on page 170](#).

Table 120. Request content for a request to create a software services template.				
Field name	Type	Required or optional	Valid for Template Type	Description
<b>template-type</b>	String	Optional	Standard, Composite	Identifies the type of template: <b>standard</b> Defines a single software service. <b>composite</b> Consists of multiple published templates that are provisioned together. If template-type is not specified, the type defaults to standard.
<b>composite-cluster</b>	Boolean	Optional	Composite	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
<b>composite-definition</b>	Array of objects	Required	Composite	An array of objects that define the composite template. See <a href="#">Table 121 on page 174</a> .

Table 120. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>action-definition-file</b>	String	Required	Standard	<p>Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/actions.xml.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p><b>Sequential data set</b> Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p><b>Partitioned data set</b></p> <ul style="list-style-type: none"> <li>• If the action definition file is in the same data set as the file template: just the member name.</li> <li>• If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.</li> </ul>
<b>description</b>	String	Optional	Standard, Composite	Description of the software services template.
<b>name</b>	String	Required	Standard, Composite	Descriptive name for the software services template. The name must be unique, no longer than 48 characters, and consist of alphanumeric characters (A-Z, a-z, and 0-9), national characters (\$@), underscore (_), and hyphen (-).

Table 120. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>workflow-definition-file</b>	String	Required	Standard	<p>Location of the workflow definition file, the primary XML file that defines the workflow.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf//samples/workflow_sample_automation.xml.</p> <p>The primary workflow must be a provisioning type workflow.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p><b>Sequential data set</b> Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p><b>Partitioned data set</b></p> <ul style="list-style-type: none"> <li>• If the action definition file is in the same data set as the file template: just the member name.</li> <li>• If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.</li> </ul>
<b>workflow-variable-input-file</b>	String	Optional	Standard	<p>Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>
<b>workflows-disposition</b>	String	Optional	Standard	<p>Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.</p> <p>The default is archive.</p> <p>If this field is not provided the default value of archive is used. The workflow-clean-after-provisioned field is ignored.</p>

Table 120. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>composite-variable-input-file</b>	String	Optional	Composite	<p>Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/composite.properties</p> <p>The variable names are in the following format: &lt;standard-template-name&gt;.&lt;atcreate-variable&gt;</p> <p>For example: CICS.startup=10</p> <p>If the file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables that associated with the member standard template. Values are not validated.</p>
<b>jobs-disposition</b>	String	Optional	Standard	<p>Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.</p> <p>The default is keep.</p>
<b>instances-disposition</b>	String	Optional	Standard, Composite	<p>Disposition of instances of the template after the instances are deprovisioned: keep or delete.</p> <p>The default is keep.</p>
<b>domain-name</b>	String	Optional	Standard, Composite	<p>Name of the domain. Required if the user ID has administrator authorization to more than one domain.</p>
<b>approvals</b>	Array of strings	Optional	Standard, Composite	<p>An array of strings representing user IDs of users that are responsible for approving the template.</p>
<b>workflow-clean-after-provisioned</b>	boolean	Optional	Standard	<p>This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.</p>
<b>consumer-documentation-file</b>	String	Optional	Standard, Composite	<p>Location of a file that provides information for consumers about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>

Table 120. Request content for a request to create a software services template. (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>consumer-documentation-type</b>	String	Optional	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
<b>admin-documentation-file</b>	String	Optional	Standard, Composite	Location of a file that provides information for administrators about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
<b>admin-documentation-type</b>	String	Optional	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.

Table 121. Composite-definition structure

Field	Type	Required/ optional	Description
sequence	Integer	Required	The order in which to provision the templates, starting with 1. For deprovisioning, the order is reversed.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
connectors	Array of objects	Optional	An array of connector object.  Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above.  See <a href="#">Table 122 on page 174</a> .

Table 122. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.

Table 122. Connector object (continued)			
Field	Type	Required/ optional	Description
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 175](#).

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. [Table 123 on page 175](#) lists the fields in the JSON object.

Table 123. Response from a create software services template request		
Field	Type	Description
<b>generated-name</b>	String	The generated name associated with this software services template.
<b>object-id</b>	String	The object ID of the newly created software services template. The object ID is to be used on further requests to the session.
<b>object-uri</b>	String	The object URI of the newly created software services template.
<b>SAF-resources</b>	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 124 on page 175</a> .

Table 124. Response from a create request: SAF-resource object		
Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.

Table 124. Response from a create request: SAF-resource object (continued)

Field	Type	Description
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 125. Response from a software services template request failure

Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

## Example HTTP interaction

The example in [Figure 61 on page 176](#) shows a request to create a standard software services template on the system SY1.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc
{
  "name": "s3Suspend",
  "action-definition-file": "/u/wfSuspend/a.xml",
  "workflow-definition-file": "/u/wfSuspend/p.xml",
  "domain-name": "default",
  "workflows-disposition": "keep",
  "jobs-disposition": "keep",
  "description": "This service suspends in the midst of provisioning and deprovisioning.",
  "approvals": ["zosmfad"]
}
```

Figure 61. Sample request to create a standard software services template

The following is the response body for the request.

```
{
  "generated-name": "s3Suspend.1.default",
  "object-id": "5e3c224b-eb47-47f9-847f-89456850f8aa",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/5e3c224b-eb47-47f9-847f-89456850f8aa",
  "SAF-resources": [
    {
      "description": "Controls which users are template approvers for the s3Suspend template in the default domain.",
      "ids": [
        "zosmfad"
      ],
      "groups": [],
      "role": "Template Approver",
      "resource-class": "ZMFCLLOUD",
      "resource-name": "IZUDFLT.ZOSMF.TEMPLATE.APPROVERS.IYU0.s3Suspend",
      "required-access": "SAF_READ",
      "other-required-ids": [],
      "audit-requirements": ""
    }
  ]
}
```

Figure 62. Sample response body

The example in [Figure 63 on page 177](#) shows a request to create a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc

{
  "domain-name": "default",
  "name": "S1_S2",
  "template-type": "composite",
  "composite-cluster": false,
  "description": "This is a composite template that brings up services s1 and s2. The s2 services references information from the s1 services to satisfy its run-time properties.",
  "composite-definition": [
    {
      "sequence": "1",
      "number-of-instances": 1,
      "published-template-name": "s1",
      "connectors": []
    }, {
      "sequence": "2",
      "number-of-instances": 1,
      "published-template-name": "s2",
      "connectors": [
        {
          "variable-name": "INS",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        }, {
          "variable-name": "WELSHIE",
          "source-template": "s1",
          "source-variable-name": "WELSHIE"
        }
      ]
    }
  ]
}
```

Figure 63. Sample request to create a composite software services template

The following is the response body for the request.

```
{
  "generated-name": "S1_S2.1.default",
  "object-id": "5f746dfc-ad24-4355-99d3-b83466ce4492",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/5f746dfc-ad24-4355-99d3-b83466ce4492",
  "SAF-resources": []
}
```

Figure 64. Sample response body for a composite template

## Create a new version of a software services template

You can use this operation to create a new version of a software services template, with the same name as the original, associated with the same domain and tenants, but with new source files.

### HTTP method and URI path

---

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_new_version
```

---

In this request

#### **<object-id>**

Identifies the existing software services template to create a new version of.

#### **<version>**

Is the URI path variable *<version>* that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation creates a new version of an existing software services template in the catalog. The new version has the same name as the original entry, and is associated with the same domain and tenants. However, it has new source files (workflow definition, action definition, variable input, and documentation). You cannot already have a draft software services template of this version.

The new version is assigned a version number that is the next available number in sequence.

The template that you create a new version of must be in the published, archived, or missing member state.

On successful completion, HTTP status code 201 (Normal) is returned, indicating that the request resulted in the creation of a new version of a software services template. A response body is provided, as described in [“Response content” on page 183](#).

### Request content

The request content is expected to contain a JSON object that describes the software services template to be created. See [Table 126 on page 179](#).

Table 126. Request content for the software services template request

Field name	Type	Required or optional	Valid for Template Type	Description
<b>action-definition-file</b>	String	Required	Standard	<p>Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/actions.xml.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p><b>Sequential data set</b> Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p><b>Partitioned data set</b></p> <ul style="list-style-type: none"> <li>• If the action definition file is in the same data set as the file template: just the member name.</li> <li>• If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.</li> </ul>
<b>composite-definition</b>	Array of objects	Required	Composite	<p>An array of objects that define the composite template.</p> <p>See <a href="#">Table 127 on page 182</a>.</p>
<b>composite-cluster</b>	Boolean	Optional	Composite	<p>Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.</p>
<b>description</b>	String	Optional	Standard, Composite	<p>Description of the software services template, up to 500 characters.</p>

Table 126. Request content for the software services template request (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>workflow-definition-file</b>	String	Required	Standard	<p>Location of the workflow definition file, the primary XML file that defines the workflow.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf//samples/workflow_sample_automation.xml.</p> <p>The primary workflow must be a provisioning type workflow.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p><b>Sequential data set</b> Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p><b>Partitioned data set</b></p> <ul style="list-style-type: none"> <li>• If the action definition file is in the same data set as the file template: just the member name.</li> <li>• If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.</li> </ul>
<b>workflow-variable-input-file</b>	String	Optional	Standard	<p>Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>

Table 126. Request content for the software services template request (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>composite-variable-input-file</b>	String	Optional	Composite	<p>Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/composite.properties</p> <p>The variable names are in the following format: &lt;standard-template-name&gt;.&lt;atcreate-variable&gt;</p> <p>For example: CICS.startup=10</p> <p>If the file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables that associated with the member standard template. Values are not validated.</p>
<b>approvals</b>	Array of strings	Optional	Standard, Composite	An array of strings representing user IDs of users that are responsible for approving the template.
<b>workflow-clean-after-provisioned</b>	Boolean	Optional	Standard	This field is ignored. The workflows-disposition field should be referenced instead. If the workflows-disposition field is not provided, its default value of archive is used.
<b>consumer-documentation-file</b>	String	Optional	Standard, Composite	Location of a file that provides information for consumers about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
<b>consumer-documentation-type</b>	String	Optional	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
<b>admin-documentation-file</b>	String	Optional	Standard, Composite	Location of a file that provides information for administrators about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
<b>admin-documentation-type</b>	String	Optional	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.

Table 126. Request content for the software services template request (continued)

Field name	Type	Required or optional	Valid for Template Type	Description
<b>workflows-disposition</b>	String	Optional	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.  The default is archive.  If this field is not provided the default value of archive is used. The workflow-clean-after-provisioned field is ignored.
<b>jobs-disposition</b>	String	Optional	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.  The default is keep.
<b>instances-disposition</b>	String	Optional	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.  The default is keep.

Table 127. Composite-definition structure

Field	Type	Required/ optional	Description
<b>sequence</b>	Integer	Required	The order in which to provision the templates, starting with 1. For deprovisioning, the order is reversed.
<b>number-of-instances</b>	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
<b>published-template-name</b>	String	Required	The name of an existing published template in the domain that is associated with the composite template.
<b>connectors</b>	Array of objects	Optional	An array of connector object.  Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above.  See <a href="#">Table 128 on page 183</a> .

Table 128. Connector object

Field	Type	Required/ optional	Description
<b>variable-name</b>	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
<b>source-template</b>	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
<b>source-variable-name</b>	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 183](#).

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. [Table 129 on page 183](#) lists the fields in the JSON object.

Table 129. Response from a create new version of a software services template request

Field	Type	Description
<b>generated-name</b>	String	The generated name associated with this software services template.
<b>object-id</b>	String	The object ID of the newly created software services template. The object ID is to be used on further requests to the session.
<b>object-uri</b>	String	The object URI of the newly created software services template.
<b>SAF-resources</b>	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 130 on page 184</a> .

Table 130. Response from a create request: SAF-resource object

Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

If a failure occurs, the response body contains a JSON object with a description of the error.

## Example HTTP interaction

In Figure 65 on page 184, a request is submitted to create a new version of a software services template on the system SY1.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b28463/actions/create_new_version
{
  "workflow-definition-file":"/u/wfSuspend/p.xml",
  "action-definition-file":"/u/wfSuspend/a.xml",
  "approvals":["zosmft1", "zosmfad"]
}
```

Figure 65. Sample request to create a new version of a software services template

```
{
  "generated-name": "s2.3.default",
  "object-id": "3f8ca645-f872-42b6-b0fc-3c6a9e470fcc",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/3f8ca645-f872-42b6-b0fc-3c6a9e470fcc",
  "SAF-resources": [
    {
      "description": "Controls which users are template approvers for the s2 template in the default domain.",
      "ids": [
        "zosmfad",
        "zosmft1"
      ],
      "groups": [],
      "role": "Template Approver",
      "resource-class": "ZMFCLLOUD",
      "resource-name": "IZUDFLT.ZOSMF.TEMPLATE.APPROVERS.IYU0.s2",
      "required-access": "SAF_READ",
      "other-required-ids": [],
      "audit-requirements": ""
    }
  ]
}
```

Figure 66. Sample response body

The example in Figure 63 on page 177 shows a request to create a new version of a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/5f746dfc-ad24-4355-99d3-b83466ce4492/
actions/create_new_version

{
  "description": "This is an updated version of a composite template that brings up services s1 and s2.",
  "composite-definition": [
    {
      "sequence": "1",
      "number-of-instances": 1,
      "published-template-name": "s1",
      "connectors": []
    }, {
      "sequence": "2",
      "number-of-instances": 1,
      "published-template-name": "s2",
      "connectors": [
        {
          "variable-name": "WELSHIE",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        }
      ]
    }
  ]
}
```

Figure 67. Sample request to create a new version of a composite software services template

The following is the response body for the request.

```
{
  "generated-name": "S1_S2.2.default",
  "object-id": "6f72b8d9-26ae-4552-9f4b-11eadff2225e",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/6f72b8d9-26ae-4552-9f4b-11eadff2225e",
  "SAF-resources": []
}
```

## Create a new software services template based on an existing one

You can use this operation to create a new software services template based on one that already exists, with the same source files. This operation is not valid for composite templates.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/create_based_on
```

In this request

**<object-id>**

Identifies the existing software services template.

**<version>**

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

## Query parameters

None.

## Description

This operation creates a new software services template in the catalog, based on the existing software services template identified by the object ID. It has the same source files (workflow definition, action definition, variable input, and documentation).

On successful completion, HTTP status code 201 (Normal) is returned, indicating that the request resulted in the creation of a new version of a software services template. A response body is provided, as described in [“Response content” on page 187](#).

## Request content

The request content is expected to contain a JSON object. See [Table 131 on page 186](#).

Table 131. Request content for the software services template request			
Field name	Type	Required or optional	Description
<b>name</b>	String	Required	Descriptive name for the software services template. The name must be unique, no longer than 48 characters, and consist of alphanumeric characters (A-Z, a-z, and 0-9), national characters (\$@), underscore (_), and hyphen (-).
<b>domain-name</b>	String	Varies	Name of the domain. Required if the user ID has administrator privileges to more than one domain.
<b>approvals</b>	Array of strings	Optional	An array of strings representing the user IDs that are responsible for approving the template.
<b>target-copy-path</b>	String	Required	The absolute path name of an empty z/OS UNIX directory. The source file contents of the existing software services template are copied into this location, and the new template is created based on that content. If the directory does not exist, it is created. However, the parent directory must already exist.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 187](#).

## Response content

On successful completion, the service returns a response body, which contains a JSON object. [Table 132 on page 187](#) lists the fields in the JSON object.

Table 132. Response from a successful request		
Field	Type	Description
<b>generated-name</b>	String	The generated name associated with this software services template.
<b>object-id</b>	String	The object ID of the newly created software services template. The object ID is to be used on further requests to the session.
<b>object-uri</b>	String	The object URI of the newly created software services template.
<b>SAF-resources</b>	Array of objects	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 133 on page 187</a> .

Table 133. Response from a create request: SAF-resource object		
Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

If a failure occurs, the response body contains a JSON object with a description of the error.

## Example HTTP interaction

In [Figure 68 on page 188](#), a request is submitted to create a new version of a software services template on the system SY1.

```
POST /zosmf/provisioning/rest/1.0/scc/0389ed37-fe13-4176-af65-c171b6ba6b37/actions/create_based_on
HTTP/1.1

{
  "name" : "config2",
  "target-copy-path": "/users/gg/zosmf/newConfig2"
}
```

Figure 68. Sample request to create a new software services template based on an existing one, with request body

```
{
  "generated-name": "mqUpgrade.1.default"
  "object-id": "cd00fb41-20ed-4133-b985-52e28edfcfd0"
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/cd00fb41-20ed-4133-b985-52e28edfcfd0",
  "SAF-resource": []
}
```

Figure 69. Sample response body

## Modify a software services template

You can use this operation to modify fields in a software services template in the catalog.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>
```

In this request

#### <object-id>

Identifies the software services template to be modified.

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation modifies fields in a software services template in the catalog, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 188.

On successful completion, HTTP status code 204 (Normal) is returned, indicating that the request resulted in a modified software services template.

The software services template must be in one of the draft states.

Modifying any of the definition files causes all approvals to be reset.

### Request content

The request content is expected to contain a JSON object that describes the fields to be modified. See [Request content for the software services template request](#).

Table 134. Request content for a request to modify a software services template

Field name	Type	Valid for Template Type	Description
<b>composite-definition</b>	Array of objects	Composite	An array of objects that define the composite template. See <a href="#">Table 135 on page 192</a> .
<b>composite-cluster</b>	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
<b>action-definition-file</b>	String	Standard	<p>Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/actions.xml.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p><b>Sequential data set</b> Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p><b>Partitioned data set</b></p> <ul style="list-style-type: none"> <li>• If the action definition file is in the same data set as the file template: just the member name.</li> <li>• If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.</li> </ul>
<b>description</b>	String	Standard, Composite	Description of the software services template.

Table 134. Request content for a request to modify a software services template (continued)

Field name	Type	Valid for Template Type	Description
<b>workflow-definition-file</b>	String	Standard	<p>Location of the workflow definition file, the primary XML file that defines the workflow.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/workflow_sample_automation.xml.</p> <p>The primary workflow must be a provisioning type workflow.</p> <p>File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set. The data set path is as follows.</p> <p><b>Sequential data set</b> Two forward slashes (//) followed by the fully qualified sequential data set name.</p> <p><b>Partitioned data set</b></p> <ul style="list-style-type: none"> <li>• If the action definition file is in the same data set as the file template: just the member name.</li> <li>• If the action definition file is not in the same data set as the file template: two forward slashes (//) followed by the fully qualified partitioned data set name.</li> </ul>
<b>workflow-variable-input-file</b>	String	Standard	<p>Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.</p>

Table 134. Request content for a request to modify a software services template (continued)

Field name	Type	Valid for Template Type	Description
<b>composite-variable-input-file</b>	String	Composite	<p>Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.</p> <p>Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name. For example, specify /usr/lpp/zosmf/samples/composite.properties</p> <p>The variable names are in the following format: &lt;standard-template-name&gt;.&lt;atcreate-variable&gt;</p> <p>For example: CICS.startup=10</p> <p>If the file includes any variables that are associated with standard templates that are not members of the composite, those variables are ignored. All other variable names are validated to ensure they are atCreate variables that associated with the member standard template. Values are not validated.</p>
<b>workflow-clean-after-provisioned</b>	Boolean	Standard	This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.
<b>consumer-documentation-file</b>	String	Standard, Composite	Location of a file that provides information for consumers about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
<b>consumer-documentation-type</b>	String	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified with a value that is not null.
<b>admin-documentation-file</b>	String	Standard, Composite	Location of a file that provides information for administrators about the template. Specify the fully qualified z/OS UNIX path of the file, beginning with the forward slash (/) and including the file name.
<b>admin-documentation-type</b>	String	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified with a value that is not null.

Table 134. Request content for a request to modify a software services template (continued)

Field name	Type	Valid for Template Type	Description
<b>approvals</b>	Array of strings	Standard, Composite	General approvals that are associated with the template. Each string represents the user ID of a general approval. If the array contains a user ID for a general approval that already exists for the template, the status and all of the corresponding information for that user ID is maintained for the template. If the array does not contain a user ID for a general approval that already exists for the template, that user ID is removed from the general approval list for the template. An empty array removes any existing general approvals from the template. A null value for approvals results in no changes.
<b>workflows-disposition</b>	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.  The default is archive.  If this field is not provided the default value of archive is used. The workflow-clean-after-provisioned field is ignored.
<b>jobs-disposition</b>	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.  The default is keep.
<b>instances-disposition</b>	String	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.

Table 135. Composite-definition structure

Field	Type	Required/ optional	Description
<b>sequence</b>	Integer	Required	The order in which to provision the templates, starting with 1. For deprovisioning, the order is reversed.
<b>number-of-instances</b>	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
<b>published-template-name</b>	String	Required	The name of an existing published template in the domain that is associated with the composite template.
<b>connectors</b>	Array of objects	Optional	An array of connector object.  Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above.  See <a href="#">“Modify a software services template” on page 188</a> .

Table 136. Connector object

Field	Type	Required/optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

## Example HTTP interaction

[Figure 70 on page 193](#) shows a request to modify a software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/6e39d092-745a-4e81-8b7b-d3e1327ad230
{
  "workflow-variable-input-file":"/u/wfStandard/p.props"
}
```

Figure 70. Sample request to modify a software services template

[Figure 71 on page 194](#) shows a request to modify a composite software services template.

POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/f98fb327-6714-420d-adc5-89793e7060d8

```
{
  "composite-definition": [
    {
      "sequence": "1",
      "number-of-instances": 1,
      "published-template-name": "s1",
      "connectors": []
    }, {
      "sequence": "2",
      "number-of-instances": 1,
      "published-template-name": "s2",
      "connectors": [
        {
          "variable-name": "WELSHIE",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        },
        {
          "variable-name": "INS",
          "source-template": "s1",
          "source-variable-name": "registry-instance-Name"
        }
      ]
    }
  ]
}
```

Figure 71. Sample request to modify a composite software services template

## Delete a software services template

You can use this operation to delete a software services template from the catalog.

### HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>
```

In this request

#### <object-id>

Identifies the software services template to be deleted.

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation deletes a software services template from the catalog.

On successful completion, HTTP status code 200 Ok is returned, indicating that the request resulted in a software services template being deleted.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 Ok is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. [Table 137 on page 195](#) lists the fields in the JSON object.

Table 137. Response from the software services template request		
Field	Type	Description
<b>composites-affected</b>	Collection of objects	Collection of composite-affected objects that shows the composite templates that were affected by this action. It is returned only if the action was performed for a standard software services template.

Table 138. Composite-affected object		
Field	Type	Description
<b>name</b>	String	Name of the composite template that was affected by an update to a standard template.
<b>state</b>	String	State of the composite template. For example, the state changes from published to missing_required_member when a standard published template is no longer available to satisfy the member requirement.

## Example HTTP interaction

[Figure 72 on page 195](#) shows a request to delete a software services template.

```
DELETE https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/33d4171f-b759-4926-8a48-e359e589474a
```

*Figure 72. Sample request to delete a software services template*

The response body is as follows.

```
{
  "composites-affected": null
}
```

## Get a software services template

Use this operation to retrieve a software services template from the catalog.

## HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>
```

In this request:

**<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

**<object-id>**

Identifies the software services template to retrieve.

**Query parameters**

None.

**Description**

This operation retrieves a software services template from the catalog.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 196](#).

**Request content**

None.

**Authorization requirements**

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:  
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

**HTTP status codes**

On successful completion, HTTP status code 200 (OK) is returned.

**Response content**

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. [Table 139 on page 196](#) lists the fields in the JSON object.

<i>Table 139. Response from a request to get a software services template.</i>			
Field	Type	Valid for Template Type	Description
<b>base-object-id</b>	String	Standard, Composite	The object ID that is associated with all of the versions of the software services template.
<b>generated-name</b>	String	Standard, Composite	Generated name for the software services template.
<b>name</b>	String	Standard, Composite	The name associated with the software services template.
<b>version</b>	String	Standard, Composite	Version of the software services template.
<b>owner</b>	String	Standard, Composite	User ID of the software services template owner.

Table 139. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
<b>state</b>	String	Standard, Composite	Indicates the status of the software services template. See <a href="#">“State values” on page 201</a> .
<b>description</b>	String	Standard, Composite	Description of the software services template.
<b>tenants</b>	Array of Strings	Standard, Composite	Each string represents a tenant that the template is associated with.
<b>domain-shared-tenants</b>	Array of Strings	Standard, Composite	Each string represents a tenant in the domain that the template is associated with through the domain shared resource pool.
<b>domain-name</b>	String	Standard, Composite	The domain the template is associated with.
<b>approvals</b>	Array of objects	Standard, Composite	Array of Approval-Object containing information about the approvals associated with this software services template. See <a href="#">Table 140 on page 202</a> .
<b>action-definition-file</b>	String	Standard	Location of the action definition file.
<b>action-definition-file-original-source</b>	String	Standard	Original user specified location of the action definition file
<b>action-definition-file-original-timestamp</b>	String	Standard	Last-modified time stamp for when the original action definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>actions</b>	Array of objects	Standard	Array of Action-Object containing information about the actions associated with the template. See <a href="#">Table 141 on page 203</a> .
<b>software-id</b>	String	Standard	A short, arbitrary, value that identifies the software that is being provisioned.
<b>software-name</b>	String	Standard	Name of the software that is being provisioned.
<b>software-type</b>	String	Standard	Type of software that is being provisioned.
<b>software-version</b>	String	Standard	Version of the software that is being provisioned.
<b>workflow-definition-file</b>	String	Standard	Location of the workflow definition file, the primary XML file that defines the workflow
<b>workflow-definition-file-original-source</b>	String	Standard	Original user-specified location of the workflow definition file.

Table 139. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
<b>workflow-definition-file-original-timestamp</b>	String	Standard	The last-modified time stamp for when the original workflow definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>workflow-id</b>	String	Standard	A short, arbitrary value that identifies the workflow.
<b>workflow-vendor</b>	String	Standard	Name of the vendor that provided the workflow definition file.
<b>workflow-version</b>	String	Standard	Version of the workflow definition file.
<b>composite-variable-input-file</b>	String	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.  The variable names are in the following format: <standard-template-name>.<atcreate-variable>  For example: CICS.startup=10
<b>composite-variable-input-file-original-source</b>	String	Composite	Location of the composite variable input file, an optional properties file used to specify in advance the values for one or more of the atCreate variables.
<b>composite-variable-input-file-original-timestamp</b>	String	Composite	The last-modified time stamp for when the original composite variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>workflow-variable-input-file</b>	String	Standard	Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.
<b>workflow-variable-input-file-original-source</b>	String	Standard	The original user-specified location of the workflow variable input file.
<b>workflow-variable-input-file-original-timestamp</b>	String	Standard	The last-modified time stamp for when the original variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>workflow-clean-after-provisioned</b>	Boolean	Standard	This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.

Table 139. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
<b>prompt-variables</b>	Array of objects	Standard	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template. See Table 143 on page 204.
<b>public-variables</b>	Array of strings	Standard	Array of strings that name the public variables for the template.
<b>at-create-variables</b>	Array of strings	Standard, Composite	<p>Array of strings that name the variables that are either prompt variables (variables that are expected to be prompted for in preparation for running the software services template), or required variables (variables for which a value is required when the software services template is run), or both.</p> <p>For a composite type template, any atCreate variable that was designated as a connector variable is excluded from the list. The variables are prefixed by the standard template name, for example MQ.BRING_UP.</p>
<b>consumer-documentation-file</b>	String	Standard, Composite	Location of the original file that provides information for consumers about the template.
<b>consumer-documentation-file-original-source</b>	String	Standard, Composite	The original user-specified location of the consumer documentation file.
<b>consumer-documentation-type</b>	String	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
<b>admin-documentation-file</b>	String	Standard, Composite	Location of a file that provides information for administrators about the template.
<b>admin-documentation-file-original-source</b>	String	Standard, Composite	The original user-specified location of the admin documentation file.
<b>admin-documentation-type</b>	String	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.
<b>create-time</b>	String	Standard, Composite	Time that this object was created, in ISO 8601 format.
<b>create-by-user</b>	String	Standard, Composite	User who created this object.
<b>last-modified-time</b>	String	Standard, Composite	The last time this object was updated, in ISO 8601 format.

Table 139. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
<b>last-modified-by-user</b>	String	Standard, Composite	User who last updated this object.
<b>published-timestamp</b>	String	Standard, Composite	<p>The last time this template was moved to published state. If the template was never in the published state, an empty string is returned. ISO 8601 format.</p> <p>When you upgrade to the April, 2017 deliverable of z/OSMF, the published-timestamp for a template already in a published state is set to the timestamp of the upgrade.</p>
<b>archived-timestamp</b>	String	Standard, Composite	<p>The last time this template was moved to the archived state. If the template was never in the archived state, an empty string is returned. ISO 8601 format.</p> <p>When you upgrade to the April, 2017 deliverable of z/OSMF, the archived-timestamp for a template already in an archived state is set to the timestamp of the upgrade.</p>
<b>workflows-disposition</b>	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.
<b>jobs-disposition</b>	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.
<b>instances-disposition</b>	String	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.
<b>automatic-security</b>	String	Standard, Composite	<p>Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization:</p> <ul style="list-style-type: none"> <li>• true if the domain that this template is associated with is set up for automatic authorization</li> <li>• false if the domain that this template is associated with is set up for manual authorization.</li> </ul>
<b>SAF-resources</b>	Array of objects	Standard, Composite	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 144 on page 205</a> .
<b>runAsUsers</b>	Array of objects	Standard	Array of RunAsUser objects containing information about runAsUser IDs that are referenced by this template. See <a href="#">Table 145 on page 205</a> .

Table 139. Response from a request to get a software services template. (continued)

Field	Type	Valid for Template Type	Description
<b>runAsUser-audit</b>	boolean	Standard	Indicates if auditing is performed on workflows and action commands that are associated with the template. This field cannot be updated and is based on the level of the Cloud Provisioning plug-in at the time that the template is created.  <b>false</b> runAsUser auditing is performed. This value is used for all templates created prior to the April, 2017 delivery.  <b>true</b> runAsUser auditing is not performed. This value is used for all templates created beginning with the April, 2017 delivery.
<b>template-type</b>	String	Standard, Composite	Identifies the type of template.  <b>standard</b> Defines a single software service.  <b>composite</b> Consists of multiple published templates that will be provisioned together.
<b>composite-cluster</b>	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
<b>composite-definition</b>	Array of objects	Composite	An array of objects that define the composite template. See <a href="#">Table 146 on page 206</a> .
<b>composite-parents</b>	Array of strings	Standard	An array of strings. Each string is a composite template that includes this standard template. For example:  <pre>[c0e4d08f-f046-4a79-8a15-6981743d07ed, c0e4d08f-f046-4a79-8a15-6981743d07e3, c0e4d08f-f046-4a79-8a15-6981743d07ed]</pre>
<b>security-wf-info</b>	Object	Standard, Composite	An object that describes the security workflow, when automatic security is in effect and an attempt to grant authorization to a runAsUser ID or approver ID failed. See <a href="#">Table 148 on page 208</a> .
<b>provisioning-version</b>	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.

## State values

### archived

The entry is hidden from consumers. You can make it available again with the **Publish** action.

### corrupted

The contents of the software services template are missing or incorrect. Delete the template.

**draft**

The entry is in the edit state and visible only to the owner and the administrator. No approvals are required for this template. The entry can be tested with the Test a software services template API.

**draft\_approved**

The software services template is in edit state and all the approvals that are associated with the template and the respective runAs user IDs have been received. The entry can be tested with the Test a software services template API.

**draft\_pending\_approvals**

The software services template is in edit state and one or more associated approvals has not been approved. The entry cannot be tested (with the Test a software services template API) in this state.

**draft\_missing\_required\_approver**

One or more of the definition files contains a runAsUser element without a corresponding approver element. Either an approver element must be added for the runAsUser element, or a domain or general approver must be added for the software services template. The entry cannot be tested (with the Test a software services template API) in this state.

**draft\_rejected**

The template is in a draft state and one or more approvers rejected an approval. The entry cannot be tested (with the Test a software services template API) in this state.

**missing\_required\_member**

One or more of the members of a composite template that was referenced in the composite definition is not available. This state applies only to published or archived composite templates.

**pending\_security\_update**

Permission to access the software services template is being processed. No API requests are allowed for the software services template until the security processing is complete.

**published**

The entry is locked and visible to consumers.

**security\_update\_failed**

Security access setup related to the software services template failed. Only the view and delete API requests are available.

Table 140. Response from a get request: Approval-Object		
Field	Type	Description
<b>status</b>	String	Status of the approval for this object: pending, approved, or rejected.
<b>comment</b>	String	Comment associated with the change in status from pending to either approved or rejected.
<b>description</b>	String	Additional detail that is provided if the approval is for a workflow definition that is associated with the action definition, for example, This workflow definition is associated with the <action-name> action.
<b>approvers</b>	Array of strings	Each string in the array is a user ID or SAF group that can approve the template, workflow step, or action. Any one of the user IDs in the array can approve or reject. The last action takes precedence.
<b>status-update-by</b>	String	User ID that performed the last approve or reject action for this approval object.
<b>time-of-update</b>	String	The last time this object was updated, in ISO 8601 format.
<b>run-as-user</b>	String	The runAsUser user ID that the approval object is for. Only applicable when the type is action_definition or step_definition.

Table 140. Response from a get request: Approval-Object (continued)		
Field	Type	Description
<b>type</b>	String	Type of approval object: general, domain, action_definition, or step_definition.
<b>object-id</b>	String	Unique object ID representing this approval object.
<b>workflow-file</b>	String	Workflow file definition associated with this runAsUser user ID.
<b>variable-input-file</b>	String	Variable input file associated with this runAsUser user ID.
<b>step-name</b>	String	Workflow file definition step associated with this runAsUser user ID.
<b>called-by-step-name</b>	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>called-by-workflow-file</b>	String	Workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>actions-file</b>	String	Actions file definition associated with this runAsUser user ID.
<b>action-name</b>	String	Action that is defined in the actions file associated with this runAsUser user ID.
<b>run-as-user-dynamic</b>	boolean	Indicates if the run-as-user ID value can change:  <b>true</b> The run-as-user ID value is not final and can change during the processing of the workflow  <b>false</b> The run-as-user ID is final and cannot change during the processing of the workflow.

Table 141. Response from a get request: Action-Object	
Field	Type
<b>name</b>	String
<b>type</b>	String
<b>is-deprovision</b>	String. The value must be either true or false.
<b>command</b>	String
<b>command-run-as-user</b>	String
<b>command-sol-key</b>	String
<b>command-unsol-key</b>	String
<b>command-detect-time</b>	String
<b>workflow-definition-file</b>	String
<b>workflow-variable-input-file</b>	String
<b>workflow-variables</b>	Variable[]
<b>instructions</b>	String

Table 141. Response from a get request: Action-Object (continued)

Field	Type
<b>prompt-variables</b>	String. The prompt variable objects that are associated with the action.
<b>workflow-clean-after-complete</b>	String. The value must be true, false, or inherit. For workflow type actions, if provided, this indicates whether the workflow instance is cleaned up after completion. If workflow-clean-after-complete is not provided, the default is inherit, and the value is inherited from the workflows-disposition field.
<b>command-run-as-user-dynamic</b>	boolean. Indicates if the command-run-as-user ID value can change:  <b>true</b> The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content.  <b>false</b> The command-run-as-user ID is final and cannot change during the processing of the command.

Table 142. Response from a get request: Variable-Object

Field	Type
<b>name</b>	String
<b>value</b>	String
<b>visibility: public or private</b>	String

Table 143. Response from a get request: Prompt-Variable-Object

Field	Type	Description
<b>name</b>	String	Name of the property.
<b>value</b>	String	Current value for the property.
<b>required</b>	boolean	Indicates whether the variable value is required during the workflow create process.
<b>label</b>	String	Short label for the UI widget.
<b>description</b>	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
<b>abstract</b>	String	Brief description of the variable for the UI widget.
<b>type</b>	String	Type of the variable element: boolean, string, integer, decimal, time, date.
<b>must-be-choice</b>	boolean	Indicates whether the value must come from the provided choices.
<b>choices</b>	Array of Strings	Contains allowable choices for the value of the variable.
<b>regex</b>	String	Standard regular expression that constrains the variable value.
<b>multi-line</b>	boolean	Indicates whether the value requires a multi-line text box.

Table 143. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
<b>min</b>	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
<b>max</b>	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
<b>places</b>	String	Maximum number of decimal places that can be specified for a variable of type decimal.
<b>error-message</b>	String	Default error message associated with an incorrect value.

Fields of type String default to null.

Table 144. Response from a create request: SAF-resource object

Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

Table 145. Response from a get request: RunAsUser object

Field	Type	Description
<b>description</b>	String	Additional detail provided if the run-as-user is for a workflow definition that is associated with the action definition. Example: This workflow definition is associated with the <action-name> action.
<b>approver-user-ids</b>	Array of Strings	Array of strings where each string is a user ID that originates from the approver element that is associated with the runAsUser for the template step or action.

Table 145. Response from a get request: RunAsUser object (continued)

Field	Type	Description
<b>run-as-user</b>	String	The runAsUser user ID that the approval object is for. This is applicable only to action_definition and step_definition type.
<b>type</b>	String	One of the following: action_definition or step_definition
<b>workflow-file</b>	String	The workflow file definition that is associated with this runAsUser user ID.
<b>variable-input-file</b>	String	The variable input file that is associated with this runAsUser user ID.
<b>step-name</b>	String	The workflow file definition step that is associated with this runAsUser user ID.
<b>called-by-step-name</b>	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the step in the parent workflow definition that called the workflow definition file that generated the approval object.
<b>called-by-workflow-file</b>	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the workflow definition that called the workflow definition file that generated the approval object.
<b>actions-file</b>	String	The actions file definition that is associated with this runAsUser user ID.
<b>action-name</b>	String	The action defined in the actions file that is associated with this runAsUser user ID.
<b>run-as-user-dynamic</b>	boolean	Indicates if the run-as-user ID value can change: <b>true</b> The run-as-user ID value is not final and can change during the processing of the workflow <b>false</b> The run-as-user ID is final and cannot change during the processing of the workflow.

Table 146. Response from a get request: Composite-definition object

Field	Type	Description
sequence	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Indicates the number of child instances to be created using the template in a composite cluster. This field is within each entry of the composite-definition.
published-template-name	String	The name of an existing published template in the domain that is associated with the composite template.

Table 146. Response from a get request: Composite-definition object (continued)

Field	Type	Description
connectors	Array of objects	<p>An array of connector object.</p> <p>Allowed for provisioning of published templates that are higher than sequence 1, that is, 2 and above.</p> <p>See <a href="#">Table 147</a> on page 207.</p>
prompt-variables	Array of objects	<p>Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template.</p> <p>If specified, this overrides the array of prompt variables that are associated with the template specified with published-template-name. Only prompt variables that are already specified for the published-template-name can be specified. An empty array will translate into not prompting for any variables. If this field is not provided or set to null, then the prompt variables that are associated with published-template-name are used.</p> <p>If the connector variable-name is also a prompt-variable, then the connector takes precedence, and the variable is not promptable.</p>
missing	boolean	<ul style="list-style-type: none"> <li>• true if no published template is available that is related to the original version used when the template was defined</li> <li>• false if a published template exists that satisfies the published template requirement</li> </ul>

Table 147. Connector object

Field	Type	Required/optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.

Table 147. Connector object (continued)

Field	Type	Required/ optional	Description
not-valid	boolean	Required	Indicates if the information (variable-name, source-template, and source-variable-name values) in this connector is valid. The value is: <ul style="list-style-type: none"> <li>false, if all of the information is accurate</li> <li>true, if one or more of the values are incorrect.</li> </ul>

Table 148. Response from a get request: Security Workflow Information

Field	Type	
wf-status	String	Status of the workflow
wf-uri	String	URI of the workflow
wf-key	String	Key of the workflow
additional-info	String	Additional information about the error

## Example HTTP interaction

In Figure 73 on page 208, a request is submitted to retrieve a standard software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/5ccbad22-94fd-4b31-bb2b-95aa8602cc48
```

Figure 73. Sample request to retrieve a standard software services template

The following is the response body for the example GET request.

```
{
  "name": "mqCBA",
  "version": "1",
  "owner": "domadmin",
  "state": "published",
  "description": "This workflow provisions an MQ for z/OS Queue Manager",
  "tenants": [...],
  "actions": [...],
  "approvals": [],
  "tested": false,
  "generated-name": "mqCBA.1.default",
  "domain-name": "default",
  "action-definition-file": "definition/qmgrActions.xml",
  "action-definition-file-original-source": "/users/gg/mqCBA/definition/qmgrActions.xml",
  "action-definition-file-original-timestamp": "2016-11-18T20:00:42Z",
  "software-id": "5655-W97",
  "software-name": "IBM MQ for z/OS",
  "software-type": "QMGr",
  "software-version": "V8.0.0",
  "workflow-definition-file": "definition/provision.xml",
  "workflow-definition-file-original-source": "/users/gg/mqCBA/definition/provision.xml",
  "workflow-definition-file-original-timestamp": "2016-11-18T20:03:47Z",
  "workflow-id": "ProvisionQueueManager",
  "workflow-vendor": "IBM",
  "workflow-version": "1.0.1",
  "workflow-variable-input-file": "definition/workflow_variables.properties",
  "workflow-variable-input-file-original-source":
    "/users/gg/mqCBA/definition/workflow_variables.properties",
  "workflow-variable-input-file-original-timestamp": "2016-11-18T20:00:42Z",
  "prompt-variables": [],
  "public-variables":
    ["CSQ_CHIN_SERVICE_CLASS_NAME", "CSQ_MSTR_SERVICE_CLASS_NAME", "CSQ_TCPIP_PORT_NUMBER",
     "CSQ_AUTO_GEN_CMD_PFX_SSID", "CSQ_CMD_PFX_FOR_AUTO_GEN", "CSQ_CHIN_REPORT_CLASS_NAME",
     "CSQ_MSTR_CLASSIFICATION_RULE_ID", "CSQ_MSTR_REPORT_CLASS_NAME", "CSQ_CMD_PFX", "CSQ_QSGDISP",
```

```

"CSQ_CHIN_CLASSIFICATION_RULE_ID", "CSQ_TCPIP_STATUS_CODE", "CSQ_TARG_LIB_HLQ", "CSQ_SSID",
"CSQ_TCPIP_PORT_ID", "CSQ_LANG_LETTER", "CSQ_ENVIRONMENT"
]

"at-create-variables": [],
"workflow-clean-after-provisioned": true,
"security-wf-info": null,
"create-time": "2016-11-18T20:00:43.504Z",
"created-by-user": "domadmin",
"last-modified-by-user": "domadmin",
"last-modified-time": "2016-11-18T20:04:50.913Z",
"admin-documentation-file-original-source": "/users/gg/mqCBA/documentation/admin-
mqaas_readme.pdf",
"admin-documentation":
"/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
admin",
"admin-documentation-type": "pdf",
"consumer-documentation-file-original-source":
"/users/gg/mqCBA/documentation/consumer-workflow_variables.properties",
"consumer-documentation":
"/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
consumer",
"consumer-documentation-type": "text",
"base-object-id": "c0e4d08f-f046-4a79-8a15-6981743d07ed",
"admin-documentation-mime-type": "application/pdf",
"consumer-documentation-mime-type": "text/plain",
"SAF-resources": [],
"runAsUsers": [],
"runAsUser-audit": true,
"automatic-security": true,
"published-timestamp": "2017-04-05T16:16:55.878Z",
"archived-timestamp": "",
"provisioning-version": "1400"
}

```

Figure 74 on page 209 shows a request to retrieve a composite software services template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b28463
```

Figure 74. Sample request to retrieve a composite software services template

The following is the response body for the GET request for the composite software services template.

```

{
  "name": "S1_S2",
  "owner": "ibmuser",
  "state": "missing_required_member",
  "description": "This is a composite template that brings up services s1 and s2. The s2
services
  references information from the s1 services to satisfy its run-time properties.",
  "tenants": [],
  "approvals": [],
  "tested": false,
  "version": "1",
  "create-time": "2017-11-11T15:14:09.701Z",
  "created-by-user": "ibmuser",
  "last-modified-by-user": "ibmuser",
  "last-modified-time": "2017-11-11T15:22:17.595Z",
  "generated-name": "S1_S2.1.default",
  "domain-name": "default",
  "SAF-resources": [],
  "automatic-security": true,
  "published-timestamp": "2017-11-11T15:22:17.595Z",
  "archived-timestamp": "2017-11-11T15:25:08.396Z",
  "template-type": "composite",
  "composite-cluster": false,
  "composite-definition": [
    {
      "sequence": 1,
      "number-of-instances": 1,
      "connectors": [],
      "missing": true,
      "description": "",
      "prompt-variables": [],
      "published-template-name": "s1",
      "software-type": ""
    }
  ],
}

```

```

{
  "sequence": 2,
  "number-of-instances": 1,
  "connectors": [
    {
      "variable-name": "INS",
      "source-template": "s1",
      "source-variable-name": "registry-instance-Name",
      "not-valid": true
    },
    {
      "variable-name": "WELSHIE",
      "source-template": "s1",
      "source-variable-name": "WELSHIE",
      "not-valid": true
    }
  ],
  "missing": false,
  "description": "",
  "prompt-variables": [
    {
      "name": "CMD",
      "label": "CMD",
      "description": "CMD",
      "type": "string",
      "value": "S BCTEST",
      "required": false,
      "choices": null,
      "regex": ".{1,1000000}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "CMD",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    },
    {
      "name": "WELSHIE",
      "label": "name",
      "description": "This variable contains the name of a welsh springer
spaniel.",
      "type": "string",
      "value": "Scout",
      "required": false,
      "choices": null,
      "regex": ".*",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Name of a Welsh Springer Spaniel",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": ".*"
    },
    {
      "name": "INS",
      "label": "INS",
      "description": "INS",
      "type": "string",
      "value": "Instructions",
      "required": false,
      "choices": null,
      "regex": ".{1,1000000}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "INS",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    }
  ],
  "published-template-name": "s2",
  "software-type": "MIX"
}
],
"composite-variable-input-file": "",
"composite-variable-input-file-original-source": "",
"composite-variable-input-file-original-timestamp": "",
"security-wf-info": null,
"admin-documentation-file-original-source": null,

```

```

    "admin-documentation": null,
    "admin-documentation-type": null,
    "consumer-documentation-file-original-source": null,
    "consumer-documentation": null,
    "consumer-documentation-type": null,
    "at-create-variables": [
        "s2.WELSHIE",
        "s2.UKEY",
        "s2.CMD",
        "s2.INS"
    ],
    "base-object-id": "f4feb4e9-f2e3-4121-a483-fa4bf10282b9",
    "admin-documentation-mime-type": null,
    "consumer-documentation-mime-type": null,
    "provisioning-version": "1400"
}

```

## Get a software services template history

Use this operation to retrieve the history for a software services template in the catalog.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/history
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

#### <object-id>

Identifies the template for which history is to be retrieved.

### Query parameters

None.

### Description

This operation retrieves the history for a software services template.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 212](#).

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

## Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the template. Table 149 on page 212 lists the fields in the history object.

Table 149. Response from a get request: History object

Field	Type	Valid for Template Type	Description
<b>action-type</b>	String	Standard, Composite	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"><li>• Create</li><li>• Add approval</li><li>• Approve</li><li>• Archive</li><li>• Modify</li><li>• Publish</li><li>• Refresh</li><li>• Reject</li><li>• Remove approval</li><li>• Run</li><li>• Test run</li><li>• Security complete</li><li>• Update approval</li></ul>
<b>user</b>	String	Standard, Composite	The user who performed the action.
<b>action-time</b>	String	Standard, Composite	The time that the action was taken.
<b>action-details</b>	String	Standard, Composite	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

## Example HTTP interaction

In Figure 75 on page 212, a request is submitted to retrieve the history for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/c06b4ba7-f72a-491e-8d63-5a38b4a4e4a3/history
```

Figure 75. Sample request to retrieve a software template history

The following is the response body for the get request in this example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:15.791Z",
      "action-details": "Created template"
    }
  ],
}
```

```
{
  "action-type": "Publish",
  "user": "ibmuser",
  "action-time": "2020-12-14T14:41:24.860Z",
  "action-details": "Published template"
}
```

## Get software services template documentation

Use this operation to retrieve software services template documentation from the catalog.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/admin
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/documentation/consumer
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

#### <object-id>

Identifies the software services template to retrieve.

#### documentation/admin

Causes the administrator documentation file to be retrieved.

#### documentation/consumer

Causes the consumer documentation file to be retrieved.

### Query parameters

None.

### Description

This operation retrieves software services template documentation from the catalog.

On successful completion, the operation returns HTTP status code 200 (OK), indicating that the request resulted in software services template documentation being retrieved.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

## Response content

The documentation file in the associated mime type.

## Example HTTP interaction

In Figure 76 on page 214, a request is submitted to retrieve the consumer documentation for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/5ccbad22-94fd-4b31-bb2b-95aa8602cc48/documentation/consumer
```

Figure 76. Sample request to retrieve software services template documentation

## Get prompt variables for a software services template

Use this operation to retrieve the variables that are required to run the software services template and for which a prompt can be used to obtain the value.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/prompt-variables
```

---

In this request

#### <object-id>

Identifies the software services template to be retrieved.

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

## Query parameters

None.

## Description

This operation retrieves the variables for which a prompt can obtain the value.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 215](#).

## Request content

None.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the prompt variables. See [Table 150 on page 215](#), , and [Table 151 on page 215](#).

Table 150. Response from a get prompt variables request				
Field	Type	Required/ Optional	Valid for Template Type	Description
<b>prompt-variables</b>	Array of objects	Required	Standard	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template. See <a href="#">Table 151 on page 215</a> .
<b>composite-prompt-variables</b>	Array of objects	Required	Composite	Array of composite prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the composite software services template. See <a href="#">Table 152 on page 216</a> .

Table 151. Response from a get request: Prompt-Variable-Object		
Field	Type	Description
<b>name</b>	String	Name of the property.
<b>value</b>	String	Current value for the property.
<b>required</b>	boolean	Indicates whether the variable value is required during the workflow create process.
<b>label</b>	String	Short label for the UI widget.
<b>description</b>	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
<b>abstract</b>	String	Brief description of the variable for the UI widget.
<b>type</b>	String	Type of the variable element: boolean, string, integer, decimal, time, date.
<b>must-be-choice</b>	boolean	Indicates whether the value must come from the provided choices.
<b>choices</b>	Array of Strings	Contains allowable choices for the value of the variable.
<b>regex</b>	String	Standard regular expression that constrains the variable value.
<b>multi-line</b>	boolean	Indicates whether the value requires a multi-line text box.
<b>min</b>	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.

Table 151. Response from a get request: Prompt-Variable-Object (continued)		
Field	Type	Description
<b>max</b>	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
<b>places</b>	String	Maximum number of decimal places that can be specified for a variable of type decimal.
<b>error-message</b>	String	Default error message associated with an incorrect value.

Fields of type String default to null.

Table 152. Response from a get request: Composite-Prompt-Variable-Object		
Field	Type	Description
<b>published-template-name</b>	String	The name of the published template in the composite template that the prompt-variables field is associated with.
<b>prompt-variables</b>	Array of objects	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the published-template-name software services template as part of the composite software services template. See <a href="#">Table 151 on page 215</a> .

## Example HTTP interactions

Figure 77 on page 216 shows a request to retrieve the prompt variables for a standard template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b2846a/prompt-variables
```

Figure 77. Sample request to retrieve prompt variables

The following is the response body for the request.

```
{
  "prompt-variables": [
    {
      "name": "WELSHIE",
      "label": "name",
      "description": "This variable contains the name of a welsh springer spaniel.",
      "type": "string",
      "value": "Scout",
      "required": false,
      "choices": null,
      "regex": ".*",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Name of a Welsh Springer Spaniel",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": ".*"
    }
  ]
}
```

Figure 78. Response body for the GET prompt variables request

Figure 79 on page 217 shows a request to retrieve the prompt variables for a composite template.

Figure 79. Sample request to retrieve prompt variables, composite template

The following is the response body for the request.

```
{
  "composite-prompt-variables": [
    {
      "prompt-variables": [],
      "published-template-name": "s1"
    },
    {
      "prompt-variables": [
        {
          "name": "CMD",
          "label": "CMD",
          "description": "CMD",
          "type": "string",
          "value": "S BCTEST",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "CMD",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        },
        {
          "name": "WELSHIE",
          "label": "name",
          "description": "This variable contains the name of a welsh springer
spaniel.",
          "type": "string",
          "value": "Scout",
          "required": false,
          "choices": null,
          "regex": ".*",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "Name of a Welsh Springer Spaniel",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": ".*"
        },
        {
          "name": "INS",
          "label": "INS",
          "description": "INS",
          "type": "string",
          "value": "Instructions",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "INS",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        }
      ],
      "published-template-name": "s2"
    }
  ]
}
```

Figure 80. Response body for the GET prompt variables request, composite template

## Get source information for a software services template

Use this operation to retrieve source information for a software services template .

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/sources
```

In this request

#### <object-id>

Identifies the software services template for which information is to be retrieved.

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation retrieves source information for a template, which includes details of the original source paths that were provided, and whether the files have been changed since the last time the template was updated with the source paths.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 218](#).

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

### Response content

On successful completion, the service returns a response body, which contains a JSON object. See [Table 153 on page 218](#).

Table 153. Response from a get source request			
Field	Type		Description
action-definition-file	Source-Info-Object	Standard	Details for the action definition file. See <a href="#">Table 154 on page 219</a> .

Table 153. Response from a get source request (continued)

Field	Type		Description
<b>workflow-definition-file</b>	Source-Info-Object	Standard	Details for the workflow definition file. See <a href="#">Table 154 on page 219</a> .
<b>workflow-variable-input-file</b>	Source-Info-Object	Standard	Details for the workflow variable input file. See <a href="#">Table 154 on page 219</a> .
<b>composite-variable-input-file</b>	Source-Info-Object	Composite	Details for the composite variable input file. See <a href="#">Table 154 on page 219</a> .

Table 154. Response from a get request: Source-Info-Object

Field	Type	Description
<b>original-source-path</b>	String	The original source path provided for the file.
<b>out-of-sync</b>	boolean	Indicates if the file that is associated with the template matches the original source file. The value is false if the current file that is associated with the template matches the original source file, and true if the current file that is associated with the template differs from the original source file, or if the original source file is not found.

## Example HTTP interactions

Figure 81 on page 219 shows a request to retrieve the source information for a standard template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/93d9c9dc-6b47-4222-89a6-f18764b28463/sources
```

Figure 81. Sample request to retrieve source information

The following is the response body for the request.

```
{
  "action-definition-file": {
    "original-source-path": "/u/wfStandard/a.xml",
    "out-of-sync": false
  },
  "workflow-definition-file": {
    "original-source-path": "/u/wfStandard/p.xml",
    "out-of-sync": false
  },
  "workflow-variable-input-file": null
}
```

Figure 82. Response body for the get source request

Figure 83 on page 219 shows a request to retrieve the source information for a composite template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/scc/asdfasdf-asdfasdf-asdfasdf-asdfas/sources
```

Figure 83. Sample request to retrieve source information for a composite template

The following is the response body for the request.

```
{
  "composite-variable-input-file": null
}
```

Figure 84. Response body for the get source request for a composite template

## List the software services templates

You can use this operation to list the software services templates that are defined in the catalog.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/scc
```

---

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software services template service. The following value is valid: `1.0`.

### Query parameters

You can specify the following query parameters on this request. Objects matching all query parameters are returned.

**domain-name**

Optional, specifies the domain name.

**name**

Optional, regular expression, specifies the external name of the software services template.

**owner**

Optional, specifies the user ID or group ID that identifies the owner of the software services template.

**software-type**

Optional, specifies the type of software being provisioned.

**state**

Optional, regular expression, specifies the state.

**template-type**

Optional, specifies the type (standard or composite).

If you specify no query parameters, then all software services templates are returned.

### Description

This operation lists the software services templates in the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a list of software services templates being retrieved. A response body is provided, as described in [“Response content” on page 221](#).

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services templates. See [Table 155 on page 221](#), [Table 156 on page 221](#).

Table 155. Array of objects		
Field	Type	Description
<b>scc-list</b>	Array of objects	Array of software services template objects. The array is filter based on any query parameters that were provided.

Table 156. Fields for each software services template			
Field	Type	Value Returned for Template Type	Description
<b>generated-name</b>	String	Standard, Composite	The generated name for the software services template.
<b>object-id</b>	String	Standard, Composite	The unique ID that identifies the software services template.
<b>base-object-id</b>	String	Standard	The object ID that is associated with all of the versions of the software services template.
<b>name</b>	String	Standard, Composite	Descriptive name for the software services template.
<b>version</b>	String	Standard	Version of the software services template.
<b>owner</b>	String	Standard, Composite	User ID of the software services template owner.
<b>state</b>	String	Standard, Composite	Indicates the status of the software services template. See <a href="#">“State values” on page 223</a> .
<b>description</b>	String	Standard, Composite	Description of the software services template.
<b>action-definition-file</b>	String	Standard	Location of the action definition file, a file in XML format that defines the actions for the software services instance that is provisioned from the template.
<b>software-id</b>	String	Standard	A short, arbitrary value that identifies the software that is being provisioned.
<b>software-name</b>	String	Standard	Name of the software that is being provisioned.
<b>software-type</b>	String	Standard	Identifies the type of software that is being provisioned.
<b>software-version</b>	String	Standard	Version of the software that is being provisioned.
<b>workflow-definition-file</b>	String	Standard	Location of the workflow definition file, the primary XML file that defines the workflow.

Table 156. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
<b>workflow-id</b>	String	Standard	Workflow ID. A short, arbitrary value that identifies the workflow.
<b>workflow-vendor</b>	String	Standard	Name of the vendor that provided the workflow definition file.
<b>workflow-version</b>	String	Standard	Version of the workflow definition file.
<b>workflow-variable-input-file</b>	String	Standard	Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.
<b>domain-name</b>	String	Standard, Composite	The name of the domain that the template resides in.
<b>create-time</b>	String	Standard, Composite	The time that this object was created, in ISO 8601 format.
<b>created-by-user</b>	String	Standard, Composite	The user that created this object.
<b>last-modified-time</b>	String	Standard, Composite	The last time this object was updated, in ISO 8601 format.
<b>last-modified-by-user</b>	String	Standard, Composite	The user that last updated this object.
<b>template-type</b>	String	Standard, Composite	Identifies the type of template: <b>standard</b> Defines a single software service. <b>composite</b> Consists of multiple published templates that are provisioned together.
<b>composite-cluster</b>	boolean	Optional	Indicates whether child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
<b>composite-definition</b>	Array of objects	Composite	An array of objects that define the composite template (limited form).  See <a href="#">Table 157 on page 224</a> .
<b>composite-variable-input-file</b>	String	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.  The variable names are in the following format: <standard-template-name>.<atcreate-variable>  For example: CICS.startup=10

Table 156. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
<b>provisioning-version</b>	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Standard, Composite	Indicates if Get, Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> <li>• true if the operations are allowed</li> <li>• false if the operations are not allowed.</li> </ul>

**Note:** In Table 156 on page 221, the **Value Returned for Template Type** column shows the type of template for which a value is returned. For other template types, null is returned.

## State values

### archived

The entry is hidden from consumers. You can make it available again with the **Publish** action.

### corrupted

The contents of the software services template are missing or incorrect. Delete the template.

### draft

The entry is in the edit state and visible only to the owner and the administrator. No approvals are required for this template. The entry can be tested with the Test a software services template API.

### draft\_approved

The software services template is in edit state and all the approvals that are associated with the template and the respective runAs user IDs have been received. The entry can be tested with the Test a software services template API.

### draft\_pending\_approvals

The software services template is in edit state and one or more associated approvals has not been approved. The entry cannot be tested (with the Test a software services template API) in this state.

### draft\_missing\_required\_approver

One or more of the definition files contains a runAsUser element without a corresponding approver element. Either an approver element must be added for the runAsUser element, or a domain or general approver must be added for the software services template. The entry cannot be tested (with the Test a software services template API) in this state.

### draft\_rejected

The template is in a draft state and one or more approvers rejected an approval. The entry cannot be tested (with the Test a software services template API) in this state.

### missing\_required\_member

One or more of the members of a composite template that was referenced in the composite definition is not available. This state applies only to published or archived composite templates.

### pending\_security\_update

Permission to access the software services template is being processed. No API requests are allowed for the software services template until the security processing is complete.

### published

The entry is locked and visible to consumers.

### security\_update\_failed

Security access setup related to the software services template failed. Only the view and delete API requests are available.

Table 157. Contents of composite-definition object

Field	Type	Required/optional	Description
sequence	integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
missing	boolean	Required	<ul style="list-style-type: none"> <li>• true if no published template is available that is related to the original version used when the template was defined</li> <li>• false if a published template exists that satisfies the published template requirement</li> </ul>
description	String	Required	Description of the software services template.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
software-type	String	Required	Type of software that is being provisioned.

## Example HTTP interaction

Figure 85 on page 224 shows a request to retrieve a list of software services template.

```
GET https://pev243.pok.ibm.com/zosmf/provisioning/rest/1.0/scc
```

Figure 85. Sample request to list software services templates

The following is a response body.

```

{
  "scc-list": [
    {
      "name": "s2",
      "version": "2",
      "owner": "ibmuser",
      "state": "published",
      "description": "",
      "generated-name": "s2.2.default",
      "object-id": "93d9c9dc-6b47-4222-89a6-f18764b28463",
      "base-object-id": "f9211dbd-a7e8-44ca-a5b3-a4eed0a21f69",
      "domain-name": "default",
      "action-definition-file": "definition/a.xml",
      "software-id": "prodID",
      "software-name": "Product Name",
      "software-type": "MIX",
      "software-version": "Version 1",
      "workflow-definition-file": "definition/p.xml",
      "workflow-id": "wfID",
      "workflow-vendor": "IBM",
      "workflow-version": "1.0",
      "workflow-variable-input-file": "",
      "create-time": "2017-11-11T15:13:10.909Z",
      "created-by-user": "ibmuser",
      "last-modified-by-user": "ibmuser",
      "last-modified-time": "2017-11-11T15:13:18.062Z",
      "template-type": "standard",
      "composite-parents": [
        "S1_S2"
      ],
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    },
    {
      "name": "S1_S2",
      "version": "1",
      "owner": "ibmuser",
      "state": "missing_required_member",
      "description": "This is a composite template that brings up services s1 and s2. The s2 services references information from the s1 services to satisfy its run-time properties.",
      "generated-name": "S1_S2.1.default",
      "object-id": "5f746d4c-ad24-4355-99d3-b83466ce4492",
      "base-object-id": "f4feb4e9-f2e3-4121-a483-fa4bf10282b9",
      "domain-name": "default",
      "create-time": "2017-11-11T15:14:09.701Z",
      "created-by-user": "ibmuser",
      "last-modified-by-user": "ibmuser",
      "last-modified-time": "2017-11-11T15:22:17.595Z",
      "template-type": "composite",
      "composite-cluster": false,
      "composite-definition": [
        {
          "sequence": 1,
          "number-of-instances": 1,
          "missing": true,
          "description": "",
          "published-template-name": "s1",
          "software-type": ""
        },
        {
          "sequence": 2,
          "number-of-instances": 1,
          "missing": false,
          "description": "",
          "published-template-name": "s2",
          "software-type": "MIX"
        }
      ],
      "composite-variable-input-file": "",
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    }
  ],
}

```

```

{
  "name": "s1",
  "version": "2",
  "owner": "ibmuser",
  "state": "archived",
  "description": "",
  "generated-name": "s1.2.default",
  "object-id": "e214615b-ae4a-407c-8408-20b45b1a3472",
  "base-object-id": "08ead9fe-59f0-46c7-a1fb-8d5f9b39f08e",
  "domain-name": "default",
  "action-definition-file": "definition/a.xml",
  "software-id": "prodID",
  "software-name": "Product Name",
  "software-type": "MIX",
  "software-version": "Version 1",
  "workflow-definition-file": "definition/p.xml",
  "workflow-id": "wfID",
  "workflow-vendor": "IBM",
  "workflow-version": "1.0",
  "workflow-variable-input-file": "",
  "create-time": "2017-11-11T15:09:59.963Z",
  "created-by-user": "ibmuser",
  "last-modified-by-user": "ibmuser",
  "last-modified-time": "2017-11-11T15:27:42.184Z",
  "template-type": "standard",
  "composite-parents": [],
  "provisioning-version": "1400",
  "provisioning-version-supported": true
}
,
{
}

```

## Publish a software services template

You can use this operation to publish a software services template. The publish operation locks the template, preventing any further modification, and creates a public copy of it.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/publish
```

In this request

#### <object-id>

Identifies the software services template to be published.

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation publishes a software services template.

Publishing a new version of a standard template automatically archives any composite templates that include it. The user can either republish an archived affected composite template or create a new version of it. The composite template can be published only if all connector information is supplied at the time of the publish operation.

On successful completion, HTTP status code 200 Ok is returned, indicating that the request resulted in a software services template being published.

The software services template must be in the draft, draft approved, or archived state.

To work with a published software services template, use the REST APIs that are described in [“Published software service template services” on page 250](#).

## Request content

The request body is optional. It contains a JSON object that describes the publish operation. See [Table 158 on page 227](#).

Table 158. Request content for the software services template request			
Parameter	Type	Required or Optional	Description
<b>archive-existing</b>	boolean	Optional	<p>If set to true, indicates that if a published entry with this name already exists, that entry should be moved into the archived state, and publish this one instead.</p> <p>If set to false, indicates that if a published entry with this name already exists, the request should fail. False is the default if this parameter is not specified.</p> <p>If no published entry with this name already exists, then this flag is ignored.</p>
<b>ignore-test</b>	boolean	Optional	<p>If set to true, indicates a publish of the template does not require a test run to be performed.</p> <p>If set to false, a test run must be performed before a publish can be performed.</p> <p>If this parameter is not specified, then the value defaults to false.</p>
<b>ignore-source-change</b>	boolean	Optional	<p>If set to true, indicates that the publish of the entry is not restricted by the change in the original source that was used on the create or modify of the entry.</p> <p>If this parameter is not specified, then the value defaults to false.</p>

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 0k is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. [Table 159 on page 228](#) lists the fields in the JSON object.

Table 159. Response from the software services template request		
Field	Type	Description
<b>composites-affected</b>	Collection of objects	Collection of composite-affected objects that shows the composite templates that were affected by this action. It is returned only if the action was performed for a standard software services template.

Table 160. Composite-affected object		
Field	Type	Description
<b>name</b>	String	Name of the composite template that was affected by an update to a standard template.
<b>state</b>	String	State of the composite template. For example, the state changes from published to missing_required_member when a standard published template is no longer available to satisfy the member requirement.

## Example HTTP interaction

[Figure 86 on page 228](#) shows a request to publish a software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/e214615b-ae4a-407c-8408-20b45b1a3472/actions/publish
```

*Figure 86. Sample request to publish a software services template*

The response body is as follows.

```
{
  "composites-affected": [
    {
      "name": "S1_S2",
      "state": "archived"
    }
  ]
}
```

## Test a software services template

You can use this operation to test a software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/test
```

In this request

#### <object-id>

Identifies the software services template to be published.

### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

## Query parameters

None.

## Description

This operation lets you perform a test run of a software services template. The test run creates a workflow entry, starts the workflow entry, and creates a software services registry entry in being provisioned state.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the entry in the registry was created.

All approvals for a software services template must be approved before it can be tested. The software services template must be in the draft or draft approved state.

The software services template must be in the draft or draft\_approved state.

## Request content

The request content is expected to contain a JSON object that describes the test run. See [Request content for the software services template request](#) and [Table 162 on page 230](#).

Table 161. Request content for the test software services template request			
Field name	Type	Required or optional	Description
<b>input-variables</b>	Array of objects	Optional	Array of variable objects containing variables to be used for running the software services template.  For a composite template, the variable naming convention is as follows: <published-template-name>.<atCreate-variable-name>.  See <a href="#">Table 162 on page 230</a> .
<b>user-data-id</b>	String	Optional	ID of user-data. The user-data-id and user-data values are associated with the software services instance that is created and are returned with requests for the software services instance.
<b>user-data</b>	String	Optional	User-supplied data to be associated with the software services instance. Only allowed if user-data-id is also provided.
<b>tenant-name</b>	String	Optional	Required if the template is associated with more than one tenant
<b>account-info</b>	String	Optional	Account information to use in the JCL JOB statement. By default, it is the account information that is associated with the tenant resource pool
<b>systems-nicknames</b>	Array of Strings	Optional	Each string is the nickname of the system upon which to provision the software service defined by the template. The field is required if the resource pool associated with the tenant used for this operation is not set up to automatically select a system. Only one nickname is allowed. If the field is provided it is validated.

Table 161. Request content for the test software services template request (continued)

Field name	Type	Required or optional	Description
<b>expiration-period</b>	Integer	Optional	Number of days the instance will be kept provisioned after it is successfully provisioned. A value of 0 indicates that the instance does not expire. If not specified, this value defaults to the <code>rdp-instance-expiration-limit</code> value for the template.

Table 162. Runtime properties

Field	Type	Description
<b>name</b>	String	Name of the runtime property.
<b>value</b>	String	Value of the runtime property.
<b>sequence</b>	String	Provisioning sequence of the runtime property.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template request. [Table 161 on page 229](#) lists the fields in the JSON object.

Table 163. Response from a test software services template request

Field	Type	Description
<b>registry-info</b>	Object	Object mapping matching the return response body from a registry create. For a composite type template, this field reflects the parent registry instance response body.
<b>workflow-info</b>	Object	Object mapping matching the return response body from a workflow create. This field is not returned for a composite template.
<b>composite-children-registry-info</b>	Array of objects	For a composite type template, this field contains an array of composite child registry information objects. See <a href="#">Table 164 on page 231</a> .
<b>system-nickname</b>	String	Nickname of the system that the service is provisioned on.

Table 164. Composite child registry information objects		
Field	Type	Description
<b>sequence</b>	Integer	The order in which the child registry instances are being provisioned, starting with 1. The deprovisioning order is the reverse.
<b>object-name</b>	String	The name of the newly created object.
<b>object-id</b>	String	The ID of the newly created object. This object ID is to be used on further requests to the object.
<b>object-uri</b>	String	The URI of the newly created object
<b>external-name</b>	String	The external name of the newly created object.

## Example HTTP interactions

Figure 87 on page 231 shows a request to test a standard software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/6e39d092-745a-4e81-8b7b-d3e1327ad230/actions/test

{
  "tenant-name": "default",
  "input-variables": [
    { "name": "CMD", "value": "S BCTEST" },
    { "name": "INS", "value": "Instructions" },
    { "name": "WELSHIE", "value": "Truepenny Traveling a Trail of Hope" },
  ],
  "systems-nicknames": ["SY1"]
}
```

Figure 87. Sample request to test a standard software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "MIX_2",
    "object-id": "c35de2ea-3d6c-47ec-bc32-62b9013ffcd5",
    "object-uri": "/zosmf/provisioning/rest/1.0/scc/c35de2ea-3d6c-47ec-bc32-62b9013ffcd5",
    "external-name": "MIX_SCOUT01",
    "system-nickname": "SY1"
  },
  "workflow-info": {
    "workflowKey": "540ef4fa-754b-40dd-9951-0e80edd1ec3b",
    "workflowDescription": "Mix1 workflow",
    "workflowID": "wfID",
    "workflowVersion": "1.0",
    "vendor": "IBM"
  },
  "system-nickname": "SY1"
}
```

Figure 88. Sample response body

Figure 89 on page 231 shows a request to test a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/f98fb327-6714-420d-adc5-89793e7060d8/actions/test

{
  "tenant-name": "default",
  "input-variables": [
    { "name": "s1.CMD", "value": "S BCTEST" },
    { "name": "s1.WELSHIE", "value": "Tucker" }
  ],
  "systems-nicknames": ["SY1"]
}
```

Figure 89. Sample request to test a composite software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "SCOUT_3",
    "object-id": "6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "object-uri": "/zosmf/provisioning/rest/1.0/scr/6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "external-name": "SCOUT_SCOUT00",
    "system-nickname": "SY1"
  },
  "system-nickname": "SY1",
  "composite-children-registry-info": [
    {
      "sequence": 1,
      "object-name": "MIX_5",
      "object-id": "725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "external-name": "MIX_SCOUT01"
    },
    {
      "sequence": 2,
      "object-name": "MIX_6",
      "object-id": "b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "external-name": "MIX_SCOUT02"
    }
  ]
}
```

Figure 90. Sample response body

## Refresh a software services template

You can use this operation to refresh the files that are associated with a software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/refresh
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

#### <object-id>

Identifies the software services template that the approval is associated with.

#### <approval-object-id>

Identifies the approval object to delete.

### Query parameters

None.

### Description

This operation obtains the latest contents of the files that are associated with a software services template. For a standard template, this includes the workflow definition XML file, the actions XML file, the workflow variable input file if one is specified, and any documentation files that are provided. For a composite template, this includes the composite variable input file if one is specified, and any documentation files that are provided. The information in the software services template is updated to reflect the latest contents, including timestamps, of those files. The files are located by the original source paths.

Refresh causes all approvals to be reset.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

## Request content

None.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

## Example HTTP interaction

In [Figure 91 on page 233](#), a request is submitted to refresh a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/d0166782-4e18-4b07-a075-c8946c88e068/actions/refresh
```

*Figure 91. Sample request to refresh a software services template*

## Archive a software services template

You can use this operation to archive a published software services template.

## HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/archive
```

In this request

### **<object-id>**

Identifies the software services template to be archived.

### **<version>**

Is the URI path variable `<version>` that identifies the version of the z/OSMF software services template service. The following value is valid: `1.0`.

## Query parameters

None.

## Description

This operation lets you archive a software services template. This puts the software services template in an archived state.

The software services template must be in a published state.

On successful completion, HTTP status code 200 Ok is returned, indicating that the archive was successful.

## Request content

None.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord and a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:  
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 OK is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the request. [Table 137 on page 195](#) lists the fields in the JSON object.

Table 165. Response from the software services template request		
Field	Type	Description
<b>composites-affected</b>	Collection of objects	Collection of composite-affected objects that shows the composite templates that were affected by this action. It is returned only if the action was performed for a standard software services template.

Table 166. Composite-affected object		
Field	Type	Description
<b>name</b>	String	Name of the composite template that was affected by an update to a standard template.
<b>state</b>	String	State of the composite template. For example, the state changes from published to missing_required_member when a standard published template is no longer available to satisfy the member requirement.

## Example HTTP interaction

In [Figure 92 on page 234](#), a request is submitted to archive a software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/e214615b-ae4a-407c-8408-20b45b1a3472/actions/archive
```

*Figure 92. Sample request to archive a software services template*

The following is the response body for the request.

The response body is as follows.

```
{
  "composites-affected": [
    {
      "name": "S1_S2",
      "state": "missing_required_member"
    }
  ]
}
```

```
} ]
```

## Add an approval for a software services template

You can use this operation to create an approval record for a software services template. The approval record associates a user ID with the software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
```

In this request

#### <object-id>

Identifies the software services template that the approval is associated with.

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation lets you create a new general approval record for a software services template. It returns a unique approval object ID that identifies the approval. The approval is for all of the contents of the software services template and is not associated with a specific step or action.

The software services template must be in one of the draft states.

All approvals for a software services template must be approved before it can be published or tested. Once all the approvals are approved the state of the entry is updated to draft\_approved.

On successful completion, HTTP status code 201 (Normal) is returned, indicating that the approval was created.

### Request content

The request content is expected to contain a JSON object that describes the approval record. See [Request content for the software services template request](#).

Table 167. Request content for the add approval request			
Field name	Type	Required or optional	Description
user-id	String	Required	User ID associated with this approval.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord or a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Normal) is returned.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template request. [Table 167 on page 235](#) lists the fields in the JSON object.

*Table 168. Response from an add approval request*

Field	Type	Description
<b>object-id</b>	String	Object ID of the newly created approval. The object ID is to be used in subsequent requests to the session.
<b>object-uri</b>	String	URI of the newly created approval.

## Example HTTP interaction

In [Figure 93 on page 236](#), a request is submitted to add an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals
{
  "user-id": "nick"
}
```

*Figure 93. Sample request to add an approval record for a software services template*

The following is the response body for the request.

```
{
  "object-id": "eeb4f5a3-d883-4190-9961-412306707426",
  "object-uri": "/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/eeb4f5a3-d883-4190-9961-412306707426"
}
```

*Figure 94. Sample response body*

## Get an approval for a software services template

You can use this operation to retrieve an approval for a software services template.

## HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
    <approval-object-id>
```

In this request:

### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### <object-id>

Identifies the software services template that the approval is associated with..

### **<approval-object-id>**

Identifies the approval to retrieve.

## **Query parameters**

None.

## **Description**

This operation retrieves an approval for a software services template.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in an approval being retrieved. A response body is provided, as described in [“Response content” on page 237](#)

## **Request content**

None.

## **Authorization requirements**

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

## **HTTP status codes**

On successful completion, HTTP status code 200 (OK) is returned.

## **Response content**

On successful completion, the service returns a response body, which contains a JSON object with details about the approval. [Table 169 on page 237](#) lists the fields in the JSON object.

Table 169. Response from a get approval request		
Field	Type	Description
<b>status</b>	String	Status of the approval for this object: pending, approved, or rejected.
<b>comment</b>	String	Comment that is associated with the change in status from pending to either approved or rejected.
<b>description</b>	String	Additional detail that is provided if the approval is for a workflow definition that is associated with the action definition.
<b>approvers</b>	Array of strings	Each string in the array is a user ID or SAF group that can approve the template, workflow step, or action. Any one of the user IDs in the array can approve or reject. The last action takes precedence.
<b>status-update-by</b>	String	User ID that performed the last approve or reject action for this approval object.
<b>time-of-update</b>	String	The last time this object was updated, in ISO 8601 format.
<b>run-as-user</b>	String	The runAsUser user ID that the approval object is for. This applies only to action_definition and step_definition types.

Table 169. Response from a get approval request (continued)

Field	Type	Description
<b>type</b>	String	Type of approval object: general, domain, action_definition, or step_definition.
<b>object-id</b>	String	Unique object ID representing this approval object.
<b>workflow-file</b>	String	Workflow file definition that is associated with this runAsUser user ID.
<b>variable-input-file</b>	String	Specifies the variable input file that is associated with this runAsUser user ID.
<b>step-name</b>	String	Workflow file definition step that is associated with this runAsUser user ID.
<b>called-by-step-name</b>	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>called-by-workflow-file</b>	String	Workflow definition file that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>actions-file</b>	String	Actions definition file that is associated with this runAsUser user ID.
<b>action-name</b>	String	Action defined in the actions definition file that is associated with this runAsUser user ID.

## Example HTTP interaction

In Figure 95 on page 238, a request is submitted to get an approval record for a software services template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/
3f8ca645-f872-42b6-b0fc-3c6a9e470fcc/approvals/11aeb028-9a0a-45eb-a005-4a9460126c3a
```

Figure 95. Sample request to get an approval record for a software services template

The following is the response body for the request.

```
{
  "status": "missing_approver",
  "comment": null,
  "description": "The approver element originates from the 'Auto-Step' step in the /u/wfSuspend/p.xml
    which is the primary workflow definition file.",
  "type": "step_definition",
  "object-id": "11aeb028-9a0a-45eb-a005-4a9460126c3a",
  "user-ids": [],
  "status-update-by": null,
  "time-of-update": null,
  "run-as-user": "${instance-rau}",
  "workflow-file": "/u/wfSuspend/p.xml",
  "variable-input-file": null,
  "step-name": "Auto-Step",
  "called-by-step-name": null,
  "called-by-workflow-file": null,
  "actions-file": null,
  "action-name": null,
  "run-as-user-dynamic": true
}
```

Figure 96. Sample response body

## List the approvals for a software services template

You can use this operation to list all of the approvals for a software services template.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### <object-id>

Identifies the software services template that the approval is associated with.

### Query parameters

None.

### Description

This operation retrieves all of the approval for a software services template.

On successful completion, HTTP status code 200 (Normal) is returned. A response body is provided, as described in [“Response content” on page 239](#)

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord, domain administrator, domain approver, or template approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:  
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned.

### Response content

On successful completion, the service returns a JSON object. See [Table 170 on page 239](#) and [Table 171 on page 240](#).

Table 170. Response from a list approvals request		
Field	Type	Description
approvals	Array of objects	Array of Approval-Object containing information about the approvals associated with this software services template.

Table 171. Response from a get approval request

Field	Type	Description
<b>status</b>	String	Status of the approval for this object: pending, approved, or rejected.
<b>comment</b>	String	Comment that is associated with the change in status from pending to either approved or rejected.
<b>description</b>	String	Additional detail that is provided if the approval is for a workflow definition that is associated with the action definition.
<b>approvers</b>	Array of strings	Each string in the array is a user ID or SAF group that can approve the template, workflow step, or action. Any one of the user IDs in the array can approve or reject. The last action takes precedence.
<b>status-update-by</b>	String	User ID that performed the last approve or reject action for this approval object.
<b>time-of-update</b>	String	The last time this object was updated, in ISO 8601 format.
<b>run-as-user</b>	String	The runAsUser user ID that the approval object is for. This applies only to action_definition and step_definition types.
<b>type</b>	String	Type of approval object: general, domain, action_definition, or step_definition.
<b>object-id</b>	String	Unique object ID representing this approval object.
<b>workflow-file</b>	String	Workflow file definition that is associated with this runAsUser user ID.
<b>variable-input-file</b>	String	Specifies the variable input file that is associated with this runAsUser user ID.
<b>step-name</b>	String	Workflow file definition step that is associated with this runAsUser user ID.
<b>called-by-step-name</b>	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>called-by-workflow-file</b>	String	Workflow definition file that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>actions-file</b>	String	Actions definition file that is associated with this runAsUser user ID.
<b>action-name</b>	String	Action defined in the actions definition file that is associated with this runAsUser user ID.

## Example HTTP interaction

In Figure 97 on page 240, a request is submitted to get an approval record for a software services template.

```
GET https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/3f8ca645-f872-42b6-b0fc-3c6a9e470fcc/approvals
```

Figure 97. Sample request to list the approval records for a software services template

The following is the response body for the request.

```
{
  "approvals": [
    {
      "status": "missing_approver",
      "comment": null,
      "description": "The approver element originates from the 'Auto-Step' step in the /u/wfSuspend/p.xml
        which is the primary workflow definition file.",
      "type": "step_definition",
      "object-id": "11aeb028-9a0a-45eb-a005-4a9460126c3a",
      "approvers": [],
      "status-update-by": null,
      "time-of-update": null,
      "run-as-user": "${instance-rau}",
      "workflow-file": "/u/wfSuspend/p.xml",
      "variable-input-file": null,
      "step-name": "Auto-Step",
      "called-by-step-name": null,
      "called-by-workflow-file": null,
      "actions-file": null,
      "action-name": null,
      "run-as-user-dynamic": true
    },
    {
      "status": "pending",
      "comment": null,
      "description": "The approver element originates from the primary action file.",
      "type": "action_definition",
      "object-id": "263c87d7-4043-4fe4-895f-ccd3ad092966",
      "approvers": [
        "zosmfad",
        "ibmuser",
        "agrp1",
      ],
      "status-update-by": null,
      "time-of-update": null,
      "run-as-user": "zosmfad",
      "workflow-file": null,
      "variable-input-file": null,
      "step-name": null,
      "called-by-step-name": null,
      "called-by-workflow-file": null,
      "actions-file": "/u/wfSuspend/a.xml",
      "action-name": "command1",
      "run-as-user-dynamic": false
    }
  ]
}
```

Figure 98. Sample response body

## Approve an approval record for a software services template

You can use this operation to approve the contents of approval record for a software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/  
<approval-object-id>/actions/approve
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### **<object-id>**

Identifies the software services template that the approval is associated with.

#### **<approval-object-id>**

Identifies the approval object to approve.

### Query parameters

None.

### Description

This operation approves the contents of an approval record for the software services template.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

### Request content

The request content is expected to contain a JSON object that describes the approval. See [Table 172 on page 242](#).

Table 172. Request content for the software services template request			
Field name	Type	Required or optional	Description
comment	String	Optional	Text describing the approval.

### Authorization requirements

The user's z/OS user ID must be defined as a domain approver or template approver, or be one of the approvers in the approval object.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:  
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

## Example HTTP interaction

In [Figure 99](#) on [page 243](#), a request is submitted to approve an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/
dacea656-ffbe-48ce-a193-575161ff9d43/actions/approve
```

*Figure 99. Sample request to approve an approval record for a software services template*

## Batch approve approval records for a software services template

You can use this operation to batch approve or reject the contents of one or more approval records for a software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
/actions/update
```

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### **<object-id>**

Identifies the software services template that the approvals are associated with.

### Query parameters

None.

### Description

This operation performs batch approval of the contents of one or more approval records for the software services template.

On successful completion, HTTP status code 204 (No content) is returned.

The software services template must be in one of the draft states.

### Request content

The request content is expected to contain a JSON object that describes the approval. See [Table 173](#) on [page 243](#).

Table 173. Request content for the software services template request			
Field name	Type	Required or optional	Description
<b>approve</b>	Array of approval objects	Optional	Identify what is being approved. See <a href="#">Table 174</a> on <a href="#">page 244</a> .
<b>reject</b>	Array of approval objects	Optional	Identify what is being rejected. See <a href="#">Table 174</a> on <a href="#">page 244</a> .

Table 174. Approval objects

Field name	Type	Required or optional	Description
<b>approval-object-ids</b>	Array of Strings	Required	Strings that identify the approval objects on which to perform the specified approval or rejection.
<b>comment</b>	String	Optional	Text describing the approval.

## Authorization requirements

The user's z/OS user ID must be defined as a domain approver or template approver, or be one of the approvers in the approval object.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See “Authorization requirements” on page 48.

## HTTP status codes

On successful completion, HTTP status code 204 (No Content) is returned.

## Example HTTP interactions

In Figure 100 on page 244, a request is submitted to approve an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/9e00c325-9bc3-47bb-b106-c7c41772eea3/approvals/actions/update
{
  "approve": {
    "approval-object-ids": ["6b5e7b0d-34d0-4ca8-8b73-4c9f6d178ad2"],
    "comment": "This is a comment"
  }
}
```

Figure 100. Sample request to perform batch approval for a software services template

In Figure 101 on page 244, a request is submitted to reject approval records for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/9e00c325-9bc3-47bb-b106-c7c41772eea3/approvals/actions/update
{
  "reject": {
    "approval-object-ids": ["cc9ecc32-c32d-48d0-8d55-8349269f51ee",
                           "9e00c325-9bc3-47bb-b106-c7c41772eea3",
                           "aaeec169-6637-4e37-9f71-01838f1f1ce8"]
  }
}
```

Figure 101. Sample request to perform batch rejection of approvals for a software services template

In Figure 102 on page 245, a request is submitted to approve and reject approval records for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/9e00c325-9bc3-47bb-b106-c7c41772eea3/approvals/actions/update
{
  "approve": {
    "approval-object-ids": ["6b5e7b0d-34d0-4ca8-8b73-4c9f6d178ad2"],
    "comment": "This is a comment"
  },
  "reject": {
    "approval-object-ids": ["cc9ecc32-c32d-48d0-8d55-8349269f51ee",
                           "9e00c325-9bc3-47bb-b106-c7c41772eea3",
                           "aaeec169-6637-4e37-9f71-01838f1f1ce8"]
  }
}
```

Figure 102. Sample request to perform batch approval and rejection of approvals for a software services template

## Reject the use of a user ID with a software services template

You can use this operation to reject the use of your user ID with a software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
<approval-object-id>/actions/reject
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### <object-id>

Identifies the software services template that the approval is associated with.

#### <approval-object-id>

Identifies the approval object to reject.

### Query parameters

None.

### Description

This operation rejects an approval that is associated with a software services template. Rejecting the approval means that your user ID is not allowed to be used with the software services template.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

### Request content

The request content is expected to contain a JSON object that describes the rejection. See [Table 175 on page 245](#).

Table 175. Request content for the software services template request			
Field name	Type	Required or optional	Description
comment	String	Optional	Text describing the rejection.

## Authorization requirements

The user's z/OS user ID must be defined as a domain approver or template approver, or be one of the approvers in the approval object.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

See [“Authorization requirements”](#) on page 48.

## HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

## Example HTTP interaction

In [Figure 103 on page 246](#), a request is submitted to reject an approval record for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/
dacea656-ffbe-48ce-a193-575161ff9d43/actions/reject

{
  "comment":"disagree with this, rework required"
}
```

*Figure 103. Sample request to reject an approval record for a software services template*

## Delete an approval for a software services template

You can use this operation to delete an approval that is associated with a software services template.

### HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scc/<object-id>/approvals/
<approval-object-id>
```

In this request:

#### <version>

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### <object-id>

Identifies the software services template that the approval is associated with.

#### <approval-object-id>

Identifies the approval object to delete.

## Query parameters

None.

## Description

This operation deletes an approval that is associated with a software services template.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in one of the draft states.

## Request content

None.

## Authorization requirements

The user's z/OS user ID must be defined as a landlord or a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:  
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

## Example HTTP interaction

In [Figure 104 on page 247](#), a request is submitted to delete an approval record for a software services template.

```
DELETE https://localhost:4444/zosmf/provisioning/rest/1.0/scc/8abd70b5-ac74-4f4a-bc09-266bf7cf8270/approvals/  
dacea656-ffbe-48ce-a193-575161ff9d43
```

*Figure 104. Sample request to delete an approval record for a software services template*

## Set security complete for a software services template

You can use this operation to indicate that the required security setup has been completed for a software services template.

### HTTP method and URI path

---

```
POST /zosmf/provisioning/rest/<version>/scc/<object-id>/actions/  
security_complete
```

---

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### **<object-id>**

Identifies the software services template that the approval is associated with.

### Query parameters

None.

### Description

This operation indicates that the required security setup has been completed for a software services template. The template can move to the next state.

On successful completion, HTTP status code 204 (Successful) is returned.

The software services template must be in the pending security update or pending security failed state.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must be defined as a landlord or a domain administrator.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class:  
<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 204 (Successful) is returned.

### Example HTTP interaction

In [Figure 105 on page 248](#), a request is submitted to indicate that security setup is complete for a software services template.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/scc/d0166782-4e18-4b07-a075-c8946c88e068/  
actions/security_complete
```

*Figure 105. Sample request to indicate security is complete for a software services template*



## Published software service template services

The published software service template services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software services templates that are in the published state.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 45.

The published software services catalog contains a list of the software services templates that are in the published state.

Table 176 on page 250 lists the operations that the published software service template services provide.

Table 176. z/OSMF published software service template services: operations summary

Operation name	HTTP method and URI path
<a href="#">“Run a published software service template”</a> on page 252	POST /zosmf/provisioning/rest/<version>/psc/<name>/actions/run
<a href="#">“Get a published software service template”</a> on page 257	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>
<a href="#">“Get a published software service template history”</a> on page 269	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/history
<a href="#">“Get consumer documentation for a published software service template”</a> on page 271	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/documentation/consumer
<a href="#">“Get prompt variables for a published software service template”</a> on page 273	GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/prompt-variables
<a href="#">“List the published software service templates”</a> on page 278	GET /zosmf/provisioning/rest/<version>/psc/
<a href="#">“Modify a published software service template”</a> on page 282	POST /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>

### Authorization requirements

Use of the published software service template services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 2.

The specific requirements for each published software service template service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements” on page 48](#). For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

## Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 177. Response from a software services template request failure		
Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

## Error logging

Errors from the software services template services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 201 Created

The request succeeded and resulted in the creation of an object.

### HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

### HTTP 204 No content

The request succeeded, but no content is available to be returned.

### HTTP 400 Bad request

The request contained incorrect parameters.

### HTTP 403 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

### HTTP 404 Not found

The requested resource does not exist.

### HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

## Related information

The run operation for a published template creates a workflow, starts the workflow, and creates a corresponding software services instance in the software services registry. To work with a software services instance, use the REST APIs described in [“Software services instance services” on page 286](#).

## Run a published software service template

Use this operation to run a software services template that is in the published state.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/psc/<name>/actions/run
```

In this request:

#### <version>

Is the URI path variable <version> that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

#### <name>

Identifies the software services template to be run.

## Query parameters

None.

## Description

This operation creates a workflow, starts the workflow, and creates a corresponding software services instance in the software services registry, with a state of being-provisioned.

To work with a software services instance, use the REST APIs described in [“Software services instance services” on page 286](#).

## Request content

The request content is expected to contain a JSON object as described in [Table 178 on page 252](#) and [Table 179 on page 253](#).

Table 178. Request content for the run software services template request			
Field name	Type	Required or optional	Description
<b>input-variables</b>	Array of Objects	Optional	An array of required runtime property objects. See <a href="#">Table 179 on page 253</a>  For a composite template, the variable naming convention is as follows: <published-template-name>.<atCreate-variable-name>.
<b>domain-name</b>	String	Optional	Required if the user has consumer authorization to more than one domain with this template name.
<b>tenant-name</b>	String	Optional	Required if the user has consumer authorization to more than one tenant in the same domain that contains this template name.
<b>user-data-id</b>	String	Optional	ID for the user data specified with user-data. Passed into the software services registry.

Table 178. Request content for the run software services template request (continued)

Field name	Type	Required or optional	Description
<b>account-info</b>	String	Optional	Account information to use in the JCL JOB statement. The default is the account information that is associated with the resource pool for the tenant.
<b>user-data</b>	String	Optional	User data that is passed into the software services registry. Can be specified only if user-data-id is provided.
<b>systems-nicknames</b>	Array of Strings	Optional	Each string is the nickname of the system upon which to provision the software service defined by the template. The field is required if the resource pool associated with the tenant used for this operation is not set up to automatically select a system. Only one nickname is allowed. If the field is provided it is validated.
<b>expiration-period</b>	Integer	Optional	Number of days the instance will be kept provisioned after it is successfully provisioned. A value of 0 indicates that the instance does not expire. If not specified, this value defaults to the <code>rdp-instance-expiration-limit</code> value for the template.

Table 179. Runtime properties

Field	Type	Description
<b>name</b>	String	Name of the runtime property.
<b>value</b>	String	Value of the runtime property.
<b>sequence</b>	String	Provisioning sequence of the runtime property.

## Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 48](#).

## HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 253](#).

## Response content

On successful completion, the service returns a response body, which contains a JSON object, as described in [Table 180 on page 254](#).

Table 180. Response from a run software services template request

Field	Type	Description
<b>registry-info</b>	Object	Object mapping that matches the response body returned from a registry create action. For a composite type template, this field reflects the parent registry instance response body.
<b>workflow-info</b>	Object	Object mapping that matches the response body returned from a workflow create action.
<b>composite-children-registry-info</b>	Array of objects	For a composite type template, this field contains an array of composite child registry information objects. See <a href="#">Table 181 on page 254</a> .
<b>system-nickname</b>	String	Nickname of the system that the service is provisioned on.

Table 181. Composite child registry information objects

Field	Type	Description
<b>sequence</b>	Integer	The order in which the child registry instances are being provisioned, starting with 1. The deprovisioning order is the reverse.
<b>object-name</b>	String	The name of the newly created object.
<b>object-id</b>	String	The ID of the newly created object. This object ID is to be used on further requests to the object.
<b>object-uri</b>	String	The URI of the newly created object
<b>external-name</b>	String	The external name of the newly created object.

## Example HTTP interaction

In [Figure 106 on page 254](#), a request is submitted to run the software services template named bringUpDB2.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2/actions/run

{
  "input-variables": [{ "name": "CSQ_MQ_SSID", "value": "ZCT1" },
    { "name": "CSQ_CMD_PFX", "value": "ZCT1" }, { "name": "CSQ_ENVIRONMENT", "value": "TEST" } ],
  "domain-name": "default",
  "tenant-name": "tenant1"
}
```

Figure 106. Sample request to run a software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "QMgr_7",
    "object-id": "c5a8ecdd-db35-466b-aad9-cba0f33bb84b",
    "object-uri": "/zosmf/provisioning/rest/1.0/scr/c5a8ecdd-db35-466b-aad9-cba0f33bb84b"
  },
  "workflow-info": {
    "workflowKey": "ff96459f-27fa-490a-a3e4-4086649c12f3",
    "workflowDescription": "Procedure to provision a MQ for zOS Queue Manager",
    "workflowID": "ProvisionQueueManager",
    "workflowVersion": "1.0.1",
    "vendor": "IBM",
  },
  "system-nickname": "DUMBNODE"
}
```

Figure 107. Sample response body

Figure 108 on page 255 shows a request to test a composite software services template.

```
POST https://pev184.pok.ibm.com/zosmf/provisioning/rest/1.0/scc/f98fb327-6714-420d-adc5-89793e7060d8/actions/test

{
  "tenant-name": "default",
  "input-variables": [
    { "name": "s1.CMD", "value": "S BCTEST" },
    { "name": "s1.WELSHIE", "value": "Tucker" }
  ],
  "systems-nicknames": ["SY1"]
}
```

Figure 108. Sample request to test a composite software services template

The following is the response body for the request.

```
{
  "registry-info": {
    "object-name": "SCOUT_3",
    "object-id": "6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "object-uri": "/zosmf/provisioning/rest/1.0/scr/6d7fcc96-50a8-49c1-880f-578ad0245e77",
    "external-name": "SCOUT_SCOUT00",
    "system-nickname": "SY1"
  },
  "system-nickname": "SY1",
  "composite-children-registry-info": [
    {
      "sequence": 1,
      "object-name": "MIX_5",
      "object-id": "725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/725aa201-5ba3-414e-bef0-cfe04f8c7fd2",
      "external-name": "MIX_SCOUT01"
    },
    {
      "sequence": 2,
      "object-name": "MIX_6",
      "object-id": "b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "object-uri": "/zosmf/provisioning/rest/1.0/scr/b73967c6-cd32-43db-8c39-a51dc3d52c2c",
      "external-name": "MIX_SCOUT02"
    }
  ]
}
```

Figure 109. Sample response body



## Get a published software service template

Use this operation to retrieve a published software service template from the catalog.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>
```

---

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### **<existing-entry-name>**

Identifies the software services template to be retrieved.

### Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

#### **domain-name**

Optional, string, specifies the domain name.

If you specify no query parameters, then all templates are returned.

### Description

This operation retrieves a published software service template from the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content”](#) on page 257.

### Request content

None.

### Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

See [“Authorization requirements”](#) on page 48.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

### Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services template. See [Table 183 on page 262](#), [Table 184 on page 263](#), [Table 185 on page 264](#), and [Table 186 on page 264](#).

Table 182. Response from a get software services template request

Field	Type	Valid for Template Type	Description
<b>template-type</b>	String	Standard	Identifies the type of template: <b>standard</b> Defines a single software service. <b>composite</b> Consists of multiple published templates that are provisioned together.
<b>composite-definition</b>	Array of objects	Composite	An array of objects that define the composite template. See <a href="#">Table 188 on page 265</a> .
<b>composite-type</b>	String	Standard	Type of composite template (software service).
<b>composite-parents</b>	Array of strings	Standard	An array of strings. Each string is a composite template that includes this standard template. For example:  [c0e4d08f-f046-4a79-8a15-6981743d07ed, c0e4d08f-f046-4a79-8a15-6981743d07e3, c0e4d08f-f046-4a79-8a15-6981743d07ed]
<b>composite-cluster</b>	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
<b>base-object-id</b>	String	Standard	The object ID that is associated with all of the versions of the software services template.
<b>generated-name</b>	String	Standard, Composite	Generated name for the software services template.
<b>object-id</b>	String	Standard, Composite	Identifier for the software services template.
<b>name</b>	String	Standard, Composite	The name associated with the software services template.
<b>version</b>	String	Standard	Version of the software services template.
<b>owner</b>	String	Standard, Composite	User ID of the software services template owner.
<b>state</b>	String	Standard, Composite	Indicates the status of the software services template. It is always published. The entry is locked and visible to consumers.
<b>description</b>	String	Standard, Composite	Description of the software services template.
<b>tenants</b>	Array of Strings	Standard, Composite	Each string represents a tenant that the template is associated with.
<b>domain-shared-tenants</b>	Array of Strings	Standard, Composite	Each string represents a tenant in the domain that the template is associated with through the domain shared resource pool.

Table 182. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
<b>domain-name</b>	String	Standard, Composite	The domain the template is associated with.
<b>approvals</b>	Array of objects	Standard, Composite	Array of Approval-Object containing information about the approvals associated with this software services template. See <a href="#">Table 184 on page 263</a> .
<b>action-definition-file</b>	String	Standard	Location of the action definition file.
<b>action-definition-file-original-source</b>	String	Standard	Original user specified location of the action definition file
<b>action-definition-file-original-timestamp</b>	String	Standard	Last-modified time stamp for when the original action definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>actions</b>	Array of objects	Standard	Array of Action-Object containing information about the actions associated with the template. See <a href="#">Table 185 on page 264</a> .
<b>software-id</b>	String	Standard	A short, arbitrary, value that identifies the software that is being provisioned.
<b>software-name</b>	String	Standard	Name of the software that is being provisioned.
<b>software-type</b>	String	Standard	Type of software that is being provisioned.
<b>software-version</b>	String	Standard	Version of the software that is being provisioned.
<b>workflow-definition-file</b>	String	Standard	Location of the workflow definition file, the primary XML file that defines the workflow
<b>workflow-definition-file-original-source</b>	String	Standard	Original user-specified location of the workflow definition file.
<b>workflow-definition-file-original-timestamp</b>	String	Standard	The last-modified time stamp for when the original workflow definition file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>workflow-id</b>	String	Standard	A short, arbitrary value that identifies the workflow.
<b>workflow-vendor</b>	String	Standard	Name of the vendor that provided the workflow definition file.
<b>workflow-version</b>	String	Standard	Version of the workflow definition file.

Table 182. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
<b>composite-variable-input-file</b>	String	Composite	Location of the user-specified properties file that supplies values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.  The variable names are in the following format: <standard-template-name>.<atcreate-variable>  For example: CICS.startup=10
<b>composite-variable-input-file-original-source</b>	String	Composite	Location of the composite variable input file, an optional properties file used to specify in advance the values for one or more of the atCreate variables.
<b>composite-variable-input-file-original-timestamp</b>	String	Composite	The last-modified time stamp for when the original composite variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>workflow-variable-input-file</b>	String	Standard	Location of the workflow variable input file, an optional properties file used to specify in advance the values for one or more of the variables that are defined in the workflow definition file.
<b>workflow-variable-input-file-original-source</b>	String	Standard	The original user-specified location of the workflow variable input file.
<b>workflow-variable-input-file-original-timestamp</b>	String	Standard	The last-modified time stamp for when the original variable input file source was specified, in ISO 8601 format. Not available if the location of the file is a data set.
<b>workflow-clean-after-provisioned</b>	Boolean	Standard	This field is ignored. The workflows-disposition field should be referenced instead. The default is false. If the workflows-disposition field is not provided, its default value of archive is used.
<b>prompt-variables</b>	Array of objects	Standard	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template. See <a href="#">Table 183 on page 262</a> .
<b>public-variables</b>	Array of strings	Standard	Array of strings that name the public variables for the template.

Table 182. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
<b>at-create-variables</b>	Array of strings	Standard	<p>Array of strings that name the variables that are either prompt variables (variables that are expected to be prompted for in preparation for running the software services template), or required variables (variables for which a value is required when the software services template is run), or both.</p> <p>For a composite type template, any atCreate variable that was designated as a connector variable is excluded from the list. The variables are prefixed by the standard template name, for example MQ.BRING_UP.</p>
<b>consumer-documentation-file</b>	String	Standard, Composite	Location of the original file that provides information for consumers about the template.
<b>consumer-documentation-type</b>	String	Standard, Composite	Type of the consumer documentation file, either text or pdf. This is required if consumer-documentation-file is specified.
<b>admin-documentation-file</b>	String	Standard, Composite	Location of a file that provides information for administrators about the template.
<b>admin-documentation-type</b>	String	Standard, Composite	Type of the administrator documentation file, either text or pdf. This is required if admin-documentation-file is specified.
<b>create-time</b>	String	Standard, Composite	Time that this object was created, in ISO 8601 format.
<b>create-by-user</b>	String	Standard, Composite	User who created this object.
<b>last-modified-time</b>	String	Standard, Composite	The last time this object was updated, in ISO 8601 format.
<b>last-modified-by-user</b>	String	Standard, Composite	User who last updated this object.
<b>workflows-disposition</b>	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete.
<b>jobs-disposition</b>	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete.
<b>instances-disposition</b>	String	Standard, Composite	Disposition of instances of the template after the instances are deprovisioned: keep or delete.
<b>automatic-security</b>	String	Standard	<p>Indicates if the domain is setup to automatically create, update, or delete SAF profiles that are required for successful SAF authorization:</p> <ul style="list-style-type: none"> <li>• true if the domain that the template is associated with is set up for automatic authorization</li> <li>• false if the domain that the template is associated with is set up for manual authorization.</li> </ul>

Table 182. Response from a get software services template request (continued)

Field	Type	Valid for Template Type	Description
<b>SAF-resources</b>	Array of objects	Standard	Array of SAF-resource objects containing information about SAF resources used to authorize access. See <a href="#">Table 187 on page 265</a> .
<b>runAsUsers</b>	Array of objects	Standard	Array of RunAsUser objects containing information about runAsUser IDs that are referenced by this template. See <a href="#">Table 190 on page 267</a> .
<b>provisioning-version</b>	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.

Table 183. Response from a get request: Prompt-Variable Object

Field	Type	Description
<b>name</b>	String	Name of the property.
<b>value</b>	String	Current value for the property.
<b>required</b>	boolean	Indicates whether the variable value is required during the workflow create process.
<b>label</b>	String	Short label for the UI widget.
<b>description</b>	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
<b>abstract</b>	String	Brief description of the variable for the UI widget.
<b>type</b>	String	Type of the variable element: boolean, string, integer, decimal, time, date.
<b>must-be-choice</b>	boolean	Indicates whether the value must come from the provided choices.
<b>choices</b>	Array of Strings	Contains allowable choices for the value of the variable.
<b>regex</b>	String	Standard regular expression that constrains the variable value.
<b>multi-line</b>	boolean	Indicates whether the value requires a multi-line text box.
<b>min</b>	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
<b>max</b>	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
<b>places</b>	String	Maximum number of decimal places that can be specified for a variable of type decimal.
<b>error-message</b>	String	Default error message associated with an incorrect value.

Fields of type String default to null.

<i>Table 184. Response from a get request: Approval-Object</i>		
Field	Type	Description
<b>status</b>	String	Status of the approval for this object: pending, approved, or rejected.
<b>comment</b>	String	Comment associated with the change in status from pending to either approved or rejected.
<b>description</b>	String	Additional detail provided if the approval is for a workflow definition that's associated with the action definition. For example: This workflow definition is associated with the <action-name> action.
<b>user-ids</b>	Array of strings	Each string in the array is a user ID. Any one of the user IDs in the array can approve or reject the item. The action of the last ID takes precedence.
<b>status-update-by</b>	String	The user ID that performed the last approve or reject action for this approval object
<b>time-of-update</b>	String	The last time this object was updated, in ISO 8601 format.
<b>run-as-user</b>	String	The runAsUser user ID that the approval object is for. This applies only when the type is action_definition or step_definition.
<b>type</b>	String	Type of approval: general (for the template), domain, action_definition, step_definition
<b>object-id</b>	String	Unique object id representing this approval object.
<b>workflow-file</b>	String	Workflow file definition associated with this runAsUser user ID. Null if the user ID is not associated with a workflow definition step or is a general approval.
<b>variable-input-file</b>	String	Variable input file associated with this runAsUser user ID. Null if the user ID is not associated with a workflow definition step or is a general approval.
<b>step-name</b>	String	Workflow file definition step associated with this runAsUser user ID. Null if the user ID is not associated with a workflow definition step or is a general approval.
<b>called-by-step-name</b>	String	Step in the parent workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.
<b>called-by-workflow-file</b>	String	Workflow definition that called the workflow definition file that generated the approval object. Used if the definition file that generated the approval object is a callable workflow.

Table 184. Response from a get request: Approval-Object (continued)

Field	Type	Description
<b>actions-file</b>	String	Actions file definition associated with this runAsUser user ID. Null if the user ID is not associated with an action or is a general approval.
<b>action-name</b>	String	Action defined in the actions file associated with this runAsUser user ID. Null if the user ID is not associated with an action or is a general approval.
<b>run-as-user-dynamic</b>	boolean	Indicates if the run-as-user ID value can change: <b>true</b> The run-as-user ID value is not final and can change during the processing of the workflow <b>false</b> The run-as-user ID is final and cannot change during the processing of the workflow.

Table 185. Response from a get request: Action-Object

Field	Type
<b>name</b>	String
<b>type</b>	String
<b>command</b>	String
<b>workflow-definition-file</b>	String
<b>workflow-variable-input-file</b>	String
<b>workflow-variables</b>	Variable[]
<b>instructions</b>	String
<b>prompt-variables</b>	The prompt variable objects that are associated with the action.
<b>command-run-as-user-dynamic</b>	boolean. Indicates if the command-run-as-user ID value can change: <b>true</b> The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content. <b>false</b> The command-run-as-user ID is final and cannot change during the processing of the command.

Table 186. Response from a get request: Variable-Object

Field	Type
<b>name</b>	String
<b>value</b>	String
<b>visibility: public or private</b>	String

Table 187. Response from a create request: SAF-resource object

Field	Type	Description
<b>description</b>	String	Description of the resource.
<b>ids</b>	Array of Strings	Each string represents a User ID that is expected to validate against this SAF resource.
<b>groups</b>	Array of Strings	Each string represents a group ID that is expected to validate against this SAF resource.
<b>role</b>	String	The role of the IDs and/or groups that this SAF resource validation is for, that is, domain administrator, approver, consumer.
<b>resource-class</b>	String	The class associated with SAF resource.
<b>resource-name</b>	String	The name associated with the SAF resource.
<b>required-access</b>	String	The access required for the IDs and/or groups to be authorized successfully.
<b>other-required-ids</b>	Array of Strings	These ids are not referenced by the entity that returns this SAF resource object but must maintain successful validation against this SAF resource. These ids are used by other versions of this entity and all versions of the entity validate against the same SAF resource.
<b>audit-requirements</b>	String	Audit requirement that is associated with validation requests of IDs and groups against this SAF resource, for example, All successful validations must be logged.

Table 188. Response from a get request: composite-definition object

Field	Type	Description
<b>sequence</b>	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
<b>number-of-instances</b>	Integer	Indicates the number of child instances to be created using the template in a composite cluster.
<b>published-template-name</b>	String	The name of an existing published template in the domain that is associated with the composite template.
<b>connectors</b>	Array of Objects	<p>An array of connector object.</p> <p>The connector variables specified here take precedence over the variables field in this object and any variables specified in the workflow variable input file associated with the published template.</p> <p>See <a href="#">Table 189</a> on page 266.</p>

Table 188. Response from a get request: composite-definition object (continued)

Field	Type	Description
<b>prompt-variables</b>	Array of objects	<p>Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the software services template.</p> <p>If specified, this overrides the array of prompt variables that are associated with the template specified with published-template-name. Only prompt variables that are already specified for the published-template-name can be specified. An empty array will translate into not prompting for any variables. If this field is not provided or set to null, then the prompt variables that are associated with published-template-name are used.</p> <p>If the connector variable-name is also a prompt-variable, then the connector takes precedence, and the variable is not promptable.</p>

Table 189. Connector object

Field	Type	Required/optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.
not-valid	boolean	Required	<p>Indicates if the information (variable-name, source-template, and source-variable-name values) in this connector is valid. The value is:</p> <ul style="list-style-type: none"> <li>• false, if all of the information is accurate</li> <li>• true, if one or more of the values are incorrect.</li> </ul>

Table 190. Response from a get request: RunAsUser object

Field	Type	Description
<b>description</b>	String	Additional detail provided if the run-as-user is for a workflow definition that is associated with the action definition. Example: This workflow definition is associated with the <action-name> action.
<b>approver-user-ids</b>	Array of Strings	Array of strings where each string is a user ID that originates from the approver element that is associated with the runAsUser for the template step or action.
<b>run-as-user</b>	String	The runAsUser user ID that the approval object is for. This is applicable only to action_definition and step_definition type.
<b>type</b>	String	One of the following: action_definition or step_definition
<b>workflow-file</b>	String	The workflow file definition that is associated with this runAsUser user ID.
<b>variable-input-file</b>	String	The variable input file that is associated with this runAsUser user ID.
<b>step-name</b>	String	The workflow file definition step that is associated with this runAsUser user ID.
<b>called-by-step-name</b>	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the step in the parent workflow definition that called the workflow definition file that generated the approval object.
<b>called-by-workflow-file</b>	String	Used if the definition file that generated the approval object is a callable workflow. Identifies the workflow definition that called the workflow definition file that generated the approval object.
<b>actions-file</b>	String	The actions file definition that is associated with this runAsUser user ID.
<b>action-name</b>	String	The action defined in the actions file that is associated with this runAsUser user ID.
<b>run-as-user-dynamic</b>	boolean	Indicates if the run-as-user ID value can change:  <b>true</b> The run-as-user ID value is not final and can change during the processing of the workflow  <b>false</b> The run-as-user ID is final and cannot change during the processing of the workflow.

## Example HTTP interaction

In Figure 110 on page 267, a request is submitted to retrieve a standard software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2
```

Figure 110. Sample request to get a standard software services template

The following is the response body for the example GET request.

```
{
  "name": "mqCBA",
  "version": "1",
  "owner": "domadmin",
  "state": "published",
  "description": "This workflow provisions an MQ for z/OS Queue Manager",
  "tenants": [...],
  "actions": [...],
  "approvals": [],
  "tested": false,
  "generated-name": "mqCBA.1.default",
  "domain-name": "default",
  "action-definition-file": "definition/qmgrActions.xml",
  "action-definition-file-original-source": "/users/gg/mqCBA/definition/qmgrActions.xml",
  "action-definition-file-original-timestamp": "2016-11-18T20:00:42Z",
  "software-id": "5655-W97",
  "software-name": "IBM MQ for z/OS",
  "software-type": "QMGr",
  "software-version": "V8.0.0",
  "workflow-definition-file": "definition/provision.xml",
  "workflow-definition-file-original-source": "/users/gg/mqCBA/definition/provision.xml",
  "workflow-definition-file-original-timestamp": "2016-11-18T20:03:47Z",
  "workflow-id": "ProvisionQueueManager",
  "workflow-vendor": "IBM",
  "workflow-version": "1.0.1",
  "workflow-variable-input-file": "definition/workflow_variables.properties",
  "workflow-variable-input-file-original-source":
    "/users/gg/mqCBA/definition/workflow_variables.properties",
  "workflow-variable-input-file-original-timestamp": "2016-11-18T20:00:42Z",
  "prompt-variables": [],
  "public-variables":
    ["CSQ_CHIN_SERVICE_CLASS_NAME", "CSQ_MSTR_SERVICE_CLASS_NAME", "CSQ_TCPIP_PORT_NUMBER",
     "CSQ_AUTO_GEN_CMD_PFX_SSID", "CSQ_CMD_PFX_FOR_AUTO_GEN", "CSQ_CHIN_REPORT_CLASS_NAME",
     "CSQ_MSTR_CLASSIFICATION_RULE_ID", "CSQ_MSTR_REPORT_CLASS_NAME", "CSQ_CMD_PFX", "CSQ_QSGDISP",
     "CSQ_CHIN_CLASSIFICATION_RULE_ID", "CSQ_TCPIP_STATUS_CODE", "CSQ_TARG_LIB_HLQ", "CSQ_SSID",
     "CSQ_TCPIP_PORT_ID", "CSQ_LANG_LETTER", "CSQ_ENVIRONMENT"]
  ],
  "at-create-variables": [],
  "workflow-clean-after-provisioned": true,
  "security-wf-info": null,
  "create-time": "2016-11-18T20:00:43.504Z",
  "created-by-user": "domadmin",
  "last-modified-by-user": "domadmin",
  "last-modified-time": "2016-11-18T20:04:50.913Z",
  "admin-documentation-file-original-source": "/users/gg/mqCBA/documentation/admin-
mqaas_readme.pdf",
  "admin-documentation":
    "/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
admin",
  "admin-documentation-type": "pdf",
  "consumer-documentation-file-original-source":
    "/users/gg/mqCBA/documentation/consumer-workflow_variables.properties",
  "consumer-documentation":
    "/zosmf/provisioning/rest/1.0/scc/5b0c3367-b856-4727-99ac-f9a79c9abf28/documentation/
consumer",
  "consumer-documentation-type": "text",
  "base-object-id": "c0e4d08f-f046-4a79-8a15-6981743d07ed",
  "admin-documentation-mime-type": "application/pdf",
  "consumer-documentation-mime-type": "text/plain",
  "SAF-resources": [],
  "runAsUsers": [],
  "runAsUser-audit": true,
  "automatic-security": true,
  "published-timestamp": "2017-04-05T16:16:55.878Z",
  "archived-timestamp": "",
  "provisioning-version": "1400"
}
```

## Get a published software service template history

Use this operation to retrieve the history for a published software service template.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/history
```

---

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service.  
The following value is valid: 1.0.

#### **<existing-entry-name>**

Identifies the published template for which history is to be retrieved.

### Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

#### **domain-name**

Optional, string, specifies the domain name.

### Description

This operation retrieves the history for a published software service template.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in history being retrieved. A response body is provided, as described in [“Response content” on page 269](#).

### Request content

None.

### Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

See [“Authorization requirements” on page 48](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

### Response content

On successful completion, the service returns a JSON response body. The response contains an array of history objects, each of which contains information about an action that is associated with the published software service template. [Table 191 on page 270](#) lists the fields in the history object.

Table 191. Response from a get request: History object

Field	Type	Valid for Template Type	Description
<b>action-type</b>	String	Standard, Composite	The type of action taken on the object. The following action-types are valid: <ul style="list-style-type: none"> <li>• Create</li> <li>• Add approval</li> <li>• Approve</li> <li>• Archive</li> <li>• Modify</li> <li>• Publish</li> <li>• Refresh</li> <li>• Reject</li> <li>• Remove approval</li> <li>• Run</li> <li>• Test run</li> <li>• Security complete</li> <li>• Update approval</li> </ul>
<b>user</b>	String	Standard, Composite	The user who performed the action.
<b>action-time</b>	String	Standard, Composite	The time that the action was taken.
<b>action-details</b>	String	Standard, Composite	A brief description of the action that was taken. This field is set in the code of the action that was taken. For example, on template approval, this field contains the approval comments.

## Example HTTP interaction

In Figure 111 on page 270, a request is submitted to retrieve the history for a published software service template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/template1/history
```

Figure 111. Sample request to retrieve a published software template history

The following figure shows the response body for the sample request in the previous example.

```
{
  "history": [
    {
      "action-type": "Create",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:15.791Z",
      "action-details": "Created template"
    },
    {
      "action-type": "Publish",
      "user": "ibmuser",
      "action-time": "2020-12-14T14:41:24.860Z",
      "action-details": "Published template"
    }
  ]
}
```

```
} ]
```

## Get consumer documentation for a published software service template

Use this operation to retrieve the consumer documentation for a published software service template from the catalog.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/documentation/  
consumer
```

---

In this request:

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

#### **<existing-entry-name>**

Identifies the software services template to be retrieved.

#### **documentation/consumer**

Causes the consumer documentation file to be retrieved.

### Query parameters

None.

### Description

This operation retrieves the consumer documentation for a published software service template from the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in the consumer documentation for a software services template being retrieved.

### Request content

None.

### Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

### Response content

None.

### Example HTTP interaction

In [Figure 112 on page 272](#), a request is submitted to retrieve consumer documentation for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2/documentation/consumer
```

*Figure 112. Sample request to get consumer documentation for a software services template*

## Get prompt variables for a published software service template

Use this operation to retrieve the name/value pairs for variables that are required to run the software services template and for which a prompt can be used to obtain the value.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>/  
prompt-variables
```

---

In this request:

#### **<existing-entry-name>**

Identifies the software services template for which variables are to be retrieved.

#### **<version>**

Is the URI path variable that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

You can specify the following query parameter on this request to limit the software services templates that are returned. To be returned, a software services template must all query parameters.

#### **domain-name**

Optional, specifies the domain name.

### Description

This operation retrieves the variables for which a prompt can obtain the value.

On successful completion, HTTP status code 200 (Normal) is returned, indicating that the request resulted in a software services template being retrieved. A response body is provided, as described in [“Response content” on page 273](#)

### Request content

None.

### Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

### Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the prompt variables. [Table 192 on page 274](#) lists the fields in the JSON object.

Table 192. Response from a get prompt variables request

Field	Type	Template Type	Description
<b>prompt-variables</b>	Array of objects	Standard	An array of required prompt variable objects. See <a href="#">Table 193 on page 274</a> .
<b>composite-prompt-variables</b>	Array of objects	Composite	Array of composite-prompt-variables objects that contains information about the variables that are expected to be prompted for in preparation for running the composite software services template. See <a href="#">Table 194 on page 275</a> .

Table 193. Response from a get request: Prompt-Variable Object

Field	Type	Description
<b>name</b>	String	Name of the property.
<b>value</b>	String	Current value for the property.
<b>required</b>	boolean	Indicates whether the variable value is required during the workflow create process.
<b>label</b>	String	Short label for the UI widget.
<b>description</b>	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
<b>abstract</b>	String	Brief description of the variable for the UI widget.
<b>type</b>	String	Type of the variable element: boolean, string, integer, decimal, time, date.
<b>must-be-choice</b>	boolean	Indicates whether the value must come from the provided choices.
<b>choices</b>	Array of Strings	Contains allowable choices for the value of the variable.
<b>regex</b>	String	Standard regular expression that constrains the variable value.
<b>multi-line</b>	boolean	Indicates whether the value requires a multi-line text box.
<b>min</b>	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
<b>max</b>	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
<b>places</b>	String	Maximum number of decimal places that can be specified for a variable of type decimal.
<b>error-message</b>	String	Default error message associated with an incorrect value.

Table 194. Response from a get request: Composite-Prompt-Variable Object

Field	Type	Description
<b>published-template-name</b>	String	The name of the published template in the composite template that the prompt-variables field is associated with.
<b>prompt-variables</b>	Array of objects	Array of prompt variable objects containing information about the variables that are expected to be prompted for in preparation for running the published-template-name software services template as part of the composite software services template. See <a href="#">Table 193</a> on page 274.

## Example HTTP interaction

Figure 113 on page 275 shows a request to retrieve the prompt variables for a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc/mq/prompt-variables
```

Figure 113. Sample request to get the prompt variables for a published software service template

The following is the response body for the request.

```

{
  "prompt-variables": [
    {
      "name": "CSQ_MQ_SSID",
      "label": "MQ_SSID",
      "description": "The name of the MQ subsystem to be provisioned.",
      "type": "string",
      "value": "ZCT1",
      "required": true,
      "choices": null,
      "regex": "[A-Z0-9]{1,4}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Subsystem identifier",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    },
    {
      "name": "CSQ_CMD_PFX",
      "label": "CMD_PFX",
      "description": "The MQ subsystem command prefix.",
      "type": "string",
      "value": "!ZCT1",
      "required": true,
      "choices": null,
      "regex": "[!\\sa-zA-Z0-9.,!()?)*+|=|;%_?:$@#&<>]{1,8}",
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Command prefix",
      "multi-line": false,
      "must-be-choice": false,
      "error-message": "The value entered is not valid."
    },
    {
      "name": "CSQ_ENVIRONMENT",
      "label": "ENVIRONMENT",
      "description": "The environment for which the queue manager is to be provisioned/de-provisioned.  
The BSDS, Log and Pageset datasets vary depending on the environment.",
      "type": "string",
      "value": "TEST",
      "required": true,
      "choices": [
        "DEV",
        "TEST",
        "QA",
        "PROD"
      ],
      "regex": null,
      "min": null,
      "max": null,
      "places": null,
      "abstract": "Environment for which the queue manager is to be provisioned/de-provisioned  
(DEV, TEST, QA, PROD)",
      "multi-line": false,
      "must-be-choice": true,
      "error-message": "The value entered is not valid."
    }
  ]
}

```

Figure 114 on page 276 shows a request to retrieve the prompt variables for a published composite template.

```
GET /zosmf/provisioning/rest/1.0/psc/scout/prompt-variables
```

*Figure 114. Sample request to retrieve prompt variables, composite template*

The following is the response body for the request.

```

{
  "composite-prompt-variables": [
    {
      "prompt-variables": [],
      "published-template-name": "s1"
    },
    {
      "prompt-variables": [
        {
          "name": "CMD",
          "label": "CMD",
          "description": "CMD",
          "type": "string",
          "value": "S BCTEST",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "CMD",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        },
        {
          "name": "WELSHIE",
          "label": "name",
          "description": "This variable contains the name of a welsh springer
spaniel.",
          "type": "string",
          "value": "Scout",
          "required": false,
          "choices": null,
          "regex": ".*",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "Name of a Welsh Springer Spaniel",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": ".*"
        },
        {
          "name": "INS",
          "label": "INS",
          "description": "INS",
          "type": "string",
          "value": "Instructions",
          "required": false,
          "choices": null,
          "regex": ".{1,1000000}",
          "min": null,
          "max": null,
          "places": null,
          "abstract": "INS",
          "multi-line": false,
          "must-be-choice": false,
          "error-message": "The value entered is not valid."
        }
      ],
      "published-template-name": "s2"
    }
  ]
}

```

Figure 115. Response body for the GET prompt variables request, composite template

## List the published software service templates

Use this operation to list the software services templates in the catalog that are in the published state.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/psc/
```

---

In this request, the URI path variable *<version>* identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### Query parameters

You can specify the following query parameter on this request to limit the software services templates that are returned. To be returned, a software services template must all query parameters.

**name**

Optional, regular expression, specifies the external name of the software services template.

**owner**

Optional, specifies the user ID or group ID that identifies the owner of the software services template.

**software-type**

Optional, specifies the type of software being provisioned.

**domain-name**

Optional, specifies the domain name.

**template-type**

Optional, specifies the type (standard or composite).

### Description

This operation retrieves software services templates that are in the published state from the catalog.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in software services templates being retrieved. A response body is provided, as described in [“Response content” on page 278](#).

### Request content

None.

### Authorization requirements

The user ID must be in a tenant that the template is associated with, or be an approver.

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

### Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the software services templates. See [Table 195 on page 279](#) and [Table 196 on page 279](#).

Table 195. Array of objects

Field	Type	Description
<b>pssc-list</b>	Array of objects	Array of software services template objects. The array is filtered based on any query parameters that were provided.

Table 196. Fields for each software services template

Field	Type	Value Returned for Template Type	Description
<b>generated-name</b>	String	Standard, Composite	The generated name for the software services template.
<b>object-id</b>	String	Standard, Composite	The ID that identifies the software services template.
<b>base-object-id</b>	String	Standard	The object ID that is associated with all the versions of the template. software services template.
<b>name</b>	String	Standard, Composite	Descriptive name for the software services template. The name must be unique. It can be up to 100 characters. The name cannot contain the symbols for less than (<), greater than (>), or ampersand (&).
<b>version</b>	String	Standard	Version of the software services template.
<b>owner</b>	String	Standard, Composite	User ID of the software services template owner.
<b>state</b>	String	Standard, Composite	Indicates the current status of the software services template: <b>published</b> The entry is locked and visible in the marketplace.
<b>description</b>	String	Standard, Composite	Description of the software services template.
<b>domain-name</b>	String	Standard, Composite	Name of the domain this template resides in.
<b>action-definition-file</b>	String	Standard	Location of the action definition file.
<b>software-id</b>	String	Standard	A short, arbitrary, value that identifies the software being provisioned.
<b>software-name</b>	String	Standard	Name of the software being provisioned.
<b>software-type</b>	String	Standard	Identifies the type of software being provisioned.
<b>software-version</b>	String	Standard	Version of the software being provisioned.

Table 196. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
<b>workflow-definition-file</b>	String	Standard	Location of the workflow definition file for the software services template. This file is the primary XML file for the workflow definition.
<b>workflow-id</b>	String	Standard	Identifies the workflow.
<b>workflow-vendor</b>	String	Standard	Name of the vendor that provided the workflow definition file.
<b>workflow-version</b>	String	Standard	Version of the workflow definition file.
<b>workflow-variable-input-file</b>	String	Standard	Optional properties file that specifies values for one or more of the variables that are defined in the workflow definition file.
<b>create-time</b>	String	Standard, Composite	The time that this software services template was created, in ISO 8601 format.
<b>create-by-user</b>	String	Standard, Composite	The user that created this software services template.
<b>last-modified-time</b>	String	Standard, Composite	The last time this software services template was updated, in ISO 8601 format.
<b>last-modified-by-user</b>	String	Standard, Composite	The user that last updated this software services template.
<b>template-type</b>	String	Standard, Composite	Identifies the type of template: <b>standard</b> Defines a single software service. <b>composite</b> Consists of multiple published templates that are provisioned together.
<b>composite-definition</b>	Array of objects	Composite	An array of objects that define the composite template. Not valid if template-type is standard. See <a href="#">Table 197 on page 281</a> .
<b>composite-variable-input-file</b>	String	Composite	Location of the properties file that you can use to specify in advance values for one or more of the atCreate variables that are defined in the member standard template workflow definition files.  The variable names are in the following format: <standard-template-name>.<atcreate-variable>  For example: CICS.startup=10

Table 196. Fields for each software services template (continued)

Field	Type	Value Returned for Template Type	Description
<b>composite-parents</b>	Array of strings	Standard	An array of strings. Each string is a composite template that includes this standard template. For example:  [c0e4d08f-f046-4a79-8a15-6981743d07ed, c0e4d08f-f046-4a79-8a15-6981743d07e3, c0e4d08f-f046-4a79-8a15-6981743d07ed]
<b>composite-cluster</b>	boolean	Optional	Indicates if child instances are created in a composite cluster. The value is true if child instances are created in a composite cluster, and false if child instances are not created in a composite cluster.
<b>provisioning-version</b>	String	Standard, Composite	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Standard, Composite	Indicates if Get, Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> <li>• true if the operations are allowed</li> <li>• false if the operations are not allowed.</li> </ul>

Table 197. Contents of composite-definition object

Field	Type	Required/optional	Description
sequence	integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
number-of-instances	Integer	Required	Indicates the number of child instances to be created using the template in a composite cluster.
missing	boolean	Required	<ul style="list-style-type: none"> <li>• true if no published template is available that is related to the original version used when the template was defined</li> <li>• false if a published template exists that satisfies the published template requirement</li> </ul>
description	String	Required	Description of the software services template.
published-template-name	String	Required	The name of an existing published template in the domain that is associated with the composite template.
software-type	String	Required	Type of software that is being provisioned.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 198. Response from a software services template request failure

Field	Type	Description
http-status	String	HTTP status code.
request-method	String	HTTP request method.
request-uri	String	HTTP request URI.
reason	String	HTTP status reason code.
message	String	Message describing the error.
detailed-message	String	Message describing the error in more detail.
debug	String	Debug information about for the error.

## Example HTTP interaction

Figure 116 on page 282 shows a request to retrieve a software services template.

```
GET https://localhost:4444/zosmf/provisioning/rest/1.0/psc HTTP/1.1
```

Figure 116. Sample request to list all published software service templates

The following is the response body for the request.

```
{
  "psc-list": [
    {
      "name": "mqCBA",
      "version": "1",
      "owner": "domadmin",
      "state": "published",
      "description": "This workflow provisions an MQ for z/OS Queue Manager",
      "generated-name": "mqCBA.1.default",
      "object-id": "5b0c3367-b856-4727-99ac-f9a79c9abf28",
      "base-object-id": "c0e4d08f-f046-4a79-8a15-6981743d07ed",
      "domain-name": "default",
      "action-definition-file": "definition/qmgrActions.xml",
      "software-id": "5655-W97",
      "software-name": "IBM MQ for z/OS",
      "software-type": "QMGR",
      "software-version": "V8.0.0",
      "workflow-definition-file": "definition/provision.xml",
      "workflow-id": "ProvisionQueueManager",
      "workflow-vendor": "IBM",
      "workflow-version": "1.0.1",
      "workflow-variable-input-file": "definition/workflow_variables.properties",
      "create-time": "2016-11-18T20:00:43.504Z",
      "created-by-user": "domadmin",
      "last-modified-by-user": "domadmin",
      "last-modified-time": "2016-11-18T20:28:43.951Z",
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    }
  ]
}
```

Figure 117. Response body for the GET request

## Modify a published software service template

Use this operation to modify fields in a published software services template.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/psc/<existing-entry-name>
```

In this request:

### **<version>**

Is the URI path variable `<version>` that identifies the version of the z/OSMF software services template service. The following value is valid: 1.0.

### **<name>**

Identifies the published software service template to be modified.

## **Query parameters**

None.

## **Description**

This operation modifies fields in a published software service template, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 283.

**Note:** Any existing instances that are already created from the template are not affected if this service is used to modify the jobs disposition or workflow disposition.

## **Request content**

The request content is expected to contain a JSON object as described in [Table 199](#) on page 283.

Table 199. Request content for the modify software services template request			
Field name	Type	Valid for template type	Description
<b>description</b>	String	Standard, Composite	Description of the software services template (up to 500 characters).
<b>workflows-disposition</b>	String	Standard	Disposition of provisioning and action workflows after they complete successfully: archive, keep, or delete. The default is archive.
<b>jobs-disposition</b>	String	Standard	Disposition of jobs from the provisioning and action workflows after they complete: keep or delete. The default is keep.
<b>instances-disposition</b>	String	Standard, Composite	Disposition of instances after they deprovision successfully: keep or delete. The default is keep.

## **Authorization requirements**

The user ID must be in a tenant that the template is associated with, or be an approver.

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements”](#) on page 48.

## **HTTP status codes**

For the valid HTTP status codes, see [“HTTP status codes”](#) on page 251.

## **Response content**

On successful completion, HTTP status code 204 (Normal) is returned, indicating that the request resulted in a modified published software service template.

## Example HTTP interaction

In Figure 118 on page 284, a request is submitted to modify a published software service template named bringUpDB2.

```
POST https://localhost:4444/zosmf/provisioning/rest/1.0/psc/bringUpDB2
{
  "description": "New description text here",
  "jobs-disposition": "delete",
  "workflows-disposition": "archive"
  "instances-disposition": "keep"
}
```

*Figure 118. Sample request to run a software services template*



## Software services instance services

The software services instance services are application programming interfaces (APIs), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software services instances in the software services registry.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 45.

The software services registry contains a list of the software on z/OS that has been registered as being provisioned, typically through the use of software services templates, although provisioning can be done manually. It is maintained in the z/OSMF data repository and has a sysplex-wide scope.

Table 200 on page 286 lists the operations that the software services instance services provide.

Table 200. z/OSMF software services instance services: operations summary	
Operation name	HTTP method and URI path
<a href="#">“Create a software services instance” on page 288</a>	POST /zosmf/provisioning/rest/<version>/scr
<a href="#">“Get the contents of a software services instance” on page 301</a>	GET /zosmf/provisioning/rest/<version>/scr/<object-id>
<a href="#">“Get the variables for a software services instance” on page 320</a>	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/variables
<a href="#">“Get key-value variables for a software services instance” on page 323</a>	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/key-value-variables
<a href="#">“List the software services instances” on page 312</a>	GET /zosmf/provisioning/rest/<version>/scr
<a href="#">“Update a software services instance” on page 325</a>	PUT /zosmf/provisioning/rest/<version>/scr/<object-id>
<a href="#">“Delete a software services instance” on page 333</a>	DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>
<a href="#">“Perform an action against a software services instance” on page 337</a>	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action>
<a href="#">“Resume a provisioning workflow” on page 335</a>	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/resume-workflow
<a href="#">“Resume an action workflow” on page 340</a>	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/resume-workflow

Table 200. z/OSMF software services instance services: operations summary (continued)	
Operation name	HTTP method and URI path
<a href="#">“Retry a provisioning workflow” on page 342</a>	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/retry-workflow
<a href="#">“Retry an action workflow” on page 344</a>	POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/retry-workflow
<a href="#">“Update variables in a software services instance” on page 330</a>	PUT /zosmf/provisioning/rest/version/scr/object-id/variables
<a href="#">“Get the response for an action performed against a software services instance” on page 346</a>	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>
<a href="#">“List the responses for actions performed against a software services instance” on page 351</a>	GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions
<a href="#">“Delete the response for an action performed against a software services instance” on page 354</a>	DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>

## Composite software services instances

Composite software services instances include parent and child software services instances. For more information, see [“Composite templates” on page 165](#).

## Authorization requirements

Use of the software services instance services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

In addition, the user’s z/OS user ID may need access to other resources, including those that define roles such as the domain administrator. The specific requirements for each software services instance service are described in the topic for that service. For an overview of the security requirements for cloud provisioning roles, see [“Authorization requirements” on page 48](#). For details, see [Steps for setting up security in IBM z/OS Management Facility Configuration Guide](#).

## Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 201. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.

Table 201. Response from a request failure (continued)

Field	Type	Description
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Error logging

Errors from the software services instance services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 201 Created

The request succeeded and resulted in the creation of an object.

### HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

### HTTP 204 No content

The request succeeded, but no content is available to be returned.

### HTTP 400 Bad request

The request contained incorrect parameters.

### HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

### HTTP 404 Not found

The requested resource does not exist.

### HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

## Create a software services instance

You can use this operation to create a software services instance in the registry.

## HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software services instance service. The following value is valid: 1.0.

## Query parameters

None.

## Description

This operation creates a software services instance in the registry, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content”](#) on page 289.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new software services instance. The URI path for the software object is provided in the Location response header and a response body is provided, as described in [“Response content”](#) on page 297.

## Request content

The request content is expected to contain a JSON object. [Table 202 on page 289](#) lists the fields in the JSON object.

Table 202. Request content for a create software services instance request			
Field name	Type	Required or optional	Description
<b>type</b>	String	Required	Type of the software. Up to 8 characters.
<b>registry-type</b>	String	Required	The type of software registry object: catalog or general. An object with registry-type catalog is created from a software services template in the software services catalog. When the registry type is catalog, a catalog object ID and catalog name are also required.
<b>state</b>	String	Required	The current state of the software: <ul style="list-style-type: none"><li>• being-initialized</li><li>• being-provisioned</li><li>• provisioned</li><li>• provisioning-suspended</li><li>• being-deprovisioned</li><li>• deprovisioning-suspended</li><li>• deprovisioned</li><li>• provisioning-failed</li><li>• deprovisioning-failed</li></ul>
<b>catalog-object-id</b>	String	Required when registry-type is catalog	The identifier of the software services template to be used to create the software services instance.
<b>template-owner</b>	String	Optional	The owner of the template that the registry instance was created from. This field is not valid when the value for the registry-type field is general.
<b>catalog-object-name</b>	String	Required when registry-type is catalog	The name of the software services template to be used to create the software services instance.

Table 202. Request content for a create software services instance request (continued)

Field name	Type	Required or optional	Description
<b>external-name</b>	String	Optional	The external name to identify the software registry object. If the external name is not provided then it is set from object-name in the response body. Up to 34 characters.
<b>system-nickname</b>	String	Optional	The nickname of the system the service is provisioned on and where corresponding actions will be run.
<b>system</b>	String	Optional	System on which the software is provisioned. Up to 8 characters.
<b>sysplex</b>	String	Optional	Sysplex on which the software is provisioned. Up to 8 characters.
<b>vendor</b>	String	Optional	Vendor of the software. Up to 24 characters.
<b>version</b>	String	Optional	Version of the software. Up to 24 characters.
<b>description</b>	String	Optional	Description for the software. Up to 256 characters.
<b>owner</b>	String	Optional	The user ID that identifies the owner of the software registry object. Up to 8 characters.
<b>provider</b>	String	Optional	The user ID that identifies the provider of the software, . Up to 8 characters. This is the owner of the software catalog object.
<b>quality-attributes</b>	String	Optional	The quality attributes associated with the software. Up to 16 characters.
<b>workflow-key</b>	String	Optional	The workflow key associated with provisioning the software. This field is not valid when the value for registry-type is general.
<b>workflow-clean-after-provisioned</b>	String	Optional	The indication of whether the workflow instance used to provision this instance will be removed after the workflow is completed. Must be archive, true, or false. The default is archive. This field is not valid when the value for registry-type is general.
<b>job-statement</b>	String	Optional	The JOB statement.
<b>jobs-disposition</b>	String	Optional	<p>Indicates the disposition of jobs.</p> <p><b>keep</b> Keep all completed jobs located on the JES spool from the provisioning workflow and all action workflows.</p> <p><b>delete</b> Delete all completed jobs located on the JES spool from the provisioning workflow and all action workflows.</p> <p>The default is keep.</p>

Table 202. Request content for a create software services instance request (continued)

Field name	Type	Required or optional	Description
<b>instances-disposition</b>	String	Optional	Indicates the disposition of the software instance after it is deprovisioned.  <b>keep</b> Keep the instance after it is deprovisioned.  <b>delete</b> Delete the instance after it is deprovisioned.  The default is keep.
<b>actions</b>	Action[]	Optional	The actions for the software. See <a href="#">Table 203 on page 293</a> .
<b>variables</b>	Variable[]	Optional	The variables for the software. See <a href="#">Table 204 on page 294</a> .
<b>user-data-id</b>	String	Optional	The user data ID.
<b>user-data</b>	String	Optional	The user data.
<b>domain-id</b>	String	See description.	The domain ID. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
<b>tenant-id</b>	String	See description.	The tenant ID. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
<b>domain-name</b>	String	See description.	The name of the domain. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
<b>tenant-name</b>	String	See description.	The name of the tenant. This field is not valid when the value for registry-type is general. It is required when the value for registry-type is catalog.
<b>ssin</b>	String	Optional	Software service instance name, used in generating names for software services instances. This field is not valid when the value for registry-type is general.

Table 202. Request content for a create software services instance request (continued)

Field name	Type	Required or optional	Description
<b>runAsUser-audit</b>	boolean	Optional	<p>Indicates if auditing is performed on workflows and command actions that are associated with the instance.</p> <p><b>false</b> z/OSMF performs validation to ensure that the runAsUser user ID is a valid MVS user ID. No further authorization checking is done prior to switching to the runAsUser ID.</p> <p><b>true</b> Prior to switching identities to the runAsUser user ID, z/OSMF does an authorization check for access to this resource. If the authorization is successful, the runAsUser ID has access and an audit record is generated. If the authorization check fails, no audit record is generated and switching to the runAsUser user ID does not occur. The workflow might fail.</p> <p>The default is false.</p> <p>This field is valid only when the registry-type is catalog.</p>
<b>composite-data</b>	Array of objects	Required if the instance is a parent of a composite	<p>If set, indicates that this instance is the parent of a composite. Specifies an array of composite data objects. Each object represents information about an existing catalog type registry instance (registry-type is catalog) that is a child member of this composite software services instance.</p> <p>See <a href="#">Table 206 on page 295</a>.</p>
<b>composite-cluster</b>	String	Optional	<p>Indicates if the instance is either a composite cluster parent or a member, as follows:</p> <p><b>true</b> The instance is either a composite cluster parent or a member.</p> <p><b>false</b> The instance is neither a composite cluster parent nor a member.</p>
<b>composite-parent-template-name</b>	String	Optional	Name of the template for the composite parent.
<b>composite-parent-template-id</b>	String	Optional	ID of the template for the composite parent.
<b>expiration-period</b>	String	Optional	Number of days the software services instance is kept provisioned after it is successfully provisioned. A value of "0" indicates that the instance does not expire. By default, this value is "0".

Table 203. Action structure for a "create a software services instance" request

Field	Type	Description
<b>name</b>	String	The name of the action. If the name of the action is deprovision, the action is for deprovisioning the software.  You can indicate that the action is for deprovisioning either by setting the is-deprovision field to true or by naming the action deprovision.
<b>type</b>	String	The type of the action.  The value must be one of the following: <ul style="list-style-type: none"> <li>• command</li> <li>• workflow</li> <li>• instructions</li> </ul>
<b>is-deprovision</b>	String	Indicates if the action deprovisions the software, as follows: <ul style="list-style-type: none"> <li>• If true, the action deprovisions the software.</li> <li>• If false or not set, the action does not deprovision the software. This is overridden if the value of the name field is deprovision.</li> </ul>
<b>description</b>	String	The description of the action. This field is optional. If not provided, the description is empty.
<b>command</b>	String	For command type actions, the command.
<b>command-run-as-user</b>	String	For command type actions, if provided, the user ID to be used when the command is run. This is not valid when the registry-type is general.
<b>command-sol-key</b>	String	For command type actions, if provided, the key to search for in the solicited messages command response.
<b>command-unsol-key</b>	String	For command type actions, if provided, the key to search for in the unsolicited messages.
<b>command-detect-time</b>	String	For command type actions, if provided, the time in seconds to detect for the command-unsol-key in the unsolicited messages.  Also, the minimum time before a command response is checked for after the command is submitted for execution.  If not provided, the default command-detect-time is 15 seconds when the command-unsol-key is specified or 10 seconds when the command-unsol-key is not specified.
<b>workflow-definition-file</b>	String	For workflow type actions, the workflow definition file.
<b>workflow-variable-input-file</b>	String	For workflow type actions, if provided, the workflow variable input file.
<b>variables</b>	Variable[]	For workflow type actions, if provided, the workflow variables.

Table 203. Action structure for a "create a software services instance" request (continued)

Field	Type	Description
<b>workflow-clean-after-complete</b>	String	For workflow type actions, if provided, specifies whether the instance of the workflow is deleted after it completes. The values are true, false, or inherit. If no value is provided, the value is inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance.
<b>instructions</b>	String	For instruction type actions, the instructions.
<b>prompt-variables</b>	PromptVariable[]	For workflow type actions, if provided, the prompt variables, which are the variables that will have their values prompted for at create time. See <a href="#">Table 205 on page 294</a> .
<b>at-create-variables</b>	String[]	For workflow type actions, if provided, the names of the at create variables, which are the only variables allowed on input-variables for the do action operation.
<b>command-run-as-user-dynamic</b>	boolean	Indicates if the command-run-as-user ID value can change:  <b>true</b> The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content.  <b>false</b> The command-run-as-user ID is final and cannot change during the processing of the command.

Table 204. Variable structure

Field	Type	Description
<b>name</b>	String	Name of the variable.
<b>description</b>	String	Description of the variable.
<b>value</b>	String	Value of the variable.
<b>visibility</b>	String. The value must be public or private.	Visibility of the variable.
<b>update-registry</b>	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Table 205. Response from a get request: Prompt-Variable-Object

Field	Type	Description
<b>name</b>	String	Name of the property.
<b>value</b>	String	Current value for the property.
<b>required</b>	boolean	Indicates whether the variable value is required during the workflow create process.
<b>label</b>	String	Short label for the UI widget.
<b>description</b>	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.

Table 205. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
<b>abstract</b>	String	Brief description of the variable for the UI widget.
<b>type</b>	String	Type of the variable element: boolean, string, integer, decimal, time, date.
<b>must-be-choice</b>	boolean	Indicates whether the value must come from the provided choices.
<b>choices</b>	Array of Strings	Contains allowable choices for the value of the variable.
<b>regex</b>	String	Standard regular expression that constrains the variable value.
<b>multi-line</b>	boolean	Indicates whether the value requires a multi-line text box.
<b>min</b>	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
<b>max</b>	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
<b>places</b>	String	Maximum number of decimal places that can be specified for a variable of type decimal.
<b>error-message</b>	String	Default error message associated with an incorrect value.

Table 206. Composite-data structure

Field	Type	Required/ optional	Description
<b>object-id</b>	String	Required	Object ID that is associated with the existing child instance.
<b>sequence</b>	Integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
<b>published-template-name</b>	String	Required	The name of an existing published template in the domain that is associated with the composite template.
<b>connectors</b>	Array of objects	Optional	<p>An array of connector object.</p> <p>The connector variables specified here take precedence over the variables field in this object and any variables that are specified in the workflow variable input file that is associated with the published template.</p> <p>See <a href="#">Table 207 on page 296</a>.</p>

Table 206. Composite-data structure (continued)

Field	Type	Required/ optional	Description
<b>variables</b>	Array of objects	Optional	<p>A list of one or more variables for the provisioning workflow that is associated with this published template. The variables specified here take precedence over the variables that are specified in the workflow variable input file.</p> <p>Specify this property as an array of name-value objects, for example:</p> <pre>"variables": [ {"name":"user_name","value":"IBMUSER"}, {"name":"file_name","value":"textfile.txt"} ]</pre>
<b>workflow-definition-file</b>	String	Required	The absolute path for the provisioning workflow definition file.
<b>variable-input-file</b>	String	Optional	The absolute path for the variable input file that is associated with the provisioning workflow definition file.

Table 207. Connector object

Field	Type	Required/ optional	Description
variable-name	String	Required	The name of an atCreate variable that is associated with this published template name, the value of which will be overridden with the value of the source-variable-name field. If the connector variable-name is also a prompt variable, then the connector takes precedence and the variable is no longer promptable.
source-template	String	Required	The name of a standard template from which the overriding source variable name is obtained. The sequence number of the composite object that is associated with the source template must be lower than the sequence number of this composite object. If a template occurs multiple times in the sequence, values for variables come from the first occurrence of the template.
source-variable-name	String	Required	The name of the variable that is associated with the source template or constant registry-instance-Name. The value of registry-instance-Name resolves to the name of the registry instances created for the source template.
not-valid	boolean	Required	<p>Indicates if the information (variable-name, source-template, and source-variable-name values) in this connector is valid. The value is:</p> <ul style="list-style-type: none"> <li>• false, if all of the information is accurate</li> <li>• true, if one or more of the values are incorrect.</li> </ul>

## Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 287](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 297](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 208. HTTP error response codes for a create software services instance request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

## Response content

On successful completion, the service returns the following:

- URI path of the created software services instance in the Location response header.
- Response body, which contains a JSON object with details about the software services instance. [Table 209 on page 297](#) lists the fields in the JSON object.

Table 209. Response from a create software services instance request		
Field	Type	Description
object-id	String	The object ID of the newly created object. The object ID is to be used on further requests to the object.
object-uri	String	The object URI of the newly created object.
object-name	String	The object name of the newly created object.
external-name	String	The external name of the newly created object.
system-nickname	String	Nickname of the system that the service is provisioned on.
cluster-instance-name	String	The name of the cluster instance that this entry belongs to.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 210. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.

<i>Table 210. Response from a request failure (continued)</i>		
<b>Field</b>	<b>Type</b>	<b>Description</b>
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interactions

In [Figure 119 on page 299](#), a request is submitted to create a software services instance on the system SY1.

```

{
  "type": "DB2",
  "external-name": "DB2B",
  "vendor": "IBM",
  "version": "V5R10",
  "description": "DB2 for test1",
  "registry-type": "catalog",
  "catalog-object-id": "9f7c659e-38f5-4585-b9f9-9cd448bf9cc3",
  "catalog-object-name": "DB2template1",
  "template-owner": "ZOSMFAD",
  "workflow-key": "02e1ec78-e0db-482b-8013-3d435b52e2e3",
  "workflow-clean-after-provision": "true",
  "system-nickname": "SYSTEM1",
  "system": "SY1",
  "sysplex": "PLEX1",
  "state": "being-provisioned",
  "owner": "ZOSMFAD",
  "provider": "ZOSMFAD",
  "quality-attributes": "123456789ABCDEF0",
  "user-data": "my data",
  "user-data-id": "udid1",
  "domain-id": "izu$0",
  "tenant-id": "izu$002",
  "domain-name": "default",
  "tenant-name": "default",
  "ssin": "SSIN1",
  "composite-cluster": "true",
  "composite-parent-template-id": "2ed65dd8-7c4a-4029-8e37-576714df38ee",
  "composite-parent-template-name": "c5",
  "variables": [
    {
      "name": "IACTION_NAME",
      "value": "Instructions1",
      "visibility": "public"
    },
    {
      "name": "COMMAND1",
      "value": "d a,l",
      "visibility": "public"
    },
    {
      "name": "C_DETECT_TIME",
      "value": "25",
      "visibility": "public"
    },
    {
      "name": "C_SOL_K",
      "value": "VLF",
      "visibility": "public"
    },
    {
      "name": "C_UNSQL_K",
      "value": "CSV",
      "visibility": "public"
    }
  ],
  "actions": [
    {
      "name": "Instructions1",
      "type": "instructions",
      "is-deprovision": "false",
      "instructions": "These are the instructions for the ${IACTION_NAME} action."
    },
    {
      "name": "command1",
      "type": "command",
      "is-deprovision": "false",
      "command": "${COMMAND1}",
      "command-detect-time": "${C_DETECT_TIME}",
      "command-run-as-user": "IBMUSER",
      "command-sol-key": "${C_SOL_K}",
      "command-unsol-key": "${C_UNSQL_K}"
    },
    {
      "name": "deprovision",
      "type": "instructions",
      "is-deprovision": "true",
      "instructions": "Do the deprovision manually."
    }
  ]
}

```

Figure 119. Sample request to create a software services instance

The response is shown below.

```

{
  "object-name": "DB2_1",
  "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9",
  "object-uri": "/zosmf/provisioning/rest/1.0/scr/c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9",
  "external-name": "DB2_DY01",
  "cluster-instance-name": "Y0",

```

```
"system-nickname": "SY1"
}
```

In Figure 120 on page 300, a request is submitted to create a composite software services instance.

```
{
  "type": "forCics",
  "description": "composite for Cics",
  "registry-type": "catalog",
  "catalog-object-id": "9f7c659e-38f5-4585-b9f9-9cd448bf9cc3",
  "catalog-object-name": "cics_comp",
  "state": "being-provisioned",
  "domain-id": "izu0",
  "tenant-id": "izu002",
  "domain-name": "default",
  "tenant-name": "default",
  "composite-data": [
    {
      "sequence": 1,
      "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9",
      "published-template-name": "mq",
      "connectors": [],
      "variables": [
        { "name": "user_name", "value": "IBMUSER" },
        { "name": "defect_status", "value": "approved" }
      ],
      "deprovisioning-action": "deprovision_2",
      "workflow-definition-file": "/users/gg/zosmf/IzuProvisioning/IzuScc/mq.1.default/definition/p.xml"
    },
    {
      "sequence": 2,
      "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94fa",
      "published-template-name": "cics",
      "connectors": [
        {
          "variable-name": "cics_mq_ssn",
          "source-template": "mq",
          "source-variable-name": "mq_ssn"
        }
      ],
      "variables": [
        { "name": "defect_status", "value": "approved" }
      ],
      "deprovisioning-action": "deprovision_2",
      "workflow-definition-file": "/users/gg/zosmf/IzuProvisioning/IzuScc/cics.1.default/definition/p.xml",
      "variable-input-file": "/users/gg/zosmf/IzuProvisioning/IzuScc/cics.1.default/definition/var.properties"
    }
  ]
}
```

Figure 120. Sample request for a composite

## Get the contents of a software services instance

You can use this operation to retrieve the contents of a software services instance.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be retrieved.

### Query parameters

None.

### Description

This operation retrieves the properties of a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 212 on page 302](#).

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 212 on page 302](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 211. HTTP error response codes for a get software services instance contents request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found; the software services instance does not exist.

## Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 212](#) on page 302 lists the fields in the JSON object.

Table 212. JSON object that is returned for a get software services instance property request		
Field	Type	Description
<b>object-id</b>	String	The object-id for the software services instance
<b>object-name</b>	String	The object-name for the software services instance
<b>type</b>	String	Type of the software. The value is null for composite parent registry instances.
<b>registry-type</b>	String	Type of registry object: catalog or general
<b>external-name</b>	String	External name of the software services instance
<b>system-nickname</b>	String	The nickname of the system that the software is provisioned on.
<b>system</b>	String	System that the software is provisioned on.
<b>sysplex</b>	String	Sysplex that the software is provisioned on.
<b>last-known-system</b>	String	The name of the system on which the software for the instance was last known to be running.
<b>last-known-system-nickname</b>	String	The nickname of the system on which the software for the instance was last known to be running.
<b>vendor</b>	String	Vendor of the software. Null for a composite instance.
<b>version</b>	String	Version of the software. Null for a composite instance.
<b>description</b>	String	Description for the software
<b>owner</b>	String	The user ID that identifies the owner of the software
<b>provider</b>	String	The user ID that identifies the provider of the software
<b>template-owner</b>	String	The owner of the template that the registry instance was created from.
<b>catalog-object-id</b>	String	The identifier of the template that is used when partitioning the software represented by this instance. Only valid when registry-type is catalog.
<b>catalog-object-name</b>	String	The name of the template that was used when partitioning the software represented by this instance.
<b>workflow-key</b>	String	The workflow key that is associated with provisioning the software.

Table 212. JSON object that is returned for a get software services instance property request (continued)

Field	Type	Description
<b>workflow-clean-after-provisioned</b>	String	The indication of whether the workflow instance used to provision this instance is removed after the workflow is completed. Must be archive, true, or false. The default is archive. This field is not valid when the value for registry-type is general.
<b>job-statement</b>	String	The JOB statement.
<b>jobs-disposition</b>	String	Indicates the disposition of jobs.  <b>keep</b> Keep all completed jobs located on the JES spool from the provisioning workflow and all action workflows.  <b>delete</b> Delete all completed jobs located on the JES spool from the provisioning workflow and all action workflows.  The default is keep.
<b>instances-disposition</b>	String	Indicates the disposition of the software instance after it is deprovisioned.  <b>keep</b> Keep the instance after it is deprovisioned.  <b>delete</b> Delete the instance after it is deprovisioned.  The default is keep.
<b>state</b>	String	The current state of the software: <ul style="list-style-type: none"> <li>• being-initialized</li> <li>• being-provisioned</li> <li>• provisioned</li> <li>• provisioning-suspended</li> <li>• being-deprovisioned</li> <li>• deprovisioning-suspended</li> <li>• deprovisioned</li> <li>• provisioning-failed</li> <li>• deprovisioning-failed</li> </ul>
<b>quality-attributes</b>	String	The quality attributes of the software
<b>actions</b>	Action[]	The actions for the software. <a href="#">Table 213 on page 306</a>
<b>variables</b>	Variable[]	The variables for the software. See <a href="#">Table 214 on page 308</a> .
<b>workflow-start-time</b>	String	The time that workflow processing started, in ISO8601 format.  The value is null if the workflow was not started.

Table 212. JSON object that is returned for a get software services instance property request (continued)

Field	Type	Description
<b>workflow-stop-time</b>	String	The time that workflow automation last stopped, in ISO8601 format.  The value is null if the workflow automation has not stopped.
<b>workflow-current-step-name</b>	String	The name of the step that is being processed automatically in the provisioning workflow or action workflow. This field is set if workflow automation is in progress or stopped.
<b>workflow-current-step-number</b>	String	The number of the step that is being processed automatically in the provisioning workflow or action workflow. This field is set if workflow automation is in progress or stopped.
<b>workflow-total-steps</b>	String	The total number of steps in the workflow.
<b>created-time</b>	String	The time the object was created. The time is in the ISO8601 format.
<b>last-modified-time</b>	String	The time the object was updated. The time is in the ISO8601 format.  This field might be updated during creation, and the value might be later than the value for the created-time field.
<b>expiration-period</b>	String	Number of days the software services instance will be kept provisioned after it is successfully provisioned. A value of "0" indicates that the instance does not expire. By default, this value is "0". This field is optional.
<b>has-expired</b>	String	Indicates whether the provisioned instance is expired ("true" or "false").
<b>expiration-date</b>	String	The date and time the instance expires. The date is in the ISO8601 format. By default, this value is null. This field is optional.
<b>expiration-upcoming-notified</b>	String	The time the consumer was notified of the upcoming instance expiration. The time is in the ISO8601 format.
<b>expiration-process-notified</b>	String	The time the consumer was notified that the instance expired. The time is in ISO8601 format.
<b>expiration-processed-reminder-notified</b>	String	The time the consumer and domain administrators were last sent a reminder that the instance is expired. The time is in ISO8601 format.
<b>created-by-user</b>	String	The user ID that created the object
<b>last-modified-by-user</b>	String	The user ID that last updated the object
<b>last-action-name</b>	String	The name of the last action that was performed.
<b>last-action-object-id</b>	String	The action ID of the last action that was performed.

Table 212. JSON object that is returned for a get software services instance property request (continued)

Field	Type	Description
<b>last-action-state</b>	String	The state of the last action that was performed.
<b>user-data-id</b>	String	The user data ID.
<b>user-data</b>	String	The user data.
<b>domain-id</b>	String	The domain ID.
<b>tenant-id</b>	String	The tenant ID.
<b>domain-name</b>	String	The name of the domain.
<b>tenant-name</b>	String	The name of the tenant.
<b>ssin</b>	String	Software service instance name, used in generating names for software services instances.
<b>runAsUser-audit</b>	boolean	<p>Indicates if auditing is performed on workflows and command actions that are associated with the instance.</p> <p><b>false</b> z/OSMF performs validation to ensure that the runAsUser user ID is a valid MVS user ID. No further authorization checking is done prior to switching to the runAsUser ID.</p> <p><b>true</b> Prior to switching identities to the runAsUser user ID, z/OSMF does an authorization check for access to this resource. If the authorization is successful, the runAsUser ID has access and an audit record is generated. If the authorization check fails, no audit record is generated and switching to the runAsUser user ID does not occur. The workflow might fail.</p> <p>The default is false.</p> <p>This field is valid only when the registry-type is catalog.</p>
<b>composite-instance-data</b>	Array of objects	<p>Array of composite instance data objects. Each object represents information about an existing catalog type registry instance that is a member of this registry instance. This includes various properties of that instance.</p> <p>See <a href="#">Table 216 on page 309</a>.</p>
<b>composite-parent-object-id</b>	String	If set, indicates that this instance is a child member of a composite. Specifies the object ID of the registry instance that is the parent of the composite.

Table 212. JSON object that is returned for a get software services instance property request (continued)		
Field	Type	Description
<b>composite-children</b>	Array of objects	If set, indicates that this instance is the parent of a composite. Specifies an array of composite child objects. Each object contains information about an existing catalog type registry instance that is a child member of this composite software services instance.  See <a href="#">Table 217 on page 309</a> .
<b>object-uri</b>	String	The object URI of the instance.
<b>workflow-message-text</b>	String	If set, the message text that is associated with the provisioning workflow.
<b>account-information</b>	String	The account information.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>composite-cluster</b>	String	Indicates whether the instance is either a composite cluster parent or a member, as follows:  <b>true</b> The instance is either a composite cluster parent or a member.  <b>false</b> The instance is not a composite cluster parent or a member.
<b>cluster-instance-name</b>	String	Name of the cluster instance.
<b>composite-parent-template-name</b>	String	Name of the template for the composite parent.
<b>composite-parent-template-id</b>	String	ID of the template for the composite parent.

Table 213. Action structure for a "get the contents of a software services instance" request		
Field	Type	Description
<b>name</b>	String	The name of the action. If the name of the action is deprovision, the action is for deprovisioning the software.  You can indicate that the action is for deprovisioning either by setting the is-deprovision field to true or by naming the action deprovision.
<b>type</b>	String	The type of the action.  The value must be one of the following: <ul style="list-style-type: none"> <li>• command</li> <li>• workflow</li> <li>• instructions</li> </ul>

Table 213. Action structure for a "get the contents of a software services instance" request (continued)

Field	Type	Description
<b>is-deprovision</b>	String	Indicates if the action deprovisions the software, as follows: <ul style="list-style-type: none"> <li>• If true, the action deprovisions the software.</li> <li>• If false or not set, the action does not deprovision the software. This is overridden if the value of the name field is deprovision.</li> </ul>
<b>description</b>	String	The description of the action. This field is optional. If not provided, the description is empty.
<b>command</b>	String	For command type actions, the command.
<b>command-run-as-user</b>	String	For command type actions, if provided, the user ID to be used when the command is run. This is not valid when the registry-type is general.
<b>command-sol-key</b>	String	For command type actions, if provided, the key to search for in the solicited messages command response.
<b>command-unsol-key</b>	String	For command type actions, if provided, the key to search for in the unsolicited messages.
<b>command-detect-time</b>	String	For command type actions, if provided, the time in seconds to detect for the command-unsol-key in the unsolicited messages.  Also, the minimum time before a command response is checked for after the command is submitted for execution.  If not provided, the default command-detect-time is 15 seconds when the command-unsol-key is specified or 10 seconds when the command-unsol-key is not specified.
<b>workflow-definition-file</b>	String	For workflow type actions, the workflow definition file.
<b>workflow-variable-input-file</b>	String	For workflow type actions, if provided, the workflow variable input file.
<b>variables</b>	Variable[]	For workflow type actions, if provided, the workflow variables.
<b>workflow-clean-after-complete</b>	String	For workflow type actions, if provided, specifies whether the instance of the workflow is deleted after it completes. The values are true, false, or inherit. If no value is provided, the value is inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance.
<b>instructions</b>	String	For instruction type actions, the instructions.
<b>prompt-variables</b>	PromptVariable[]	For workflow type actions, if provided, the prompt variables, which are the variables that are expected to be prompted for in preparation for running the software services template. See Table 215 on page 308.

Table 213. Action structure for a "get the contents of a software services instance" request (continued)

Field	Type	Description
<b>at-create-variables</b>	String[]	For workflow type actions, if provided, these are the names of the variables that are either prompt variables (variables that are expected to be prompted for in preparation for running the software services template), or required variables (variables for which a value is required when the software services template is run), or both. These are the only variables allowed on input-variables for the do action operation.
<b>command-run-as-user-dynamic</b>	boolean	Indicates if the command-run-as-user ID value can change:  <b>true</b> The command-run-as-user ID value is not final. It can change through variable substitution prior to the processing of the command, based on the provisioning workflow content.  <b>false</b> The command-run-as-user ID is final and cannot change during the processing of the command.
<b>email</b>	String	Email.

Table 214. Variable structure

Field	Type	Description
<b>name</b>	String	Name of the variable.
<b>description</b>	String	Description of the variable.
<b>value</b>	String	Value of the variable.
<b>visibility</b>	String. The value must be public or private.	Visibility of the variable.
<b>update-registry</b>	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Table 215. Response from a get request: Prompt-Variable-Object

Field	Type	Description
name	String	Name of the property.
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.

Table 215. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

Table 216. Composite-instance data

Field	Type	Required/optional	Description
<b>sequence</b>	Integer	Required	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
<b>object-id</b>	String	Required	Object ID that is associated with the existing child instance.
<b>catalog-object-name</b>	String	Optional	The published template name that is associated with the child software service instance.
<b>state</b>	String		The current state of the child software service instance.
<b>variables</b>	Variable[]		The variables for the child software service instance. See <a href="#">Table 214 on page 308</a> .
<b>actions</b>	Action[]		The actions for the child software service instance. See <a href="#">Table 213 on page 306</a> .
<b>workflow-key</b>	String		The workflow key that is associated with provisioning the child software service instance.

Table 217. Composite child object

Field	Type	Description
<b>sequence</b>	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
<b>object-id</b>	String	Object ID that is associated with the existing child instance.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 218. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the GET method is used to retrieve information about a software services instance. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 350386fc-0f9e-414c-b343-847c300dff8d.

```
GET /zosmf/provisioning/rest/ 1.0/scr/350386fc-0f9e-414c-b343-847c300dff8d
```

Figure 121. Sample request to get software services instance properties

The following is an example of the response.

```
{
  "type": "DB2",
  "system": "SY1",
  "sysplex": "PLEX1",
  "vendor": "IBM",
  "version": "V5R10",
  "description": "DB2for test1",
  "owner": "ibmuser",
  "provider": "ibmuser",
  "state": "provisioned",
  "variables": [
    {
      "name": "IACTION_NAME",
      "value": "Instructions1",
      "visibility": "public",
      "update-registry": "false"
    }
  ],
  "actions": [
    {
      "name": "Instructions1",
      "type": "instructions",
      "command": null,
      "instructions": "These are the instructions for the ${IACTION_NAME} action.",
      "is-deprovision": "false",
      "command-run-as-user": null,
      "command-sol-key": null,
      "command-unsol-key": null,
      "command-detect-time": null,
      "workflow-definition-file": null,
      "workflow-variable-input-file": null,
      "variables": null,
      "workflow-clean-after-complete": null,
      "prompt-variables": null,
      "at-create-variables": null
    }
  ],
  "ssin": "DB2001",
  "object-id": "350386fc-0f9e-414c-b343-847c300dff8d",
  "object-name": "DB2_1",
  "object-uri": "/zosmf/provisioning/rest/1.0/scr/350386fc-0f9e-414c-b343-847c300dff8d",
```

```

"registry-type": "catalog",
"external-name": "DB2_DB2001",
"system-nickname": "SY1",
"last-known-system-nickname": "SY1",
"last-known-system": "SY1",
"catalog-object-id": "7e147191-8519-402d-a31a-59e978e5a0ee",
"catalog-object-name": "A1",
"quality-attributes": "123456789ABCDEF0",
"workflow-key": "02e1ec78-e0db-482b-8013-3d435b52e2e3",
"workflow-clean-after-provisioned": "false",
"jobs-disposition": "keep",
"created-time": "2018-06-12T15:24:55.695Z",
"last-modified-time": "2018-06-12T15:24:58.320Z",
"created-by-user": "ibmuser",
"last-modified-by-user": "ibmuser",
"last-action-name": null,
"last-action-object-id": null,
"last-action-state": null,
"user-data": "my data",
"user-data-id": "udid1",
"tenant-id": "IYU000",
"tenant-name": "default",
"domain-id": "IYU0",
"domain-name": "default",
"job-statement": "",
"account-info": null,
"runAsUser-audit": "true",
"workflow-start-time": "2018-06-12T15:24:56.025Z",
"workflow-stop-time": "2018-06-12T15:24:56.040Z",
"composite-children": null,
"composite-instance-data": null,
"composite-parent-object-id": null,
"composite-cluster": "true",
"composite-parent-template-id": "2ed65dd8-7c4a-4029-8e37-576714df38ee",
"composite-parent-template-name": "c5",
"cluster-instance-name": "Y0"

"provisioning-version": "1300",
"workflow-message-text":
  "IZUWF0162I: Automation processing for workflow \"DB2_DB2001provision1528817095765\" is
complete.",
"template-owner": "ibmuser",
"workflow-current-step-name": "",
"workflow-current-step-number": "",
"workflow-total-steps": "1"
}

```

## List the software services instances

You can use this operation to list the software services instances in the software services registry.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/scr
```

---

In this request, the URI path variable is as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.

### Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

**type**

Optional, specifies the type of the software.

**registry-type**

Optional, specifies the type of the registry object: Catalog or general.

**object-name**

Optional, regular expression, specifies the name for the software services instance.

**external-name**

Optional, regular expression, specifies the external name of the software.

**system**

Optional, specifies the system on which the software is provisioned.

**sysplex**

Optional, specifies the sysplex on which the software is provisioned.

**vendor**

Optional, specifies the vendor of the software.

**owner**

Optional, specifies the user ID that identifies the owner of the software.

**provider**

Optional, specifies the user ID that identifies the provider of the software.

**state**

Optional, specifies the current state of the software:

- being-initialized
- being-provisioned
- provisioned
- provisioning-suspended
- being-deprovisioned
- deprovisioning-suspended
- deprovisioned
- provisioning-failed
- deprovisioning-failed

**catalog-object-id**

Optional, specifies the catalog object ID associated with the creation of this software services instance.

**user-data-id**

Optional, specifies the ID for the user data.

**domain-id**

Optional, specifies the ID of the domain.

**tenant-id**

Optional, specifies the ID of the tenant.

**tenant-name**

Optional, regular expression that specifies the name of the tenant.

**domain-name**

Optional, regular expression that specifies the name of the domain.

If you specify no query parameters, then all objects are returned.

**Description**

This operation returns the object ID and a subset of fields for software services instances.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 313](#).

**Authorization requirements**

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

For catalog registry type objects, the user that issues the request must be at least one of the following for a software services instance to be returned in the list:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 287](#).

**HTTP status codes**

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 313](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 219. HTTP error response codes for a list software services instances request</i>	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained incorrect parameters.
<b>HTTP 401 Unauthorized</b>	The requester user ID is not authorized for this request.

**Response content**

On successful completion, the response body contains a JSON object, named `scr-list`, of registry objects that consist of a subset of the fields for all software services instances matching the query. [Table 220 on page 314](#) lists the fields in the JSON object.

Table 220. JSON object that is returned for a list software services instances request

Field	Type	Description
<b>object-id</b>	String	The object ID for the software services instance.
<b>object-name</b>	String	The object name for the software services instance.
<b>type</b>	String	The type of the software, for example, the subsystem name, such as Db2 or IBM MQ. Not returned for composite parent registry instances.
<b>registry-type</b>	String	Type of registry object: Catalog or general.
<b>external-name</b>	String	External name of the software services instance.
<b>template-owner</b>	String	The owner of the template that the registry instance was created from.
<b>catalog-object-id</b>	String	The identifier of the software services template used when partitioning the software represented by this software services instance. Valid only when the value for registry-type is catalog.
<b>catalog-object-name</b>	String	The name of the template that was used when partitioning the software represented by this software services instance.
<b>system</b>	String	Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member..
<b>sysplex</b>	String	Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility..
<b>system-nickname</b>	String	The nickname of the system, indicating the system and sysplex on which the instance was provisioned.
<b>last-known-system</b>	String	The name of the system on which the software for the instance was last known to be running.
<b>last-known-system-nickname</b>	String	The nickname of the system on which the software for the instance was last known to be running.
<b>vendor</b>	String	Vendor of the software.
<b>version</b>	String	Version of the software.
<b>description</b>	String	Description for the software.
<b>owner</b>	String	The user ID that identifies the owner of the software.
<b>provider</b>	String	The user ID that identifies the provider of the software.

Table 220. JSON object that is returned for a list software services instances request (continued)

Field	Type	Description
<b>state</b>	String	The current state of the software: <ul style="list-style-type: none"> <li>• being-initialized</li> <li>• being-provisioned</li> <li>• provisioned</li> <li>• provisioning-suspended</li> <li>• being-deprovisioned</li> <li>• deprovisioning-suspended</li> <li>• deprovisioned</li> <li>• provisioning-failed</li> <li>• deprovisioning-failed</li> </ul>
<b>created-time</b>	String	The time the object was created. The time is in the ISO8601 format.
<b>last-modified-time</b>	String	The time the object was updated. The time is in the ISO8601 format.
<b>expiration-period</b>	String	Number of days the software services instance will be kept provisioned after it is successfully provisioned. A value of "0" indicates that the instance does not expire. By default, this value is "0". This field is optional.
<b>has-expired</b>	String	Indicates whether the provisioned instance is expired ("true" or "false").
<b>expiration-date</b>	String	The date and time the instance expires. The date is in the ISO8601 format. By default, this value is null. This field is optional.
<b>expiration-upcoming-notified</b>	String	The time the consumer was notified of the upcoming instance expiration. The time is in the ISO8601 format.
<b>expiration-process-notified</b>	String	The time the consumer was notified that the instance expired. The time is in ISO8601 format.
<b>expiration-processed-reminder-notified</b>	String	The time the consumer and domain administrators were last sent a reminder that the instance is expired. The time is in ISO8601 format.
<b>created-by-user</b>	String	The user ID that created the object.
<b>last-modified-by-user</b>	String	The user ID that last updated the object.
<b>last-action-name</b>	String	The name of the last action that was performed.
<b>last-action-object-id</b>	String	The action ID of the last action that was performed.
<b>last-action-state</b>	String	The state of the last action that was performed.
<b>user-data-id</b>	String	The user data ID.
<b>tenant-id</b>	String	The tenant ID.
<b>domain-id</b>	String	The domain ID.
<b>tenant-name</b>	String	The name of the tenant.
<b>domain-name</b>	String	The name of the domain.

Table 220. JSON object that is returned for a list software services instances request (continued)

Field	Type	Description
<b>composite-children</b>	Array of objects	If set, indicates that this instance is the parent of a composite. Specifies an array of composite child objects. Each object contains information about an existing catalog type registry instance that is a child member of this composite software services instance.  See <a href="#">Table 221 on page 316</a> .
<b>composite-parent-object-id</b>	String	If set, indicates that this instance is a child member of a composite. Specifies the object ID of the registry instance that is the parent of the composite.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"> <li>• true if the operations are allowed</li> <li>• false if the operations are not allowed.</li> </ul>
<b>composite-cluster</b>	String	Indicates whether the instance is either a composite cluster parent or a member, as follows: <p><b>true</b> The instance is either a composite cluster parent or a member.</p> <p><b>false</b> The instance is not a composite cluster parent or a member.</p>
<b>cluster-instance-name</b>	String	Name of the cluster instance.

Table 221. Composite child object

Field	Type	Description
<b>sequence</b>	Integer	The order in which to provision the templates, starting with 1. The deprovisioning order is the reverse.
<b>object-id</b>	String	Object ID that is associated with the existing child instance.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 222. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.

Table 222. Response from a request failure (continued)		
Field	Type	Description
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In [Figure 122 on page 317](#), the GET method is used to list the software services instances.

```
GET /zosmf/provisioning/rest/ 1.0/scr
```

*Figure 122. Sample request to list software services instances*

[Figure 123 on page 318](#) shows the response.

```

{
  "scr-list": [
    {
      "object-id": "76963ea5-81a4-42d6-99d6-f3e19747cf61",
      "object-name": "DB2_1",
      "system": "SYS1",
      "sysplex": "PLEX1",
      "type": "DB2",
      "last-known-system-nickname": "SY1",
      "system-nickname": "SY1",
      "last-known-system": "SY1",
      "vendor": "IBM",
      "version": "V1R0",
      "owner": "admin1",
      "provider": "provd1",
      "registry-type": "catalog",
      "catalog-object-id": "9f7c659e-38f5-4585-b9f9-9cd448bf9cc3",
      "catalog-object-name": "DB2template1",
      "state": "provisioned",
      "user-data-id": "udid1",
      "description": "DB2for test1",
      "created-time": "2018-06-12T14:32:42.551Z",
      "last-modified-time": "2018-06-12T14:58:12.452Z",
      "created-by-user": "admin1",
      "last-modified-by-user": "admin1",
      "last-action-name": "Instructions1",
      "last-action-object-id": "672c54bd-6c2b-49fd-ad3b-2ec027f54089",
      "last-action-state": "complete",
      "domain-id": "izu$0",
      "tenant-id": "izu$002",
      "tenant-name": "default",
      "domain-name": "default",
      "composite-cluster": "true",
      "provisioning-version": "1300",
      "provisioning-version-supported": true,
      "template-owner": "ibmuser"
    },
    {
      "object-id": "cb98c8b1-9753-4e5f-86cb-169565b4f606",
      "object-name": "DB2_2",
      "system": "SYS1",
      "sysplex": "PLEX1",
      "type": "DB2",
      "last-known-system-nickname": "SY1",
      "system-nickname": "SY1",
      "last-known-system": "SY1",
      "vendor": "IBM",
      "version": "V1R0",
      "owner": "admin1",
      "provider": "provd1",
      "registry-type": "catalog",
      "catalog-object-id": "9f7c659e-38f5-4585-b9f9-9cd448bf9cc3",
      "catalog-object-name": "DB2template1",
      "state": "being-provisioned",
      "user-data-id": "udid2",
      "description": "DB2for test2",
      "created-time": "2018-06-12T17:26:17.128Z",
      "last-modified-time": "2018-06-12T17:26:17.128Z",
      "created-by-user": "admin1",
      "last-modified-by-user": "admin1",
      "last-action-name": "Instructions2",
      "last-action-object-id": "472c54bd-6c2b-49fd-ad3b-2ec027f54080",
      "last-action-state": "complete",
      "domain-id": "izu$0",
      "tenant-id": "izu$002",
      "tenant-name": "default",
      "domain-name": "default",
      "composite-parent-object-id": null,
      "composite-children": [
        {
          "sequence": 1,
          "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9"
        },
        {
          "sequence": 2,
          "object-id": "c7156cbf-e1ce-4f05-b7c7-96d73dfb94f9"
        }
      ],
      "composite-cluster": "true",
      "provisioning-version": "1300",
      "provisioning-version-supported": true,
      "template-owner": "ibmuser"
    }
  ]
}

```

Figure 123. Sample response from a list software services instances request



## Get the variables for a software services instance

You can use this operation to retrieve the variables for a software services instance.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/variables
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be retrieved.

### Query parameters

You can specify the following query parameter on this request:

#### name

Use this optional parameter to specify variable names. You can use regular expressions.

#### visibility

Use this optional parameter to specify the visibility of the variables.

### Description

This operation retrieves the variables for a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 321](#).

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 321](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 223. HTTP error response codes for a get software services instance variables request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.

Table 223. HTTP error response codes for a get software services instance variables request (continued)	
HTTP error status code	Description
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

## Response content

On successful completion, the response body contains a JSON object consisting of an array of the variable names and values for the software services instance, described in the tables that follow.

Table 224. Get variables request: Format of the variables object		
Field name	Type	Description
<b>variables</b>	Array of variables	Variables for the software services instance. Refer to <a href="#">Table 225 on page 321</a>

Table 225. Variable structure		
Field	Type	Description
<b>name</b>	String	Name of the variable.
<b>description</b>	String	Description of the variable.
<b>value</b>	String	Value of the variable.
<b>visibility</b>	String. The value must be public or private.	Visibility of the variable.
<b>update-registry</b>	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 226. Response from a request failure		
Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the GET method is used to retrieve variables for a software object. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
GET /zosmf/provisioning/rest/1.0/76963ea5-81a4-42d6-99d6-f3e19747cf61/variables
```

*Figure 124. Sample request to get software services instance variables*

An example of the response is shown in the figures that follow.

```
{
  "variables": [
    {
      "name": "INS",
      "description": "This is some text that describes the variable.",
      "value": "Instructions",
      "visibility": "public",
      "update-registry": "false"
    }
  ]
}
```

*Figure 125. Sample response from a get software services instance variables request*

## Get key-value variables for a software services instance

You can use this operation to retrieve the variables for a software services instance in key-value format.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/key-value-variables
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance to be retrieved.

### Query parameters

You can specify the following query parameter on this request:

#### name

Use this optional parameter to specify variable names. You can use regular expression.

### Description

This operation retrieves the variables for a z/OSMF software services instance in key-value format.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [“Response content” on page 324](#).

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: **<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES**.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 200 (normal) is returned and the response body is provided, as described in [“Response content” on page 324](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 227. HTTP error response codes for a get software services instance key-value variables request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requester user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance instance was not found; the software services instance does not exist.

## Response content

On successful completion, the response body contains a JSON object that consists of an array of the variable names and values for the software services instance, described in the tables that follow.

Table 228. Get key-value variables request: Format of the variables object		
Field name	Type	Description
<b>variables</b>	List of variable names and values in key-value pair format.  Example: <pre>{ "var1": "val1",   "var2": "val2" }</pre>	Variables for the software services instance. Only variables with public visibility are returned.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 229. Response from a request failure		
Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the GET method is used to retrieve key-value variables for a software object. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
GET /zosmf/provisioning/rest/1.0/76963ea5-81a4-42d6-99d6-f3e19747cf61/key-value-variables
```

Figure 126. Sample request to get software services instance variables in key-value format

The following is an example of the response.

```
{  
  "variables": {  
    { "var1": "val1",  
      "var2": "val2" }  
  }  
}
```

Figure 127. Sample response from a get key-value variables request

## Update a software services instance

You can use this operation to update fields in a software services instance.

### HTTP method and URI path

```
PUT /zosmf/provisioning/rest/<version>/scr/<object-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be updated.

### Query parameters

None.

### Description

This operation updates a software services instance.

### Request content

The request content is expected to contain a JSON object containing the fields to be updated. [Table 230](#) on page 325 lists the fields that are valid.

Table 230. Request content for an update software services instance request			
Field name	Type	Required or optional	Description
<b>state</b>	String	Optional	The current state of the software: <ul style="list-style-type: none"><li>• being-initialized</li><li>• being-provisioned</li><li>• provisioned</li><li>• provisioning-suspended</li><li>• being-deprovisioned</li><li>• deprovisioning-suspended</li><li>• deprovisioned</li><li>• provisioning-failed</li><li>• deprovisioning-failed</li></ul>
<b>external-name</b>	String	Optional	The external name to identify the software registry object. Up to 34 characters.
<b>system</b>	String	Optional	System on which the software is provisioned, up to eight characters.  Cannot be updated if the registry type is catalog for the software services instance.

Table 230. Request content for an update software services instance request (continued)

Field name	Type	Required or optional	Description
<b>sysplex</b>	String	Optional	Sysplex on which the software is provisioned, up to eight characters  Cannot be updated if the registry type is catalog for the software services instance.
<b>vendor</b>	String	Optional	Vendor of the software, up to 24 characters  Cannot be updated if the registry type is catalog for the software services instance.
<b>version</b>	String	Optional	Version of the software, up to 24 characters  Cannot be updated if the registry type is catalog for the software services instance.
<b>description</b>	String	Optional	Description for the software, up to 256 characters
<b>owner</b>	String	Optional	The user ID that identifies the owner of the software registry object, up to eight characters  Cannot be updated if the registry type is catalog for the software services instance.
<b>provider</b>	String	Optional	The user ID that identifies the provider of the software, up to eight characters. This is the owner of the software catalog object.  Cannot be updated if the registry type is catalog for the software services instance.
<b>quality-attributes</b>	String	Optional	The quality attributes associated with the software, up to 16 characters  Cannot be updated if the registry type is catalog for the software services instance.
<b>workflow-key</b>	String	Optional	The workflow key associated with provisioning the software. This field is not valid when the value for registry-type is general.
<b>workflow-clean-after-provisioned</b>	String	Optional	The indication of whether the workflow instance used to provision this instance will be removed after the workflow is completed. Must be true or false. This field is not valid when the value for registry-type is general.
<b>actions</b>	Action[]	Optional	The actions for the software.  Cannot be updated if the registry type is catalog for the software services instance.  See <a href="#">Table 231 on page 327</a> .

Table 230. Request content for an update software services instance request (continued)

Field name	Type	Required or optional	Description
<b>variables</b>	Variable[]	Optional	The variables for the software. Refer to <a href="#">Table 232 on page 328</a> . Cannot be updated if the registry type is catalog for the software services instance.
<b>user-data-id</b>	String	Optional	The user data ID.
<b>user-data</b>	String	Optional	The user data.
<b>ssin</b>	String	Optional	Software service instance name, used in generating names for software services instances. This field is not valid when the value for registry-type is general.

Table 231. Action structure for an "update software services instance" request

Field	Type	Description
<b>name</b>	String	The name of the action. If the name of the action is deprovision, the action is for deprovisioning the software. You can indicate that the action is for deprovisioning either by setting the is-deprovision field to true or by naming the action deprovision.
<b>type</b>	String Must be one of the following values: <ul style="list-style-type: none"> <li>• command</li> <li>• workflow</li> <li>• instructions</li> </ul>	The type of the action.
<b>is-deprovision</b>	String	Indicates if the action deprovisions the software, as follows: <ul style="list-style-type: none"> <li>• If true, the action deprovisions the software.</li> <li>• If false or not set, the action does not deprovision the software. This is overridden if the value of the name field is deprovision.</li> </ul>
<b>description</b>	String	The description of the action. This field is optional. If not provided, the description is empty.
<b>command</b>	String	For command type actions, the command.
<b>command-run-as-user</b>	String	For command type actions, if provided, the user ID to be used when the command is run. This is not valid when the registry-type is general.
<b>command-sol-key</b>	String	For command type actions, if provided, the key to search for in the solicited messages command response.
<b>command-unsol-key</b>	String	For command type actions, if provided, the key to search for in the unsolicited messages.

Table 231. Action structure for an "update software services instance" request (continued)

Field	Type	Description
<b>command-detect-time</b>	String	For command type actions, if provided, the time in seconds to detect for the command-unsol-key in the unsolicited messages.  Also, the minimum time before a command response is checked for after the command is submitted for execution.  If not provided, the default command-detect-time is 15 seconds when the command-unsol-key is specified or 10 seconds when the command-unsol-key is not specified.
<b>workflow-definition-file</b>	String	For workflow type actions, the workflow definition file.
<b>workflow-variable-input-file</b>	String	For workflow type actions, if provided, the workflow variable input file.
<b>variables</b>	Variable[]	For workflow type actions, if provided, the workflow variables. See Table 232 on page 328.
<b>instructions</b>	String	For instruction type actions, the instructions.
<b>workflow-clean-after-complete</b>	String	For workflow type actions, if provided, specifies whether the instance of the workflow is deleted after it completes. The values are true, false, or inherit. If no value is provided, the value is inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance.
<b>prompt-variables</b>	PromptVariable[]	Prompt variables, for workflow type actions, if any are provided. At create time, there are prompts for the values. See Table 233 on page 328.
<b>at-create-variables</b>	String	Names of the at create variables, for workflow type actions, if any are provided. These are the only variables that are allowed on input variables for the do action operation.

Table 232. Variable structure

Field	Type	Description
<b>name</b>	String	Name of the variable.
<b>description</b>	String	Description of the variable.
<b>value</b>	String	Value of the variable.
<b>visibility</b>	String. The value must be public or private.	Visibility of the variable.
<b>update-registry</b>	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

Table 233. Response from a get request: Prompt-Variable-Object

Field	Type	Description
name	String	Name of the property.

Table 233. Response from a get request: Prompt-Variable-Object (continued)

Field	Type	Description
value	String	Current value for the property.
required	boolean	Indicates whether the variable value is required during the workflow create process.
label	String	Short label for the UI widget.
description	String	Explanation of what the variable is used for and perhaps what the syntactic requirements are.
abstract	String	Brief description of the variable for the UI widget.
type	String	Type of the variable element: boolean, string, integer, decimal, time, date.
must-be-choice	boolean	Indicates whether the value must come from the provided choices.
choices	Array of Strings	Contains allowable choices for the value of the variable.
regex	String	Standard regular expression that constrains the variable value.
multi-line	boolean	Indicates whether the value requires a multi-line text box.
min	String	For a string type, indicates the minimum string length of the value. For all other types, indicates the minimum value required.
max	String	For a string type, indicates the maximum string length of the value. For all other types, indicates the maximum value required.
places	String	Maximum number of decimal places that can be specified for a variable of type decimal.
error-message	String	Default error message associated with an incorrect value.

## Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

The user issuing the request must be one of the following:

- Owner of the software services instance
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances
- For catalog registry type objects, a domain administrator of the software services instance
- For general registry type objects, the landlord.

For more information, see [“Authorization requirements” on page 287](#).

## HTTP status codes

On successful completion, HTTP status code 204 Normal is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 234. HTTP error response codes for a update software services instance request

HTTP error status code	Description
<b>HTTP 400 Bad request</b>	Request contained incorrect parameters.
<b>HTTP 401 Unauthorized</b>	The requestor user ID is not authorized for this request.
<b>HTTP 404 Not found</b>	The specified software services instance instance was not found; the software services instance does not exist.
<b>HTTP 409</b>	The field cannot be updated for the registry type.

## Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 235. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the PUT method is used to update a software services instance. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
PUT /zosmf/provisioning/rest/ 1.0/scr/ 76963ea5-81a4-42d6-99d6-f3e19747cf61
```

```
{
  "state": "provisioned"
}
```

Figure 128. Sample request to update a software services instance property

## Update variables in a software services instance

You can use this operation to update variables in a software services instance.

### HTTP method and URI path

```
PUT /zosmf/provisioning/rest/version/scr/object-id/variables
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance for which variables are to be updated.

## Query parameters

None.

## Description

This operation updates variables in a software services instance. If it already exists, the value and visibility are updated based on the values in the variable structure.

## Request content

The request body contents must be a JSON object containing a variables field with the variables to be updated in the object. See [Table 236 on page 331](#).

Table 236. Request content for the update software services instance variables request		
Field name	Type	Description
<b>variables</b>	Variable[]	The variables for the software, with the structure that is described in <a href="#">Table 237 on page 331</a> . The name field identifies the variable. If a variable in the variables array does not already exist in the software object it is added. If it does already exist the name and visibility are updated based on the values in the variable structure.

Table 237. Variable structure		
Field	Type	Description
<b>name</b>	String	Name of the variable.
<b>description</b>	String	Description of the variable.
<b>value</b>	String	Value of the variable.
<b>visibility</b>	String. The value must be public or private.	Visibility of the variable.
<b>update-registry</b>	String. The value must be true or false. The default is false.	Indicates whether to update the variables in the instance from the workflow.

## Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

The user issuing the request must be one of the following:

- Owner of the software services instance
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances
- For catalog registry type objects, a domain administrator of the software services instance
- For general registry type objects, the landlord.

For more information, see [“Authorization requirements” on page 287](#).

## HTTP status codes

On successful completion, HTTP status code 204 Normal is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 238. HTTP error response codes for a update software services instance request	
HTTP error status code	Description
HTTP 400 Bad request	Request contained incorrect parameters.
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance instance was not found.

## Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 239. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the PUT method is used to update variables for a software services instance. The software services instance is uniquely identified by the software services instance key, which is represented by the following string value: 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
PUT /zosmf/provisioning/rest/1.0/scr/76963ea5-81a4-42d6-99d6-f3e19747cf61/variables
```

Variable structure:

```
{
  "variables": [
    { "name": "var1", "value": "val1", "visibility": "public" },
    { "name": "var2", "value": "val2", "visibility": "public" }
  ]
}
```

Figure 129. Sample request to update variables for a software services instance

## Delete a software services instance

The delete operation removes a software services instance from the software services registry.

### HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be deleted.

### Query parameters

None.

### Description

This operation removes a z/OSMF software services instance. The state of a software services instance must be one of the following:

- deprovisioned
- deprovisioning-failed
- provisioning-failed.

Deleting a composite software services instance deletes all of the associated child instances.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

The user issuing the request must be the owner of the software services instance.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 204 *Normal* is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

<i>Table 240. HTTP error response codes for a delete software services instance request.</i>	
deletion of a parent software service instance will delete all the corresponding children instances.	
deletion of a parent software service instance will delete all the corresponding children instances.	
HTTP error status code	Description
<b>HTTP 403 Unauthorized</b>	The requester user ID is not authorized for this request.
<b>HTTP 404 Not found</b>	The specified software services instance was not found because it does not exist.
<b>HTTP 409 Request conflict</b>	The software services instance could not be removed because its state was not either deprovisioned, deprovisioning-failed, or provisioning-failed.

## Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

*Table 241. Response from a request failure*

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the DELETE method is used to delete a software services instance. The software services instance is uniquely identified by a key, which is the string value 76963ea5-81a4-42d6-99d6-f3e19747cf61.

```
DELETE /zosmf/provisioning/rest/ 1.0/scr/76963ea5-81a4-42d6-99d6-f3e19747cf61
```

*Figure 130. Sample request to delete a software services instance*

## Resume a provisioning workflow

You can use this operation to resume a provisioning workflow that is suspended.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/resume-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.

### Query parameters

None.

### Description

This operation resumes a provisioning workflow that is suspended.

On successful completion, HTTP status code 204 (Normal) is returned.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

The user issuing the request must be the owner of the software object, or a domain administrator of the software object.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 242. HTTP error response codes for a resume provisioning workflow request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.
HTTP 409 Conflict	A conflict exists.

### Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 243. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the POST method is used to resume a provisioning workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/resume-workflow
```

Figure 131. Sample request to resume a provisioning workflow for a software services instance

The response is 204. There is no response body.

## Perform an action against a software services instance

You can use this operation to perform an action against a software services instance.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance.
- **<action>** identifies the action to be performed.

### Query parameters

None.

### Description

This operation performs an action against a software services instance.

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 338](#).

**Note:** You cannot deprovision the child instances of a composite instance (that is, an instance created from a composite template). Instead, deprovision against the composite instance.

### Request content

The request content is expected to contain a JSON object. See [Table 244 on page 337](#) for the fields.

Table 244. Request content for the perform action software services instance request			
Field name	Type	Required or optional	Description
<b>input-variables</b>	input Variable[]	Optional	The input variables to be used by workflow-type actions. See <a href="#">Table 245 on page 337</a> .
<b>target-system-nickname</b>	String	Optional	The system nickname indicating the system in the sysplex to run the action on.

Table 245. Input variable structure	
Field	Type
name	String
value	String

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

The user issuing the request must be one of the following:

- Owner of the software services instance

- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances
- For catalog registry type objects, a domain administrator of the software services instance
- For general registry type objects, the landlord.

For more information, see [“Authorization requirements” on page 287](#).

## HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 338](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

<i>Table 246. HTTP error response codes for a do action request</i>	
HTTP error status code	Description
<b>HTTP 400 Error</b>	Bad request.
<b>HTTP 403 Unauthorized</b>	The requestor user ID is not authorized for this request.
<b>HTTP 404 Not found</b>	The specified software services instance was not found because it does not exist.

## Response content

On successful completion, the response body contains a JSON object consisting of the response from the action.

<i>Table 247. Response body for the do action request</i>		
Field name	Type	Description
<b>action-id</b>	String	The ID of the action object that was created by running the action. The action ID is used on further requests to the action object.
<b>action-uri</b>	String	The URI of the new action object.

If a failure occurs, the response body contains a JSON object with a description of the error.

<i>Table 248. Response from a request failure</i>		
Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the POST method is used to perform an action for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/actions/start
```

*Figure 132. Sample request to perform an action against a software services instance variables*

An example of the response is shown in the figures that follow.

```
{
  "action-id": "65963ea5-81a4-42d6-99d6-f3e19748cf61",
  "action-uri":
    "/zosmf/provisioning/rest/1.0/scr/76963ea5-81a4-42d6-99d6-f3e19747cf61/actions/65963ea5-81a4-42d6-99d6-f3e19748cf61"
}
```

*Figure 133. Sample response from a get software services instance variables request*

## Resume an action workflow

You can use this operation to resume an action workflow that is suspended.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/resume-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.
- *<action-id>* identifies the action to be resumed.

### Query parameters

None.

### Description

This operation resumes an action workflow that is suspended.

On successful completion, HTTP status code 204 (Normal) is returned.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

The user issuing the request must be one of the following:

- Owner of the software object
- Domain administrator of the software object
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 249. HTTP error response codes for a resume action workflow request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Table 249. HTTP error response codes for a resume action workflow request (continued)

HTTP error status code	Description
HTTP 409 Conflict	A conflict exists.

## Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 250. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the POST method is used to resume an action workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/actions/f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9/
resume-workflow
```

Figure 134. Sample request to resume an action workflow for a software services instance

The response is 204. There is no response body.

## Retry a provisioning workflow

You can use this operation to restart a failed provisioning workflow.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/retry-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.

### Query parameters

None.

### Description

This operation restarts a provisioning workflow that failed. The workflow instance is restarted at the workflow step that failed.

On successful completion, HTTP status code 204 (Normal) is returned.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

The user issuing the request must be the owner of the software object, or a domain administrator of the software object.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 251. HTTP error response codes for a retry provisioning workflow request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.
HTTP 409 Conflict	A conflict exists.

### Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 252. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

### Example HTTP interaction

In the following example, the POST method is used to retry a provisioning workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/retry-workflow
```

Figure 135. Sample request to retry a provisioning workflow for a software services instance

The response is 204. There is no response body.

## Retry an action workflow

You can use this operation to restart a failed action workflow.

### HTTP method and URI path

```
POST /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>/retry-workflow
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance.
- *<action-id>* identifies the action to be retried.

### Query parameters

None.

### Description

This operation restarts an action workflow that failed. The workflow instance is restarted at the workflow step that failed.

On successful completion, HTTP status code 204 (Normal) is returned.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

The user issuing the request must be one of the following:

- Owner of the software object
- Domain administrator of the software object
- A member of the tenant that the instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions on software instances.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 204 (Normal) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 253. HTTP error response codes for a retry action workflow request	
HTTP error status code	Description
HTTP 401 Unauthorized	The requestor user ID is not authorized for this request.
HTTP 404 Not found	The specified software services instance was not found because it does not exist.

Table 253. HTTP error response codes for a retry action workflow request (continued)

HTTP error status code	Description
HTTP 409 Conflict	A conflict exists.

## Response content

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 254. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the POST method is used to retry an action workflow for a software services instance.

```
POST /zosmf/provisioning/rest/1.0/scr/81963ea5-81a4-42d6-99d6-f3e19747cf61/actions/f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9/
retry-workflow
```

Figure 136. Sample request to retry an action workflow for a software services instance

The response is 204. There is no response body.

## Get the response for an action performed against a software services instance

You can use this operation to retrieve information about the response for an action that was performed against a software services instance.

### HTTP method and URI path

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance to be retrieved.
- *<action-id>* identifies the actions object to be retrieved.

### Query parameters

None.

### Description

This operation retrieves an action object that describes the response for an action that was performed against a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 256 on page 347](#).

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

For catalog registry type objects, the user issuing the request must be at least one of the following:

- The owner of the software services instance
- A member of the tenant of the software services instance, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- A domain administrator of the software services instance.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 256 on page 347](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 255. HTTP error response codes for a get software services instance contents request	
HTTP error status code	Description
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

Table 255. HTTP error response codes for a get software services instance contents request (continued)	
HTTP error status code	Description
HTTP 404 Not found	The specified software services instance was not found; the software services instance does not exist.

## Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 256](#) on [page 347](#) lists the fields in the JSON object.

Table 256. JSON object that is returned for a get actions request		
Field	Type	Description
<b>action-id</b>	String	The action ID for the action object.
<b>name</b>	String	The name for the action.
<b>type</b>	String	Type of the action.
<b>is-deprovision</b>	String must be: <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	If true, the action is a deprovision action. Otherwise, the action is not a deprovision action.
<b>state</b>	String must be: <ul style="list-style-type: none"> <li>• in-progress</li> <li>• submitted</li> <li>• suspended</li> <li>• responded</li> <li>• warning</li> <li>• complete</li> <li>• failed</li> </ul>	The current state of the action. The values submitted, responded, and warning are valid only for command type actions. The suspended value is valid only for workflow type actions.  For the command action state of warning, see the command-response, command-sol-key-hit, and command-detection-status fields. Either no response was received, the command-sol-key-hit is false, or the command-detection-status is expired.
<b>description</b>	String	The description of the action. This field is optional. If not provided, the description is empty.
<b>ran-at-time</b>	String	The time the do action operation was done to create the action, in ISO8601 format
<b>ran-by-user</b>	String	The user ID that ran the do action operation that created the action
<b>instructions</b>	String	The instructions associated with the action, or null if no instructions are associated
<b>command</b>	String	The command associated with the action, or null if no command is associated
<b>command-response</b>	String	The solicited messages response from the command.
<b>command-sol-key-hit</b>	String: null, true, or false	If the command-sol-key was specified, indicates whether the command-sol-key was found in the solicited message response. If the command-sol-key-hit is false then the action state is set to warning.

Table 256. JSON object that is returned for a get actions request (continued)

<b>command-detection-message</b>	String	If the command-unsol-key was specified and it was found in the unsolicited messages from the command, the message containing the command-unsol-key.
<b>command-detection-status</b>	String: null, waiting, expired, or detected	If the command-unsol-key was specified, this is the status of whether the command-unsol-key was found in the unsolicited messages. If the command-detection-status is expired then the action state is set to warning.
<b>workflow-key</b>	String	The workflow key of the workflow associated with the action, or null if no workflow is associated
<b>workflow-current-step-name</b>	String	The current workflow step name of the workflow associated with the action, or null
<b>workflow-message-id</b>	String	The workflow message ID for the workflow associated with the action, or null
<b>workflow-message-text</b>	String	The workflow message text for the workflow associated with the action, or null
<b>workflow-name</b>	String	The workflow name for the workflow associated with the action, or null
<b>workflow-status-name</b>	String	The workflow status name for the workflow associated with the action, or null.
<b>workflow-start-time</b>	String	The time that workflow processing started, in ISO8601 format.  The value is null if the workflow was not started.
<b>workflow-stop-time</b>	String	The time that workflow automation last stopped, in ISO8601 format.  The value is null if the workflow automation has not stopped.
<b>system</b>	String	System that the software is provisioned on.
<b>sysplex</b>	String	Sysplex that the software is provisioned on.
<b>system-nickname</b>	String	The nickname of the system that the software is provisioned on.
<b>composite-parent-action-id</b>	String	The action ID for the composite parent's action that is associated with this action.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 257. Response from a request failure

Field	Type	Description
<b>HttpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.

Table 257. Response from a request failure (continued)

Field	Type	Description
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the GET method is used to retrieve the response for an action that was performed for a software services instance.

```
GET /zosmf/provisioning/rest/1.0/scr/b0d1806f-7d42-4b8d-ad4b-8b8747642cc3/actions/764190f2-350b-4d08-b036-f3bb0a861068
```

Figure 137. Sample request to get software services instance actions

The following is an example of the response.

```
{
  "name": "deprovision",
  "state": "complete",
  "type": "workflow",
  "command": null,
  "instructions": null,
  "action-id": "764190f2-350b-4d08-b036-f3bb0a861068",
  "ran-at-time": "2017-04-17T15:20:07.480Z",
  "ran-by-user": "IBMUSER",
  "is-deprovision": "true",
  "workflow-current-step-name": "",
  "workflow-key": "100f4645-94df-472b-81f2-f8268e2e73e3",
  "workflow-message-id": "IZUWF0162I",
  "workflow-message-text": "IZUWF0162I: Automation processing for workflow \"MIX_DB2000deprovision1492442407521\" is complete.",
  "workflow-name": "MIX_DB2000deprovision1492442407521",
  "workflow-status-name": "complete",
  "command-response": null,
  "command-sol-key-hit": null,
  "command-detection-message": null,
  "command-detection-status": null,
  "workflow-start-time": "2017-04-17T15:20:07.702Z",
  "workflow-stop-time": "2017-04-17T15:20:07.726Z",
  "system": "SYS1",
  "sysplex": "PLEX1",
  "system-nickname": "SY1"
}
```

Figure 138. Sample response for performed actions



## List the responses for actions performed against a software services instance

You can use this operation to list the responses for actions that were performed against a software services instance.

### HTTP method and URI path

---

```
GET /zosmf/provisioning/rest/<version>/scr/<object-id>/actions
```

---

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the provisioning service. The following value is valid: 1.0.
- *<object-id>* identifies the software services instance for which actions are to be retrieved.

### Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

**type**

Optional, specifies the type of the software.

**name**

Optional, regular expression, specifies the name of the action object.

**state**

Optional, specifies the current state of the action:

- in-progress
- submitted
- suspended
- responded
- warning
- complete
- failed.

If you specify no query parameters, all actions are returned.

### Description

This operation lists the action objects for actions that were performed against a software services instance.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 260 on page 352](#).

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: *<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES*.

For catalog registry-type objects, the user issuing the request must be at least one of the following:

- The owner of the software object
- A member of the tenant that the software object is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances

- A domain administrator of the software object.

For more information, see [“Authorization requirements” on page 287](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 260 on page 352](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 258. HTTP error response codes for a "get software services instance contents" request</i>	
HTTP error status code	Description
<b>HTTP 403 Unauthorized</b>	The requester user ID is not authorized for this request.
<b>HTTP 404 Not found</b>	The specified software services instance was not found; the software services instance does not exist.

## Response content

On successful completion, the response body is a JSON object that contains the retrieved data. See [Table 259 on page 352](#) and [Table 260 on page 352](#) lists the fields in the JSON object.

<i>Table 259. JSON object that is returned for a list actions request</i>		
Field	Type	Description
scr-list-actions	Array of objects	Array of action objects.

<i>Table 260. Action object for a list actions request</i>		
Field	Type	Description
action-id	String	The action ID for the action object.
name	String	The name for the action.
type	String	Type of the action.
description	String	The description of the action. This field is optional. If not provided, the description is empty.
state	String. Must be one of the following: <ul style="list-style-type: none"> <li>• in-progress</li> <li>• submitted</li> <li>• suspended</li> <li>• responded</li> <li>• warning</li> <li>• complete</li> <li>• failed.</li> </ul>	The current state of the action. The values submitted, responded, and warning are valid only for command type actions. The suspended value is valid only for workflow type actions.
ran-at-time	String	The time the do action operation was done to create the action, in ISO8601 format
ran-by-user	String	The user ID that ran the do action operation that created the action

Table 260. Action object for a list actions request (continued)

Field	Type	Description
composite-parent-action-id	String	The action ID for the composite parent's action that is associated with this action.
workflow-key	String	Workflow key of the workflow that is associated with the action, or null if no workflow is associated.
workflow-name	String	Workflow name of the workflow that is associated with the action, or null if no workflow is associated.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 261. Response from a request failure

Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the GET method is used to retrieve the list of responses for actions that were performed against a software services instance.

```
GET /zosmf/provisioning/rest/1.0/scr/b0d1806f-7d42-4b8d-ad4b-8b8747642cc3/actions
```

Figure 139. Sample request to list performed actions

The following is an example of the response.

```
{
  "scr-list-actions":
  [
    {
      "name": "Instructions1",
      "state": "complete",
      "type": "instructions",
      "action-id": "f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9",
      "ran-at-time": "2015-10-26T18:29:20.949Z",
      "ran-by-user": "ZOSMFAD"
    },
    {
      "name": "deprovision",
      "state": "complete",
      "type": "workflow",
      "action-id": "ae3ec9cc-9be3-42b4-98f5-aa64934e31a3",
      "ran-at-time": "2015-10-27T14:34:27.186Z",
      "ran-by-user": "ZOSMFAD"
    }
  ]
}
```

Figure 140. Sample response from a list actions request

## Delete the response for an action performed against a software services instance

The delete operation removes the response for an action that was performed against a software services instance.

### HTTP method and URI path

```
DELETE /zosmf/provisioning/rest/<version>/scr/<object-id>/actions/<action-id>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the provisioning service. The following value is valid: 1.0.
- **<object-id>** identifies the software services instance for which an action response is to be deleted.
- **<action-id>** identifies the action for which the response is to be deleted.

### Query parameters

None.

### Description

This operation removes the response for an action that was performed against a software services instance.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: **<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES**.

The user issuing the request must be one of the following:

- Owner of the software services instance
- For catalog registry type objects, one of the following: a domain administrator of the software services instance or a member of the tenant that the software services instance is associated with, if the option has been set in the resource pool, through the Resource Management task of z/OSMF, to allow members of the tenant to access and run actions for software instances
- For general registry type objects, the landlord.

For more information, see [“Authorization requirements” on page 287](#).

### HTTP status codes

On successful completion, HTTP status code 204 **Normal** is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 262. HTTP error response codes for a delete action response request	
HTTP error status code	Description
<b>HTTP 403 Unauthorized</b>	The requester user ID is not authorized for this request.
<b>HTTP 404 Not found</b>	The specified software services instance was not found because it does not exist.

## Response content

On successful completion, the response body contains nothing.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 263. Response from a request failure		
Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In the following example, the DELETE method is used to delete the response for an action that was performed against a software services instance.

```
DELETE /zosmf/provisioning/rest/1.0/scr/b0d1806f-7d42-4b8d-ad4b-8b8747642cc3/actions/f5c4df98-f9fd-4fca-b1a5-e0d1b7d1f0d9
```

*Figure 141. Sample request to delete a response for a performed action*

## Software service instance name services

The software service instance name (SSIN) services are application programming interfaces (APIs), which are implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage software service instance names.

For information about cloud provisioning, including a description of the roles, see [“Cloud provisioning services”](#) on page 45.

Table 264 on page 356 lists the operations that the SSIN services provide.

Table 264. SSIN services: operations summary	
Operation name	HTTP method and URI path
<a href="#">“Create software service instance names”</a> on page 357	POST    /zosmf/resource-mgmt/rest/<version>/ssin
<a href="#">“List the software service instance names”</a> on page 360	GET    /zosmf/resource-mgmt/rest/<version>/ssin
<a href="#">“Create a variable name”</a> on page 362	POST    /zosmf/resource-mgmt/rest/<version>/ssin/variable-name
<a href="#">“Create unique variable names”</a> on page 364	POST    /zosmf/resource-mgmt/rest/<version>/unique-variable-names

### Authorization requirements

Use of the SSIN services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF”](#) on page 2.

The user’s z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE\_MANAGEMENT.

### Error response content

For the 4nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 265. Response from a request failure		
Field	Type	Description
httpStatus	Integer	HTTP status code.
requestMethod	String	HTTP request method.
requestUri	String	HTTP request URI.
messageID	String	Message identifier for the error.
messageText	String	Message text describing the error.
additionalInfo	String	Additional information describing the error.
debug	String	Debug information about for the error.

## Error logging

Errors from the software services instance services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 Normal

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 201 Created

The request succeeded and resulted in the creation of an object.

### HTTP 400 Bad request

The request contained incorrect parameters.

### HTTP 403 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

## Create software service instance names

You can use this operation to create software service instance names (SSINs).

## HTTP method and URI path

---

```
POST /zosmf/resource-mgmt/rest/<version>/ssin
```

---

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

## Query parameters

None.

## Description

This operation creates SSINs. It uses the name-prefix in the resource definition profile as a basis for creating the names. An initial SSIN is created when a software instance is provisioned. A maximum of 8 generated SSINs may exist in the resource definition profile that the software instance is using. Allocation of SSINs for a provisioned software instance are released when the software instance is deprovisioned. The name-prefix in the resource definition profile must end with the special wildcard character, `*`.

For the properties that you can specify in the request body, a JSON object, see [“Request content” on page 357](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a SSINs.

## Request content

The request content is expected to contain a JSON object. [Table 266 on page 358](#) lists the fields in the JSON object.

Table 266. Request content for the create SSIN request			
Field name	Type	Required or optional	Description
<b>template-id</b>	String	Required	The ID of the template.
<b>domain-id</b>	String	Required	The ID of the domain.
<b>tenant-id</b>	String	Required	The ID of the tenant.
<b>registry-id</b>	String	Required	The ID of the software instance registry.
<b>quantity</b>	String	Required	The number of names to be generated. The value must be 1-7.

## Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE\_SERVICES.

For more information, see [“Authorization requirements” on page 356](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 358](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 267. HTTP error response codes for a create SSIN request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained incorrect parameters.
<b>HTTP 403 Unauthorized</b>	The requester user ID is not authorized for this request.

## Response content

On successful completion, the service returns a JSON object named ssin-list consisting of the names that were created. See [Table 268 on page 358](#).

Table 268. Response from a create SSIN request		
Field	Type	Description
<b>ssin-list</b>	Array	Software service instance names. See <a href="#">Table 269 on page 358</a> .

Table 269. Fields in the ssin-list array		
Field	Type	Description
<b>ssin</b>	String	Software service instance name.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 270. Response from a request failure		
Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.

Table 270. Response from a request failure (continued)

Field	Type	Description
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In Figure 142 on page 359, a request is submitted to create 2 SSINs.

```
POST /zosmf/resource-mgmt/rest/1.0/ssin
{
  "domain-id": "izu$0",
  "registry-id": "046c3cb2-7ef2-40a0-8b10-a34d8a23e5fc",
  "template-id": "9eb7df8a-284c-4550-a92e-8150bc6fe68f",
  "tenant-id": "izu$000",
  "quantity": "2"
}
```

Figure 142. Sample request to create SSINs

The response body is as follows.

```
{
  "ssin-list": [
    {
      "ssin": "INAME101"
    },
    {
      "ssin": "INAME201"
    }
  ]
}
```

## List the software service instance names

You can use this operation to list the software service instance names (SSINs).

### HTTP method and URI path

---

```
GET /zosmf/resource-mgmt/rest/<version>/ssin
```

---

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

### Query parameters

You can specify the following query parameter on this request. Objects matching all query parameters are returned.

**name**

Name of the object for which SSINs should be obtained.

**registry-id**

Identifier of the registry for which SSINs should be obtained.

If you specify no query parameters, then all SSINs are returned.

### Description

The list operation returns software service instance names based on the input query.

On successful completion, HTTP status code 200 (Normal) is returned, along with a response body.

### Request content

None.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.RESOURCE_MANAGEMENT`.

See [“Authorization requirements” on page 356](#).

### HTTP status codes

On successful completion, HTTP status code 200 (Normal) is returned and the response body is provided, as described in [“Response content” on page 361](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 271. HTTP error response codes for a list SSIN request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained incorrect parameters.
HTTP 403 Unauthorized	The requester user ID is not authorized for this request.

## Response content

On successful completion, the service returns a JSON object named `ssin-list` consisting of the names that were created. See [Table 272 on page 361](#).

*Table 272. Response from a list SSINs request*

Field	Type	Description
<b>ssin-list</b>	Array	Software service instance names. See <a href="#">Table 273 on page 361</a> .

*Table 273. Fields in the ssin-list array*

Field	Type	Description
<b>ssin</b>	String	Software service instance name.
<b>provisioning-version</b>	String	Identifies the provisioning version of the persistent data object for the entry.
<b>provisioning-version-supported</b>	boolean	Indicates if Get, Post, Put, and Delete operations are allowed for the persistent data object for the entry: <ul style="list-style-type: none"><li>• true if the operations are allowed</li><li>• false if the operations are not allowed.</li></ul>

If a failure occurs, the response body contains a JSON object with a description of the error.

*Table 274. Response from a request failure*

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

In [Figure 143 on page 361](#), a request is submitted to list the SSINs for `name=INAME.*`.

```
GET /zosmf/resource-mgmt/rest/1.0/ssin?name=INAME.*
```

*Figure 143. Sample request to list SSINs*

The response body is as follows.

```
{
  "ssin-list": [
    {
      "ssin": "INAME101",
      "provisioning-version": "1400",
      "provisioning-version-supported": true
    },
    {
      "ssin": "INAME201",
```

```
    "provisioning-version": "1400",  
    "provisioning-version-supported": true  
  }  
]
```

## Create a variable name

You can use this operation to create a variable name.

### HTTP method and URI path

Create software service instance names

---

```
POST /zosmf/resource-mgmt/rest/<version>/ssin/variable-name
```

---

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation creates a variable name based on the input variable prefix and the last 2 digits from the SSIN for the input registry ID. For the properties that you can specify, see [“Request content” on page 362](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new variable name. A response body is provided, as described in [“Response content” on page 363](#).

### Request content

The request content contains a JSON object. [Table 275 on page 362](#) lists the fields in the JSON object.

Table 275. Request content for the create variable name request			
Field name	Type	Required or optional	Description
variable-prefix	String	Required	The prefix to use to create the variable name.
registry-id	String	Required	The ID of the software instance registry entry.

### Authorization requirements

The user's z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: `<SAF-prefix>.ZOSMF.PROVISIONING.SOFTWARE_SERVICES`.

See [“Authorization requirements” on page 356](#).

### HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 363](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 276. HTTP error response codes for a create variable request

HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained incorrect parameters.
<b>HTTP 403 Unauthorized</b>	The requester user ID is not authorized for this request.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the variable name. [Table 277 on page 363](#) lists the fields in the JSON object.

Table 277. Response from a create variable name request

Field	Type	Description
<b>name</b>	String	Variable name.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 278. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

Figure 144 on page 363 shows a request to create a variable name.

```
POST /zosmf/resource-mgmt/rest/1.0/ssin/variable-name
{
  "variable-prefix": "VAR",
  "registry-id": "3196202f-9a6c-4fdf-8dcd-e307e3ce2d5b"
}
```

Figure 144. Sample request to create a variable name

The response body is as follows.

```
{
  "name": "VAR00"
}
```

## Create unique variable names

You can use this operation to create multiple unique variable names.

### HTTP method and URI path

```
POST /zosmf/resource-mgmt/rest/<version>/unique-variable-names
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF software service instance name service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation creates up to 50 unique variable names. For the properties that you can specify, see [“Request content” on page 364](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of unique variable names. A response body is provided, as described in [“Response content” on page 365](#).

### Request content

The request content contains a JSON object. [Table 279 on page 364](#) lists the fields in the JSON object.

Table 279. Request content for the create unique variable names request			
Field name	Type	Required or optional	Description
<b>prefix</b>	String	Optional	The prefix to be used when creating the variable names. If a prefix is not specified on the request, one is supplied by the service.
<b>quantity</b>	String	Required	The number of names to be generated, 1-50.

### Authorization requirements

None.

For more information, see [“Authorization requirements” on page 356](#).

### HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 365](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 280. HTTP error response codes for a create variable request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained incorrect parameters.

Table 280. HTTP error response codes for a create variable request (continued)

HTTP error status code	Description
<b>HTTP 403 Unauthorized</b>	The requester user ID is not authorized for this request.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the variable names. [Table 281 on page 365](#) lists the fields in the JSON object.

Table 281. Response from a create unique variable names request

Field	Type	Description
<b>name-list</b>	String Array	The variable names that were created.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 282. Response from a request failure

Field	Type	Description
<b>httpStatus</b>	Integer	HTTP status code.
<b>requestMethod</b>	String	HTTP request method.
<b>requestUri</b>	String	HTTP request URI.
<b>messageID</b>	String	Message identifier for the error.
<b>messageText</b>	String	Message text describing the error.
<b>additionalInfo</b>	String	Additional information describing the error.
<b>debug</b>	String	Debug information about for the error.

## Example HTTP interaction

[Figure 145 on page 365](#) shows a request to create 5 unique variable names.

```
POST /zosmf/resource-mgmt/rest/1.0/unique-variable-name
{
  "prefix": "VAR",
  "quantity": "5"
}
```

Figure 145. Sample request to create a variable name

The response body is as follows.

```
{
  "name-list": [
    "VAR1462458322735",
    "VAR1462458322737",
    "VAR1462458322739",
    "VAR1462458322741",
    "VAR1462458322743"
  ]
}
```

## Data persistence services

The data persistence services is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with user-specific data and global application data, as described in this topic.

Table 283 on page 366 lists the operations that the data persistence services provide.

Table 283. Operations provided through the data persistence services	
Operation	HTTP method and URI path
<b><u>“Persist user or application data” on page 367</u></b>	PUT /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>  PUT /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
<b><u>“Retrieve persisted user or application data” on page 369</u></b>	GET /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>  GET /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
<b><u>“Delete persisted user or application data” on page 372</u></b>	DELETE /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>  DELETE /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>

### Required authorizations

The user must be logged in to z/OSMF, and must have READ access to the SAF profile that was registered for the plug-in and task making the request.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

### Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. The following JSON object is used by all data persistence services as input and output for the requested operations. The attributes provided in the JSON object depend on the requested operation.

```
{
  "value": "data-value",
  "version": "structure-version",
  "messages": "z/OSMF-messages",
  "update": true|false
}
```

where:

#### **data-value**

The value that will be added, updated, retrieved, or removed by the data persistence services. Any data type is supported including JSON objects, JSON arrays, and scalars. The value is required.

#### **structure-version**

Version of the data persistence services and the JSON object structure used for this request. The version sequence starts at 1.0.0, and is incremented only if the services or the JSON structure changes. The version the client supports is required as input to the request. The data persistence services is backward compatible for *n*-2 versions, and accepts requests for each version it supports. If the version specified by the client is not supported or if no version is specified, the service returns an error message.

## z/OSMF-messages

z/OSMF messages received during the request. The *messages* attribute is included in the JSON object only if an error occurred during the request. The message ID and message text are provided for each z/OSMF message received.

## update

An optional input attribute, which indicates that the service is updating or replacing an existing JSON object. If you set the value to *true*, the service updates the key-value pairs you specified for the *value* attribute and preserves any other data persisted in the JSON object. You can set this attribute to *true* only when the data type is a JSON object or JSON array. If you omit this attribute or set it to *false*, the service deletes the existing JSON object and creates a new JSON object that contains only the key-value pairs you specified for the *value* attribute.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

### HTTP 200 OK

Success.

### HTTP 400® Bad request

Request contained incorrect parameters.

### HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the data persistence services.

### HTTP 404 Bad URL

Target of the request (a URL) was not found.

### HTTP 500 Internal server error

Programming error.

## Error logging

Errors from the data persistence services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Persist user or application data

You can use this operation to persist data to be used by a specific user or application.

## HTTP method and URI path

---

```
PUT /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>
PUT /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>
?saif=<saifparamValue>
```

---

where:

- **zosmf/IzuUICommon/persistence** identifies the data persistence services.
- **user** indicates that the service will persist the data only for the user who is logged into z/OSMF when the service is invoked.
- **app** indicates that the service will persist the data globally for the application.
- **<pluginId>** is the unique identifier you assigned to the plug-in.

- **<taskId>** is the unique identifier you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

For example, to persist data for the *history* attribute shown in the sample JSON object in [Figure 146](#) on page 368, specify the following resource path: /SETTINGS/history/.

```
{
  "private": {
    "created": "2013-07-09T02:52:47.921Z",
    "majorv": 0,
    "minorv": 0,
    "modified": "2014-01-13T15:01:39.409Z"
  },
  "public": {
    "SETTINGS": {
      "authorization": {
        "auth": true
      },
      "history": {
        "acct": [
          "OMVS0803"
        ],
        "proc": [
          "CEANNKJ"
        ],
        "rsize": [
          "50000"
        ],
        "ugrp": [
          "ZOSMFGRP"
        ]
      },
      "trace": {
        "init": false,
        "task": false
      }
    }
  }
}
```

*Figure 146. Sample JSON structure for persisted data*

## Query parameters

### *saf-parameter*

The SAF resource that is defined in the properties file when loading the external plug-in.

## Standard headers

Use the following standard HTTP headers with this request:

```
Accept: application/json
Content-Type: application/json
```

## Custom headers

None.

## Request content

Your request must include a JSON object that contains the value to be persisted and the version. For more details, see [“Content type used for HTTP request and response data”](#) on page 366.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Required authorizations” on page 366](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 484](#).

The response also includes a JSON object that contains the current data after being modified. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

To persist data that satisfies the following criteria, submit the request depicted in [Figure 147 on page 369](#):

- The data is for a task with the ID *MYTASK* that resides in plug-in *com.ibm.zosmf.myapp*.
- The data is being persisted for the user who is currently logged into z/OSMF.
- The JSON object that contains the persistence data uses the structure provided in [Figure 146 on page 368](#).
- The data to be persisted is updating the *rsize* attribute.

```
PUT /zosmf/IzuUICommon/persistence/user/com.ibm.zosmf.myapp/MYTASK/SETTINGS/history/ HTTP/1.1
Host: zosmf1.yourco.com
Accept: application/json
Content-Type: application/json

{
  "version" : "1.0.0",
  "value" : {"history":{"rsize":["40000"]}},
  "update" : true
}
```

*Figure 147. Sample request to persist user-specific data*

A sample response is shown in [Figure 148 on page 369](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Content-Type: application/json

{
  "version" : "1.0.0",
  "value" : {
    "authorization":{"auth":true},
    "history":{"ugrp":["ZOSMFGRP"],"acct":["OMVS0803"],"rsize":["40000"],"proc":["CEANNKJ"]},
    "trace":{"init":false,"task":false}
  }
}
```

*Figure 148. Sample response from a request to persist user-specific data*

## Retrieve persisted user or application data

You can use this operation to retrieve data that is persisted for a specific user or application.

### HTTP method and URI path

```
GET /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
GET /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
```

where:

- **zosmf/IzuUICommon/persistence** identifies the data persistence services.
- **user** indicates that the service will retrieve the data that has been persisted for the user who is logged into z/OSMF when the service is invoked.
- **app** indicates that the service will retrieve the data that has been persisted globally for the application.
- **<pluginId>** is the unique identifier you assigned to the plug-in.
- **<taskId>** is the unique identifier you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the persisted data. The persisted data is stored in a JSON object using a tree structure. To retrieve persisted data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

For example, to retrieve the data persisted for the *history* attribute shown in the sample JSON object in Figure 149 on page 370, specify the following resource path: `/SETTINGS/history/`. In which case, the value for the *acct*, *proc*, *rsize*, and *ugrp* attributes will be retrieved. To retrieve the value for only the *rsize* attribute, specify the following resource path: `/SETTINGS/history/rsize/`.

```
{
  "private": {
    "created": "2013-07-09T02:52:47.921Z",
    "majorv": 0,
    "minorv": 0,
    "modified": "2014-01-13T15:01:39.409Z"
  },
  "public": {
    "SETTINGS": {
      "authorization": {
        "auth": true
      },
      "history": {
        "acct": [
          "OMVS00803"
        ],
        "proc": [
          "CEANNKJ"
        ],
        "rsize": [
          "50000"
        ],
        "ugrp": [
          "ZOSMFGRP"
        ]
      },
      "trace": {
        "init": false,
        "task": false
      }
    }
  }
}
```

Figure 149. Sample JSON structure for persisted data

## Query parameters

### *saf-parameter*

The SAF resource that is defined in the properties file when loading the external plug-in.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 366](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 484](#).

The response also includes a JSON object that contains the retrieved data. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

To retrieve the persisted data that satisfies the following criteria, submit the request depicted in [Figure 150 on page 371](#):

- The data was persisted for a task with the ID *MYTASK* that resides in plug-in *com.ibm.zosmf.myapp*.
- The data was persisted for the user who is currently logged into z/OSMF.
- The JSON object that contains the data uses the structure provided in [Figure 149 on page 370](#).
- The data persisted for the *SETTINGS* attribute is to be retrieved.

```
GET /zosmf/IzuUICommon/persistence/user/com.ibm.zosmf.myapp/MYTASK/SETTINGS?saf=ZOSMF.IBMMYAPP.MYTASK HTTP/1.1 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 150. Sample request to retrieve persisted data*

A sample response is shown in [Figure 151 on page 371](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2014 05:39:28 +0000GMT
Connection: close

{"value":{"authorization":{"auth":true},
  "history":{"ugrp":["ZOSMFGRP"],"acct":["OMVS0803"],"rsize":["50000"],"proc":["CEANNKJ"]},
  "trace":{"init":false,"task":false}
},
  "version":"1.0.0"
}
```

*Figure 151. Sample response from a request to retrieve persisted data*

## Delete persisted user or application data

You can use this operation to remove data that is persisted for a specific user or application.

### HTTP method and URI path

```
DELETE /zosmf/IzuUICommon/persistence/user/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
DELETE /zosmf/IzuUICommon/persistence/app/<pluginId>/<taskId>/<resourcePath>?saf=<safparmValue>
```

where:

- **zosmf/IzuUICommon/persistence** identifies the data persistence services.
- **user** indicates that the service will delete data that has been persisted for the user who is logged into z/OSMF when the service is invoked.
- **app** indicates that the service will delete data that has been persisted globally for the application.
- **<pluginId>** is the unique identifier you assigned to the plug-in.
- **<taskId>** is the unique identifier you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the data to be deleted. The persisted data is stored in a JSON object using a tree structure. To delete persisted data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

For example, to delete the data persisted for the *history* attribute shown in the sample JSON object in [Figure 152](#) on page 372, specify the following resource path: `/SETTINGS/history/`. In which case, the value for the *acct*, *proc*, *rsize*, and *ugrp* attributes will be deleted. To delete the value for only the *rsize* attribute, specify the following resource path: `/SETTINGS/history/rsize/`.

```
{
  "private": {
    "created": "2013-07-09T02:52:47.921Z",
    "majorv": 0,
    "minorv": 0,
    "modified": "2014-01-13T15:01:39.409Z"
  },
  "public": {
    "SETTINGS": {
      "authorization": {
        "auth": true
      },
      "history": {
        "acct": [
          "OMVS0803"
        ],
        "proc": [
          "CEANNKJ"
        ],
        "rsize": [
          "50000"
        ],
        "ugrp": [
          "ZOSMFGRP"
        ]
      },
      "trace": {
        "init": false,
        "task": false
      }
    }
  }
}
```

Figure 152. Sample JSON structure for persisted data

## Query parameters

### *saf-parameter*

The SAF resource that is defined in the properties file when loading the external plug-in.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 366](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error has occurred. For more details, see [“Error handling” on page 484](#).

The response also includes the updated JSON object. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

To delete the persisted data that satisfies the following criteria, submit the request depicted in [Figure 153 on page 373](#):

- The data was persisted for a task with the ID *MYTASK* that resides in plug-in *com.ibm.zosmf.myapp*.
- The data was persisted for the user who is currently logged into z/OSMF.
- The JSON object that contains the data uses the structure provided in [Figure 152 on page 372](#).
- The data persisted for the *history* attribute is to be deleted.

```
DELETE /zosmf/IzuUICommon/persistence/user/com.ibm.zosmf.myapp/MYTASK/SETTINGS/history?  
saf=ZOSMF.IBMMYAPP.MYTASK  
HTTP/1.1 HTTP/1.1  
Host: zosmf1.yourco.com
```

*Figure 153. Sample request to delete persisted data*

A sample response is shown in [Figure 154 on page 374](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{
  "version" : "1.0.0",
  "value":{
    "authorization":{"auth":true},
    "history":null,
    "trace":{"init":false,"task":false}
  }
}
```

Figure 154. Sample response from a request to delete persisted data

## Multisystem routing services

To communicate with and transfer data between systems within your enterprise, z/OSMF uses z/OSMF-to-z/OSMF communication. That is, a z/OSMF instance communicates with other z/OSMF instances to collect information from or about the systems in your enterprise. To enable this capability, each system in your enterprise must be accessible by a z/OSMF instance. Typically, this requires deploying one z/OSMF instance in each monoplex or sysplex in your enterprise.

Although your enterprise can have multiple active z/OSMF instances, it is recommended that you make one instance the primary. The *primary z/OSMF instance* is the instance that:

- Is the base for configuring the other z/OSMF instances in your enterprise.
- Generates the Lightweight Third Party Authentication (LTPA) key that is used for single sign-on (if single sign-on is enabled).
- Is used to perform all z/OS system management tasks in your enterprise, which ensures that the data for each z/OSMF task is centrally managed.
- Acts as the client for all hypertext transfer protocol (HTTP) requests and drives the transfer of files between z/OSMF instances.

You can select any z/OSMF instance that is at least z/OSMF V2R1 with APAR PI32148 to be the primary instance. The remaining z/OSMF instances are referred to as *secondary z/OSMF instances*.

### Example

For example, suppose your installation is configured similar to the installation depicted in [Figure 155 on page 375](#). The installation contains three sysplexes with a total of nine running systems. A z/OSMF instance is active in each sysplex, and your web browser is connected to the z/OSMF instance that is running on System 3 in Sysplex A. Thus, this z/OSMF instance is the primary instance and the z/OSMF instance running on System 6 in Sysplex B and System 9 in Sysplex C are the secondary instances.

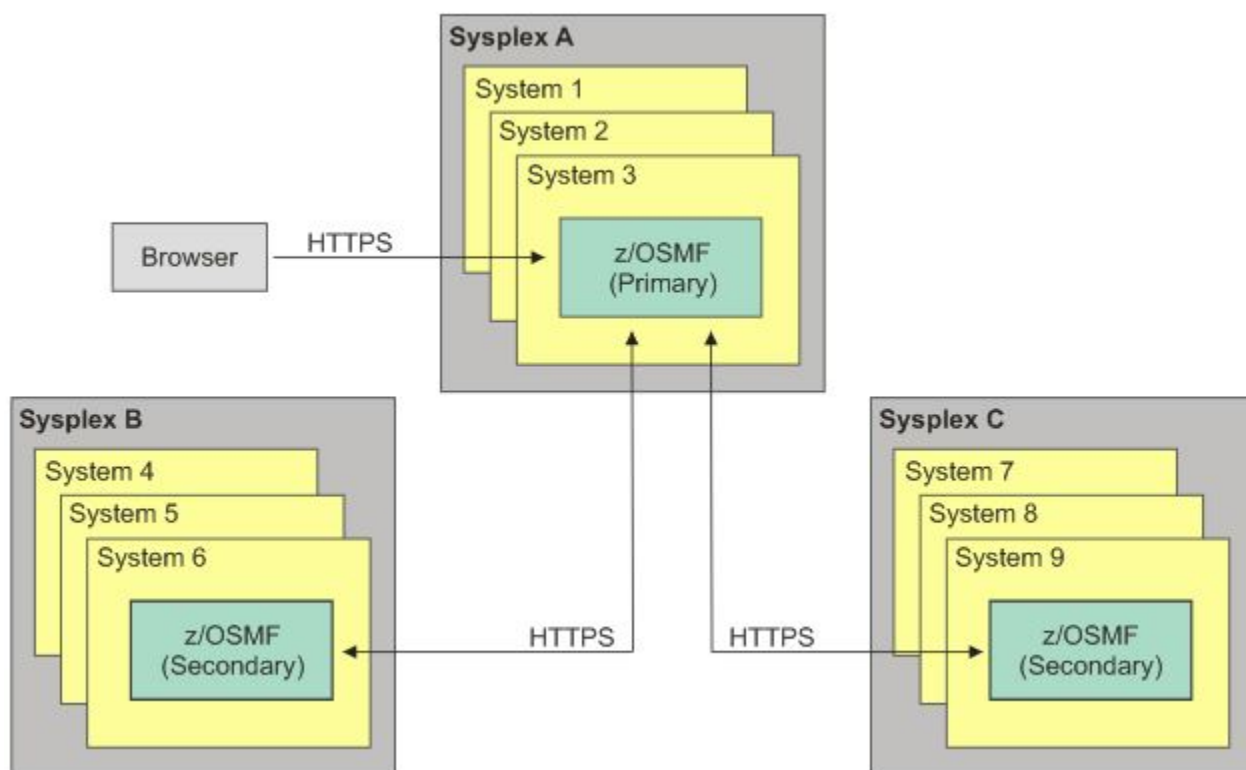


Figure 155. Example sysplex and system configuration

To obtain information from System 6 in Sysplex B, the following actions are performed:

1. A client application sends an HTTPS request to the primary z/OSMF instance.
2. The primary z/OSMF instance routes the HTTPS request to the secondary z/OSMF instance in Sysplex B.
3. The secondary z/OSMF instance processes the request and sends an HTTPS response to the primary z/OSMF instance.
4. The primary z/OSMF instance returns the HTTPS response to the browser.
5. The client application parses the response and extracts the appropriate information.

To route the request, the primary z/OSMF instance needs a system definition that specifies how to access the z/OSMF instance that is running on System 6 in Sysplex B and an HTTP proxy definition that specifies how to navigate the HTTP proxy server that is between the primary and secondary z/OSMF instances.

You can use the z/OSMF Systems task to add or modify the system and HTTP proxy definitions, and you can use the multisystem routing services to route the HTTPS request to the secondary z/OSMF instance and receive the HTTPS response. The remainder of this section describes the multisystem routing services. For information about the Systems task, see the z/OSMF online help.

## Multisystem routing services overview

The multisystem routing services is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for routing HTTPS requests to multiple systems in your enterprise. The multisystem routing services can route any HTTPS request that is supported by the z/OSMF REST services described in Chapter 1, “Using the z/OSMF REST services,” on page 1.

Table 284 on page 376 lists the operations that the multisystem routing services provide.

Table 284. Operations provided through the multisystem routing services.

Operation	HTTP method and URI path
<b><u><a href="#">“Retrieve data from one or more systems” on page 379</a></u></b>	GET /zosmf/gateway/system?content=<http-content> GET /zosmf/gateway/systems?content=<http-content> GET /zosmf/gateway/group?content=<http-content> GET /zosmf/gateway/sysplex?content=<http-content> GET /zosmf/gateway/cpc?content=<http-content>
<b><u><a href="#">“Update data for one or more systems” on page 386</a></u></b>	POST /zosmf/gateway/system POST /zosmf/gateway/systems POST /zosmf/gateway/group POST /zosmf/gateway/sysplex POST /zosmf/gateway/cpc PUT /zosmf/gateway/system PUT /zosmf/gateway/systems PUT /zosmf/gateway/group PUT /zosmf/gateway/sysplex PUT /zosmf/gateway/cpc
<b><u><a href="#">“Delete data from one or more systems” on page 392</a></u></b>	DELETE /zosmf/gateway/system?content=<http-content> DELETE /zosmf/gateway/systems?content=<http-content> DELETE /zosmf/gateway/group?content=<http-content> DELETE /zosmf/gateway/sysplex?content=<http-content> DELETE /zosmf/gateway/cpc?content=<http-content>
<b><u><a href="#">“Authenticate with a secondary z/OSMF instance” on page 398</a></u></b>	POST /zosmf/gateway/logon
<b><u><a href="#">“Authenticate with an HTTP proxy server” on page 400</a></u></b>	POST /zosmf/gateway/logon/proxy

## Required authorizations

The user must be logged into z/OSMF. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for HTTP response data. The following JSON object is used by all multisystem routing services for returning data and status about the requested operations. The attributes provided in the JSON object depend on the requested operation.

```
{
  "primaryAPIVersion": "primary-API-version",
  "systemsOutput": {
    "systemOutput": "system-output",
    "rc": "return-code",
    "error": { "msgid": "message-ID", "msgtxt": "message-text" },
    "secondaryApiVersion": "secondary-API-version",
    "systemVersion": "
    {
      "zosNode": "zos-node",
      "zosVrm": "zos-level",
    }
  }
}
```

```

        "zosSysplex": "sysplex-name"
      },
      "systemName": "system-name"
    },
    "numOfSystems": "total-systems"
  }

```

where:

### **primary-API-version**

Version of the multisystem routing services interface for the primary z/OSMF instance.

### **systemsOutput**

Contains a separate response for each system to which the HTTPS request was sent. If the request was sent to multiple systems, the systemsOutput attribute contains an array of system responses.

### **system-output**

Contains the response returned for a single system. A separate systemOutput attribute is included for each system to which the HTTPS request was sent.

### **return-code**

Code returned by the system. The return code can be one of the following values:

#### **OK**

Success.

#### **HttpConnectionFailed**

The HTTPS connection failed. Typically, this error occurs when the system hosting the secondary z/OSMF instance is unavailable, the z/OSMF instance is not running, or a network error has occurred.

#### **HttpConnectionTimedOut**

The HTTPS request did not complete in the time allotted.

#### **CertificateError**

The certificate for the secondary z/OSMF instance is not trusted.

#### **LoginRequired**

The Lightweight Third Party Authentication (LTPA) token is not valid or has expired. You must submit a separate HTTPS request to authenticate with the z/OSMF instance.

#### **InvalidLogin**

The login credentials for the z/OSMF instance are not valid.

#### **ProxyLoginRequired**

Authentication is required by the proxy server.

#### **InvalidProxyLogin**

The login credentials for the proxy server are not valid.

#### **FailedWithMessage**

The request was successful; however, an internal error occurred with the secondary z/OSMF instance.

#### **UnexpectedFailure**

An unexpected error occurred.

### **error**

If an error occurred with the request, the error attribute contains the message ID (msgid) and message text (msgtxt) for the message that was issued. Otherwise, this attribute is *null*.

### **secondary-API-version**

Version of the multisystem routing services interface for the secondary z/OSMF instance.

### **systemVersion**

Provides additional information about the system, as follows:

#### **zosNode**

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

**zosVrm**

Version, release, and modification level of the z/OS image installed on the system. The level has the format *vv.rr.mm*, where *vv* is the version, *rr* is the release, and *mm* is the modification level. You can correlate the returned value as follows:

- 04.24.00 indicates z/OS V2R1

**zosSysplex**

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

**system-name**

Unique name assigned to the system definition.

**total-systems**

Number of systems to which an HTTPS request was sent.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

**HTTP 200 OK**

Success.

**HTTP 400 Bad request**

Request contained incorrect parameters.

**HTTP 401 Unauthorized**

Submitter of the request did not authenticate with the primary or secondary z/OSMF instance, or is not authorized to use the z/OSMF REST service.

If the user ID required to authenticate with the primary and secondary z/OSMF instances are not the same, submit a separate HTTPS request to authenticate with the secondary z/OSMF instances.

**HTTP 404 Bad URL**

Target of the request (a URL) was not found.

**HTTP 500 Internal server error**

Programming error.

## Error logging

Errors from the multisystem routing services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Retrieve data from one or more systems

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to retrieve data from one system, from a list of systems, or from all the systems in a group, sysplex, or central processor complex (CPC).

### HTTP method and URI path

```
GET /zosmf/gateway/system?content=<http-content>
GET /zosmf/gateway/systems?content=<http-content>
GET /zosmf/gateway/group?content=<http-content>
GET /zosmf/gateway/sysplex?content=<http-content>
GET /zosmf/gateway/cpc?content=<http-content>
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **system** informs the service that the request will be routed to only one system.
- **systems** informs the service that the request will be routed to a list of systems.
- **group** informs the service that the request will be routed to all of the systems in a group.
- **sysplex** informs the service that the request will be routed to all of the systems in a sysplex.
- **cpc** informs the service that the request will be routed to all of the systems in a CPC.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 285 on page 379](#) lists the parameters that are supported for this request.

**Important:** If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *System#1*, specify *System%231*.

Table 285. Supported input parameters for the multisystem routing services

Parameter	Required	Description
<b>target</b>	Yes	If the request is being sent to a system or a list of systems, the target is the nickname of the system. If the request is being sent to all the systems in a group, sysplex, or CPC, the target is the name of the group, sysplex, or CPC. The specified target must be defined in the Systems task. Otherwise, the request will fail.
<b>resourcePath</b>	Yes	<p>Path to the z/OSMF REST service that will process the request. The resource path must be within the z/OSMF context. For example, to ping a TSO/E address space on the target system, you would use the TSO/E address space services to process the request. Therefore, you would specify the following resourcePath: <code>/tsoApp/ping/&lt;servletKey&gt;</code>, where <code>&lt;servletKey&gt;</code> identifies the TSO/E address space for the service to ping.</p> <p>When sending an HTTPS request to a list of systems, you can specify a different resource path and different parameters for each system included in the list. When sending an HTTPS request to all the systems in a group, sysplex, or CPC, you can specify only one resource path and one set of parameters, which will be used for all the systems in the specified group, sysplex, or CPC.</p>

Table 285. Supported input parameters for the multisystem routing services (continued)		
Parameter	Required	Description
<b>requestProperties</b>	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
<b>timeout</b>	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
<b>content</b>	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the z/OSMF REST interface that will process the request.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 376](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 378](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 376](#).

## Examples

To obtain sample HTTPS requests and responses for retrieving data from one system, from a list of systems, or from all the systems in a group, sysplex, or CPC, see the following sections:

- [“Example 1: Retrieve data from one system” on page 381](#)
- [“Example 2: Retrieve data from a list of systems” on page 381](#)
- [“Example 3: Retrieve data from all the systems in a group” on page 382](#)
- [“Example 4: Retrieve data from all the systems in a sysplex” on page 383](#)
- [“Example 5: Retrieve data from all the systems in a CPC” on page 385](#)

## Example 1: Retrieve data from one system

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from system sys057, submit the following request:

```
GET /zosmf/gateway/system?content={\"target\":\"sys057\",
\"resourcePath\":\"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP\",
\"contentType\":\"application/json\",\"charset\":\"UTF8\"}} HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 156. Sample request to retrieve data from one system

A sample response is shown in [Figure 157](#) on page 381.

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  \"primaryAPIVersion\":1.0,
  \"systemsOutput\":
  {
    \"systemOutput\":
    {
      \"error\":null,
      \"result\":[
        {
          \"id\":\"IBM.ZOSMF.IZU_IMPORT_HANDLER\",
          \"taskId\":\"IZUG_TASK_zOSMFImportManager\",
          \"enabled\":true,
          \"defaultHandler\":false,
          \"applId\":\"IzuImportManager\",
          \"type\":\"INTERNAL\",
          \"displayName\":\"Import Manager\",
          \"url\":\"/zosmf/IzuImportUtility/index.jsp\",
          \"eventId\":\"IBM.ZOSMF.IMPORT_EXTERNAL_APP\",
          \"options\":{\"CONTEXT_SUPPORT\":\"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH\"}
        }
      ],
      \"rc\":\"Ok\",
      \"secondaryApiVersion\":1.0,
      \"systemVersion\":{\"zosNode\":\"SY1\",\"zosVim\":\"04.24.00\",\"zosSysplex\":\"PLEX1\"},
      \"systemName\":\"sys057\"
    },
    \"numOfSystems\":1
  }
}
```

Figure 157. Sample response from a request to retrieve data from one system

## Example 2: Retrieve data from a list of systems

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from system sys057 and for event type IBM.ZOSMF.VIEW\_DATASET from system sys060, submit the following request:

```
GET /zosmf/gateway/systems?content=[{\"target\":\"sys057\",
\"resourcePath\":\"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP\",
\"contentType\":\"application/json\",\"charset\":\"UTF8\"},{\"target\":\"sys060\",
\"resourcePath\":\"/izual/rest/handler?eventId=IBM.ZOSMF.VIEW_DATASET\"}] HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 158. Sample request to retrieve data from a list of systems

A sample response is shown in [Figure 159](#) on page 382.

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_zOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventTypeid":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ],
        "rc":"Ok",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys057"
      },
      {
        "systemOutput":
        {
          "error":null,
          "result":[
            {
              "id":"IBM.ISPF.ISR.EPDF.B",
              "enabled":true,
              "defaultHandler":false,
              "type":"EXTERNAL",
              "displayName":"ISR Browse Data Set",
              "url":
                "%20NEWAPPL(ISR)"/zosmf/webispf/index.jsp?cmd=ISPSTART%20CMD(%25ISREPDF%20'%24dataSetName'%20B)
              "eventTypeid":"IBM.ZOSMF.VIEW_DATASET",
              "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH"}
            }
          ],
          "rc":"Ok",
          "secondaryApiVersion":1.0,
          "systemVersion":{"zosNode":"SY4","zosVrm":"04.24.00","zosSysplex":"PLEX4"},
          "systemName":"sys060"
        }
      ]
    },
    {
      "numOfSystems":2
    }
  ]
}

```

Figure 159. Sample response from a request to retrieve data from a list of systems

### Example 3: Retrieve data from all the systems in a group

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from all the systems in group *mygroup*, submit the following request:

```

GET /zosmf/gateway/group?content={"target":"mygroup",
  "resourcePath":"/izual/rest/handler?eventTypeid=IBM.ZOSMF.IMPORT_EXTERNAL_APP"} HTTP/1.1

Host: zosmf1.yourco.com

```

Figure 160. Sample request to retrieve data from all the systems in a group

A sample response is shown in [Figure 161 on page 383](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_zOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventTypeid":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ],
        "rc":"Ok",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys057"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys058\" failed with return code \"LoginRequired\" and HTTP response code \"401\"."
          },
          "result":null
        },
        "rc":"LoginRequired",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY2","zosVrm":"04.24.00","zosSysplex":"PLEX2"},
        "systemName":"sys058"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys059\" failed with return code \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
          },
          "result":null
        },
        "rc":"HttpConnectionTimedOut",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY3","zosVrm":"04.24.00","zosSysplex":"PLEX3"},
        "systemName":"sys059"
      }
    ],
    "numOfSystems":3
  }
}

```

Figure 161. Sample response from a request to retrieve data from all the systems in a group

#### Example 4: Retrieve data from all the systems in a sysplex

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from all the systems in sysplex *PLEX1*, submit the following request:

```
GET /zosmf/gateway/sysplex?content={"target":"PLEX1",
"resourcePath":"/izual/rest/handler?eventType=IBM.ZOSMF.IMPORT_EXTERNAL_APP"} HTTP/1.1

Host: zosmf1.yourco.com
```

*Figure 162. Sample request to retrieve data from all the systems in a sysplex*

A sample response is shown in [Figure 163 on page 384](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_zOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventType":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ],
        "rc":"Ok",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys057"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys077\" failed with return code
              \"LoginRequired\" and HTTP response code \"401\"."
          },
          "result":null
        },
        "rc":"LoginRequired",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys077"
      },
      {
        "systemOutput":
        {
          "error":
          {
            "msgid":"IZUG0000E",
            "msgtxt":"The HTTPS request to server \"sys195\" failed with return code
              \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
          },
          "result":null
        },
        "rc":"HttpConnectionTimedOut",
        "secondaryApiVersion":1.0,
        "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
        "systemName":"sys195"
      }
    ],
    "numOfSystems":3
  }
}
```

*Figure 163. Sample response from a request to retrieve data from all the systems in a sysplex*

## Example 5: Retrieve data from all the systems in a CPC

To retrieve the handlers that are registered for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from all the systems in CPC *CPC1*, submit the following request:

```
GET /zosmf/gateway/cpc?content={"target":"CPC1",
"resourcePath":"/izual/rest/handler?eventId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"} HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 164. Sample request to retrieve data from all the systems in a CPC

A sample response is shown in Figure 165 on page 385.

```
HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":[
          {
            "id":"IBM.ZOSMF.IZU_IMPORT_HANDLER",
            "taskId":"IZUG_TASK_zOSMFImportManager",
            "enabled":true,
            "defaultHandler":false,
            "applId":"IzuImportManager",
            "type":"INTERNAL",
            "displayName":"Import Manager",
            "url":"/zosmf/IzuImportUtility/index.jsp",
            "eventTypeid":"IBM.ZOSMF.IMPORT_EXTERNAL_APP",
            "options":{"CONTEXT_SUPPORT":"OPT_CONTEXT_SUPPORT_LAUNCH_AND_SWITCH"}
          }
        ]
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY5","zosVrm":"04.24.00","zosSysplex":"PLEX5"},
      "systemName":"sys289"
    }
  ],
  "numOfSystems":2
}
```

Figure 165. Sample response from a request to retrieve data from all the systems in a CPC

# Update data for one or more systems

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to update data for one system, for a list of systems, or for all the systems in a group, sysplex, or central processor complex (CPC).

## HTTP method and URI path

```
POST /zosmf/gateway/system
POST /zosmf/gateway/systems
POST /zosmf/gateway/group
POST /zosmf/gateway/sysplex
POST /zosmf/gateway/cpc
PUT /zosmf/gateway/system
PUT /zosmf/gateway/systems
PUT /zosmf/gateway/group
PUT /zosmf/gateway/sysplex
PUT /zosmf/gateway/cpc
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **system** informs the service that the request will be routed to only one system.
- **systems** informs the service that the request will be routed to a list of systems.
- **group** informs the service that the request will be routed to all of the systems in a group.
- **sysplex** informs the service that the request will be routed to all of the systems in a sysplex.
- **cpc** informs the service that the request will be routed to all of the systems in a CPC.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

Your request must include a JSON object or JSON object stream that describes the objects to be created, updated, or modified for each system. [Table 286 on page 386](#) lists the supported parameters.

Table 286. Supported input parameters for the multisystem routing services		
Parameter	Required	Description
target	Yes	If the request is being sent to a system or a list of systems, the target is the nickname of the system. If the request is being sent to all the systems in a group, sysplex, or CPC, the target is the name of the group, sysplex, or CPC. The specified target must be defined in the Systems task. Otherwise, the request will fail.

Table 286. Supported input parameters for the multisystem routing services (continued)

Parameter	Required	Description
<b>resourcePath</b>	Yes	<p>Path to the z/OSMF REST service that will process the request. The resource path must be within the z/OSMF context. For example, to ping a TSO/E address space on the target system, you would use the TSO/E address space services to process the request. Therefore, you would specify the following resourcePath: /tsoApp/ping/&lt;servletKey&gt;, where &lt;servletKey&gt; identifies the TSO/E address space for the service to ping.</p> <p>When sending an HTTPS request to a list of systems, you can specify a different resource path and different parameters for each system included in the list. When sending an HTTPS request to all the systems in a group, sysplex, or CPC, you can specify only one resource path and one set of parameters, which will be used for all the systems in the specified group, sysplex, or CPC.</p>
<b>requestProperties</b>	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
<b>timeout</b>	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
<b>content</b>	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the z/OSMF REST interface that will process the request.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Required authorizations”](#) on page 376.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 378.

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data”](#) on page 376.

## Examples

To obtain sample HTTPS requests and responses for updating data for one system, for a list of systems, or for all the systems in a group, sysplex, or CPC, see the following sections:

- [“Example 1: Update data for one system”](#) on page 388
- [“Example 2: Update data for a list of systems”](#) on page 388
- [“Example 3: Update data for all the systems in a group”](#) on page 389

- [“Example 4: Update data for all the systems in a sysplex” on page 390](#)
- [“Example 5: Update data for all the systems in a CPC” on page 391](#)

## Example 1: Update data for one system

To create event type IBM.ZOSMF.VIEW\_JOB\_STATUS on system sys057, submit the following request:

```
POST /zosmf/gateway/system HTTP/1.1
Host: zosmf1.yourco.com

{"target":"sys057","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.","owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}}
```

Figure 166. Sample request to update data for one system

A sample response is shown in [Figure 167 on page 388](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":
  {
    "systemOutput":
    {
      "error":null,
      "result":null,
    },
    "rc":"Ok",
    "secondaryApiVersion":1.0,
    "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
    "systemName":"sys057"
  },
  "numOfSystems":1
}
```

Figure 167. Sample response from a request to update data for one system

## Example 2: Update data for a list of systems

To create event type IBM.ZOSMF.VIEW\_JOB\_STATUS on system sys057 and event type IBM.ZOSMF.VIEW\_WLM\_STATUS on system sys060, submit the following request:

```
POST /zosmf/gateway/systems HTTP/1.1
Host: zosmf1.yourco.com

[{"target":"sys057","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.","owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}},
{"target":"sys060","resourcePath":"/izual/rest/eventtype",
"content":{"id":"IBM.ZOSMF.VIEW_WLM_STATUS","displayName":"View WLM Status",
"desc":"View the status of WLM.","owner":"IBM","params":{"sysplex":"Name of the sysplex."}}}]
```

Figure 168. Sample request to update data for a list of systems

A sample response is shown in [Figure 169 on page 389](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY4","zosVrm":"04.24.00","zosSysplex":"PLEX4"},
      "systemName":"sys060"
    }
  ],
  "numOfSystems":2
}

```

Figure 169. Sample response from a request to update data for a list of systems

### Example 3: Update data for all the systems in a group

To create event type IBM.ZOSMF.VIEW\_JOB\_STATUS for all the systems in group *mygroup*, submit the following request:

```

POST /zosmf/gateway/group HTTP/1.1
Host: zosmf1.yourco.com

{"target":"mygroup","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.", "owner":"SDSF", "params":{"jobName":
"Name of the job for which to view status."}}}

```

Figure 170. Sample request to update data for all the systems in a group

A sample response is shown in [Figure 171 on page 390](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY2","zosVrm":"04.24.00","zosSysplex":"PLEX2"},
      "systemName":"sys058"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY3","zosVrm":"04.24.00","zosSysplex":"PLEX3"},
      "systemName":"sys059"
    }
  ],
  "numOfSystems":3
}

```

Figure 171. Sample response from a request to update data for all the systems in a group

#### Example 4: Update data for all the systems in a sysplex

To create event type IBM.ZOSMF.VIEW\_JOB\_STATUS for all the systems in sysplex *PLEX1*, submit the following request:

```

POST /zosmf/gateway/sysplex HTTP/1.1
Host: zosmf1.yourco.com

{"target":"PLEX1","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.", "owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}}

```

Figure 172. Sample request to update data for all the systems in a sysplex

A sample response is shown in [Figure 173 on page 391](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys077"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys195"
    }
  ],
  "numOfSystems":3
}

```

Figure 173. Sample response from a request to update data for all the systems in a sysplex

## Example 5: Update data for all the systems in a CPC

To create event type IBM.ZOSMF.VIEW\_JOB\_STATUS for all the systems in CPC *CPC1*, submit the following request:

```

POST /zosmf/gateway/cpc HTTP/1.1
Host: zosmf1.yourco.com

{"target":"CPC1","resourcePath":"/izual/rest/eventtype","content":
{"id":"IBM.ZOSMF.VIEW_JOB_STATUS","displayName":"View Job Status",
"desc":"View the status of a job.", "owner":"SDSF","params":{"jobName":
"Name of the job for which to view status."}}}

```

Figure 174. Sample request to update data for all the systems in a CPC

A sample response is shown in [Figure 175 on page 392](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null,
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY5","zosVrm":"04.24.00","zosSysplex":"PLEX5"},
      "systemName":"sys289"
    }
  ],
  "numOfSystems":2
}

```

Figure 175. Sample response from a request to update data for all the systems in a CPC

## Delete data from one or more systems

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to delete data from one system, from a list of systems, or from all the systems in a group, sysplex, or central processor complex (CPC).

### HTTP method and URI path

```

DELETE /zosmf/gateway/system?content=<http-content>
DELETE /zosmf/gateway/systems?content=<http-content>
DELETE /zosmf/gateway/group?content=<http-content>
DELETE /zosmf/gateway/sysplex?content=<http-content>
DELETE /zosmf/gateway/cpc?content=<http-content>

```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **system** informs the service that the request will be routed to only one system.
- **systems** informs the service that the request will be routed to a list of systems.
- **group** informs the service that the request will be routed to all of the systems in a group.
- **sysplex** informs the service that the request will be routed to all of the systems in a sysplex.
- **cpc** informs the service that the request will be routed to all of the systems in a CPC.
- **content=<http-content>** represents the parameters used to qualify the request. [Table 287 on page 393](#) lists the parameters that are supported for this request.

**Important:** If the value for a parameter contains a number sign (#), encode the number sign as %23. Otherwise, everything following the number sign will be omitted from the request. For example, if the target is *System#1*, specify *System%231*.

Table 287. Supported input parameters for the multisystem routing services

Parameter	Required	Description
<b>target</b>	Yes	If the request is being sent to a system or a list of systems, the target is the nickname of the system. If the request is being sent to all the systems in a group, sysplex, or CPC, the target is the name of the group, sysplex, or CPC. The specified target must be defined in the Systems task. Otherwise, the request will fail.
<b>resourcePath</b>	Yes	<p>Path to the z/OSMF REST service that will process the request. The resource path must be within the z/OSMF context. For example, to ping a TSO/E address space on the target system, you would use the TSO/E address space services to process the request. Therefore, you would specify the following resourcePath: /tsoApp/ping/&lt;servletKey&gt;, where &lt;servletKey&gt; identifies the TSO/E address space for the service to ping.</p> <p>When sending an HTTPS request to a list of systems, you can specify a different resource path and different parameters for each system included in the list. When sending an HTTPS request to all the systems in a group, sysplex, or CPC, you can specify only one resource path and one set of parameters, which will be used for all the systems in the specified group, sysplex, or CPC.</p>
<b>requestProperties</b>	No	HTTP headers to be included in the HTTP request. Specify the HTTP headers as name and value pairs. If HTTP headers are omitted or are <i>null</i> , default values will be used, which are valid for most installations.
<b>timeout</b>	No	Amount of time in milliseconds allowed to process a request. The value can range from 1 to 5601000 milliseconds. If omitted, the default value of 20000 milliseconds is used.
<b>content</b>	Yes if the HTTP method is POST or PUT.	Parameters or JSON object to include in the body of the HTTPS request that will be sent to the z/OSMF REST interface that will process the request.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 376](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 378](#).

The response also includes a JSON object that contains the requested information. For more details, see [“Content type used for HTTP response data” on page 376](#).

## Examples

To obtain sample HTTPS requests and responses for deleting data from one system, from a list of systems, or from all the systems in a group, sysplex, or CPC, see the following sections:

- [“Example 1: Delete data from one system” on page 394](#)
- [“Example 2: Delete data from a list of systems” on page 395](#)
- [“Example 3: Delete data from all the systems in a group” on page 395](#)
- [“Example 4: Delete data from all the systems in a sysplex” on page 396](#)
- [“Example 5: Delete data from all the systems in a CPC” on page 397](#)

### Example 1: Delete data from one system

To remove handler IBM.ZOSMF.IZU\_IMPORT\_HANDLER for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from system sys057, submit the following request:

```
DELETE /zosmf/gateway/system?content={\"target\":\"sys057\",
\"resourcePath\":\"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP\"}
HTTP/1.1

Host: zosmf1.yourco.com
```

*Figure 176. Sample request to delete data from one system*

A sample response is shown in [Figure 177 on page 394](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  \"primaryAPIVersion\":1.0,
  \"systemsOutput\":
    {
      \"systemOutput\":
        {
          \"error\":null,
          \"result\":null
        },
      \"rc\":\"Ok\",
      \"secondaryApiVersion\":1.0,
      \"systemVersion\":{\"zosNode\":\"SY1\",\"zosVim\":\"04.24.00\",\"zosSysplex\":\"PLEX1\"},
      \"systemName\":\"sys057\"
    },
  \"numOfSystems\":1
}
```

*Figure 177. Sample response from a request to delete data from one system*

## Example 2: Delete data from a list of systems

To remove handler IBM.ZOSMF.IZU\_IMPORT\_HANDLER for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from system sys057 and to remove event type IBM.ZOSMF.VIEW\_DATASET from system sys060, submit the following request:

```
DELETE /zosmf/gateway/systems?content=[{"target":"sys057",
"resourcePath":"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"},
{"target":"sys060","resourcePath":"/izual/rest/eventtype/IBM.ZOSMF.VIEW_DATASET"}] HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 178. Sample request to delete data from a list of systems

A sample response is shown in [Figure 179](#) on page 395.

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":
        {
          "msgid":"IZUG698E",
          "msgtxt":"The request could not be completed because 1 handlers are registered for
event type \"IBM.ZOSMF.VIEW_DATASET\"."
        },
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY4","zosVrm":"04.24.00","zosSysplex":"PLEX4"},
      "systemName":"sys060"
    }
  ],
  "numOfSystems":2
}
```

Figure 179. Sample response from a request to delete data from a list of systems

## Example 3: Delete data from all the systems in a group

To remove handler IBM.ZOSMF.IZU\_IMPORT\_HANDLER for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from all the systems in group *mygroup*, submit the following request:

```
DELETE /zosmf/gateway/group?content={"target":"mygroup",
"resourcePath":"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"}
HTTP/1.1

Host: zosmf1.yourco.com
```

Figure 180. Sample request to delete data from all the systems in a group

A sample response is shown in [Figure 181 on page 396](#).

```
HTTP/1.1 200 OK
Date: Fri, 16 Jan 2015 04:13:56 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY2","zosVrm":"04.24.00","zosSysplex":"PLEX2"},
      "systemName":"sys058"
    },
    {
      "systemOutput":
      {
        "error":
        {
          "msgid":"IZUG0000E",
          "msgtxt":"The HTTPS request to server \"sys059\" failed with return code \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
        },
        "result":null
      },
      "rc":"HttpConnectionTimedOut",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY3","zosVrm":"04.24.00","zosSysplex":"PLEX3"},
      "systemName":"sys059"
    }
  ],
  "numOfSystems":3
}
```

*Figure 181. Sample response from a request to delete data from all the systems in a group*

#### Example 4: Delete data from all the systems in a sysplex

To remove handler IBM.ZOSMF.IZU\_IMPORT\_HANDLER for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from all the systems in sysplex *PLEX1*, submit the following request:

```
DELETE /zosmf/gateway/sysplex?content={"target":"PLEX1",
"resourcePath":"/izual/test/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
eventTypeId=IBM.ZOSMF.IMPORT_EXTERNAL_APP"}
HTTP/1.1

Host: zosmf1.yourco.com
```

*Figure 182. Sample request to delete data from all the systems in a sysplex*

A sample response is shown in [Figure 183 on page 397](#).

```

HTTP/1.1 200 OK
Date: Fri, 16 Feb 2015 04:13:56 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys077"
    },
    {
      "systemOutput":
      {
        "error":
        {
          "msgid":"IZUG0000E",
          "msgtxt":"The HTTPS request to server \"sys195\" failed with return code \"HttpConnectionTimedOut\" and HTTP response code \"0\"."
        },
        "result":null
      },
      "rc":"HttpConnectionTimedOut",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys195"
    }
  ],
  "numOfSystems":3
}

```

Figure 183. Sample response from a request to delete data from all the systems in a sysplex

## Example 5: Delete data from all the systems in a CPC

To remove handler IBM.ZOSMF.IZU\_IMPORT\_HANDLER for event type IBM.ZOSMF.IMPORT\_EXTERNAL\_APP from all the systems in CPC *CPC1*, submit the following request:

```

DELETE /zosmf/gateway/cpc?content={"target":"CPC1",
  "resourcePath":"/izual/rest/handler/IBM.ZOSMF.IZU_IMPORT_HANDLER?
  eventType=IBM.ZOSMF.IMPORT_EXTERNAL_APP"}
HTTP/1.1

Host: zosmf1.yourco.com

```

Figure 184. Sample request to delete data from all the systems in a CPC

A sample response is shown in [Figure 185 on page 398](#).

```

HTTP/1.1 200 OK
Date: Fri, 16 Feb 2015 04:13:56 +0000GMT
Connection: close

{
  "primaryAPIVersion":1.0,
  "systemsOutput":[
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY1","zosVrm":"04.24.00","zosSysplex":"PLEX1"},
      "systemName":"sys057"
    },
    {
      "systemOutput":
      {
        "error":null,
        "result":null
      },
      "rc":"Ok",
      "secondaryApiVersion":1.0,
      "systemVersion":{"zosNode":"SY5","zosVrm":"04.24.00","zosSysplex":"PLEX5"},
      "systemName":"sys289"
    }
  ],
  "numOfSystems":2
}

```

*Figure 185. Sample response from a request to delete data from all the systems in a CPC*

## Authenticate with a secondary z/OSMF instance

You can use this operation to request that the primary z/OSMF instance submit an HTTPS request to authenticate with a secondary z/OSMF instance.

### HTTP method and URI path

```
POST /zosmf/gateway/logon
```

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **logon** informs the service that the request is to authenticate with a system.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

Your request must include the following JSON object:

```

{
  "userid":"user-ID",
  "password":"password",

```

```
"systemName": "system-name"
}
```

where:

**user-ID**

z/OS user ID that allows the user to access the specified system. The user ID is the same user ID that is specified in your installation's z/OS security management facility (for example, RACF). The user ID is required.

**password**

Password or pass phrase associated with the z/OS user ID. The password is required.

**system-name**

Unique name assigned to the system definition.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 376](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 378](#).

The response also includes a JSON object that indicates whether the request was successful. If the logon request is successful, the timeout for the Lightweight Third Party Authentication (LTPA) token is returned, as depicted in [Figure 186 on page 399](#).

```
{"timeout":7564710}
```

*Figure 186. Successful response when authenticating with a system*

If the logon request is unsuccessful, the JSON object contains an error message, as depicted in [Figure 187 on page 399](#).

```
{"error":true,"errMsg":"IZUG410E: The user ID, password, or pass phrase is not valid.  
Enter the correct values for your security management product."}
```

*Figure 187. Response when the authentication request fails*

## Example

To authenticate with system `sys057`, submit the following request:

```
POST /zosmf/gateway/logon HTTP/1.1
Host: zosmf1.yourco.com

{"userid":"claire","password":"abc123","systemName":"sys057"}
```

*Figure 188. Sample request to authenticate with a system*

## Authenticate with an HTTP proxy server

You can use this operation to authenticate with the HTTP proxy server that the primary z/OSMF instance is required to navigate to communicate with a secondary z/OSMF instance.

### HTTP method and URI path

---

```
POST /zosmf/gateway/logon/proxy
```

---

where:

- **zosmf/gateway** identifies the multisystem routing services.
- **logon/proxy** informs the service that the request is to authenticate with an HTTP proxy server.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

Your request must include the following JSON object:

```
{
  "proxyUserId": "proxy-user-ID",
  "proxyPassword": "proxy-password",
  "systemName": "system-name"
}
```

where:

#### **proxy-user-ID**

User ID that allows the user to access the HTTP proxy server at your enterprise. The user ID is required.

#### **proxy-password**

Password or pass phrase associated with the proxy user ID. The password is required.

#### **system-name**

Unique name assigned to the system definition that specifies the URL for accessing the secondary z/OSMF instance.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### Required authorizations

See [“Required authorizations” on page 376](#).

### Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 378](#).

The response also includes a JSON object that indicates whether the request was successful. If the logon request is successful, *null* values are returned for the *result* and *error* attributes, as depicted in [Figure 189](#) on page 401.

```
{"result":null,"error":null}
```

*Figure 189. Successful response when authenticating with an HTTP proxy server*

If the logon request is unsuccessful, the JSON object contains an error message, as depicted in [Figure 190](#) on page 401. For a description of each attribute, see “Content type used for HTTP response data” on page 376.

```
{
  "primaryAPIVersion":1.0,
  "systemsOutput":null,
  "error":{
    "msgid":"IZUG476E",
    "msgtxt":"The HTTP request to the secondary z/OSMF instance \"sys057\"
    failed with error type \"InvalidProxyLogin\" and response code \"407\"."
  },
  "numOfSystems":0
}
```

*Figure 190. Sample response when the authentication request fails*

## Example

To authenticate with the HTTP proxy server that is between the primary z/OSMF instance and the z/OSMF instance that is running on system *sys057*, submit the following request:

```
POST /zosmf/gateway/logon/proxy HTTP/1.1
Host: zosmf1.yourco.com

{"proxyUserId":"claire","proxyPassword":"abc123","systemName":"sys057"}
```

*Figure 191. Sample request to authenticate with an HTTP proxy server*

## MVS subsystem services

The MVS subsystem services API is provided for z/OSMF tasks and vendor applications. This API is used to list the MVS subsystems on a z/OS system.

Table 288. Subsystem services method	
Operation	HTTP method and URI path
<a href="#">“List MVS subsystems” on page 402</a>	GET /zosmf/rest/mvssubs

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

**HTTP 200 OK**

Request was processed successfully.

**HTTP 206 Partial content**

Request was processed successfully, however, only a portion of the available content was received. The request contained the X-IBM-Max-Items header, which limited the amount of content that was returned.

**HTTP 400 Bad request**

Request could not be processed because it contains a syntax error or an incorrect parameter.

**HTTP 401 Unauthorized**

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

**HTTP 405 Method not allowed**

Requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

**HTTP 500 Internal server error**

Server encountered an error. See the response body for a JSON object with information about the error.

**HTTP 503 Service unavailable**

Server is not available.

**Error logging**

Errors from the MVS subsystem services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## List MVS subsystems

You can use the GET method to list the subsystems on a z/OS system. You can filter the returned list of subsystems by specifying a subsystem id or wild-card.

**HTTP method and URI path**

---

```
GET /zosmf/rest/mvssubs
GET /zosmf/rest/mvssubs?ssid=filter-criteria
```

---

Where:

- **/zosmf/rest** specifies the z/OSMF REST services API.
- **/mvssubs** indicates an MVS subsystems request.

**Query parameters****ssid**

An optional query parameter that can be used to qualify the request.

**Response Body**

If the request is successfully executed, will return 200 status code. In all cases an application/json document will be returned:

Table 289. Response

Property	Description	Required
subsys	Subsystem name. The subsystem ID or hexadecimal ID if the subsystem ID is not printable.	Yes
active	True if the subsystem is active, otherwise false.	Yes
funcs	An array of integer values, representing the subsystem function IDs that are defined by this subsystem.	Yes
primary	True if the subsystem is the primary subsystem.	No
dynamic	True if the subsystem is a dynamically defined subsystem.	No
commands	True for a dynamic subsystem that supports commands.	No
incomplete	True if the returned information for a dynamic subsystem is incomplete.	No
eventrtn	True if a dynamic subsystem has an event routine.	No
JSONversion	JSON version.	No

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example

Refer to [Figure 192 on page 404](#) for an example of a list of MVS subsystems.

```
request:GET https://zosmf1.yourco.com/zosmf/rest/mvssubs HTTP/1.1
Content-type: application/json; charset=UTF-8
JSON response document:

{"items":[
  {"subsys":"JES2", "active":true, "primary":true, "dynamic":true,
    "funcs":[1,2,3,4,5,6,7,8,9,10,11,12,13,16,17,18,19,20,21,53,54,64,70,71,75,77,79,80,81,82,83,84,85]},
  {"subsys":"MSTR", "active":true, "funcs":[4,5,6,8,9,10,12,14,15,32,33,48,50,54,63,68,72,73,78,80]},
  {"subsys":"SMS ", "active":true, "dynamic":true, "funcs":[8,15,16,17,55]},
  {"subsys":"RACF", "active":true, "dynamic":true, "funcs":[10,16,17,39,255]},
  {"subsys":"IRLM", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"JRLM", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"LOGR", "active":true, "dynamic":true, "funcs":[7,16,17,18,19,38,39]},
  {"subsys":"RRS ", "active":true, "dynamic":true, "funcs":[10,54]},
  {"subsys":"RRSA", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"BLSR", "active":true, "dynamic":true, "funcs":[7,16,17,38,39]},
  {"subsys":"ISPF", "active":false, "dynamic":true, "funcs":[]},
  {"subsys":"DSN9", "active":true, "funcs":[4,8,10,41,50]},
  {"subsys":"IRL9", "active":true, "funcs":[51]},
  {"subsys":"AXR ", "active":true, "dynamic":true, "funcs":[10]},
  {"subsys":"03E20023", "active":true, "funcs":[4,8,10]}
],
"JSONversion":1}
```

Figure 192. List MVS subsystems

## Notification services

The Notification services are provided for z/OSMF tasks and vendor applications. These services are used to send a notification in the form of a notification record or email, to a single or multiple recipients. On a successful request, all of the recipients get the notification in their z/OSMF Notification task as the default destination.

Table 290. Notification methods

Operation	HTTP method and URI path
<b><u>“Get all of the notifications received by the current user” on page 405</u></b>	GET /zosmf/notifications/inbox
<b><u>“Send a notification from a z/OSMF task, when the content is the message from the bundle file ” on page 407</u></b>	POST /zosmf/notifications/new
<b><u>“Send a notification and mail from a z/OSMF task or z/OSMF user” on page 409</u></b>	POST /zosmf/notifications/new

### Using the Swagger interface

You can use the Swagger interface to display information about the Notification services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

### Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document” on page 640](#).

The following HTTP status codes are valid:

#### HTTP 200 OK

Request was processed successfully.

#### HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

### HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

### HTTP 404 Not found

Requested resource does not exist.

### HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

## Error logging

Errors from the z/OSMF notifications services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Get all of the notifications received by the current user

You can use this operation to get all of the notifications that were received by the current user. This operation supports only the user to get notification items in the z/OSMF Notifications task. This does not apply to the get mail operation in a user's email account.

### HTTP method and URI path

```
GET /zosmf/notifications/inbox
```

### Query Parameters

None.

### Request

None.

### Response Content

On successful completion, the service returns a response body, which contains details about the notifications. [Table 291](#) on page 405 lists the fields in the response body.

Table 291. Response content from the notifications received by the current user		
Field	Type	Description
taskId	String	This is the ID of the task where the notification is initiated.
pluginId	String	This is the ID of the plug-in where the notification is initiated.
appLinkEventId	String	This is the event ID for an application event.
assignees	String	The user IDs, group names, or email addresses of all the recipients. These values are represented as a string, and are separated by a comma.

Table 291. Response content from the notifications received by the current user (continued)

Field	Type	Description
descriptionParms	String	These are the parameters which substitute the description.
defaultDescription	String	This is the default description of the notification.
descriptionId	String	This is the message ID of the description.
descriptionBundleURL	String	This is the bundle file where the description of the notification can be found.
appLinkParms	String	This is the map of the parameters, which are sent with an event.

## Authorization Requirements

Use of the Notification RESTful services API requires the client to be authenticated. For information about client authentication in z/OSMF. See [“Authenticating to z/OSMF”](#) on page 2.

You will also need SAF authority, as described in [Security structures for z/OSMF](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

On successful completion, HTTP status code 200 OK is returned and the response body is provided, see [Table 291](#) on page 405.

The HTTP status codes and error handling are described in [“Error handling”](#) on page 404.

## Example

Receive messages in your inbox.

### Request

```
GET /zosmf/notifications/inbox
```

### Response

```
{
  "items": [
    {
      "appLinkHandlers": true,
      "taskId": "Workflows",
      "pluginID": "workflow",
      "descriptionBundle": "WorkflowMessages",
      "appLinkEventID": "IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW",
      "userRead": false,
      "assignees": "zosmfad",
      "descriptionParms": ["testing -service"],
      "defaultDescription":
        "One or more steps in workflow \"testing -service\" have been assigned to
you.",
      "descriptionID": "IZUWF0040I",
      "timestamp": "1429860854757",
      "notificationID": "1429860854756",
      "bundleUrl": "\/zosmf\/workflow\/js\/zosmf",
      "descriptionBundleURL": "\/zosmf\/workflow\/js\/zosmf\/",
      "defaultTaskName": "Workflows",
      "appLinkParms": {
        "workflow_name": "testing -service"
      },
      "bundleName": "taskBundle",
      "numUnreadNotification": 1
    }
  ]
}
```

## Send a notification from a z/OSMF task, when the content is the message from the bundle file

This operation is used to send a notification from a z/OSMF task and the content of the notification is the message from the bundle file.

### HTTP method and URI path

POST /zosmf/notifications/new

### Query Parameters

None

### Description

The content of the notification contains the message from the bundle file. This operation supports application linking from the notification task to the receiver task. The destination of a notification will depend on user preferences.

### Request Content

A notification is sent from a z/OSMF task. The content should include the message ID and the message text, which originate from a bundle file. In this case the value of the post body should be JSON Object-like. See [Table 292 on page 407](#).

Table 292. Request content for the send notification request			
Input	Description	Type	Required or Optional
<b>pluginId</b>	ID of the plug-in where the notification is initiated.	String	Required
<b>taskId</b>	ID of the task where the notification is initiated.	String	Required
<b>assignees</b>	The user IDs, group names, or email addresses of all the recipients.	String	Required
<b>sendTo</b>	The assignees will receive the notification in the form of email to their configured email address. Mail is the only supported value of this parameter.	String	Required
<b>descriptionBundleURL</b>	Bundle file where the description of the notification can be found.	String	Required
<b>descriptionId</b>	Message ID of the description.	String	Required

<i>Table 292. Request content for the send notification request (continued)</i>			
<b>Input</b>	<b>Description</b>	<b>Type</b>	<b>Required or Optional</b>
<b>defaultDescription</b>	Default description of the notification.	String	Required
<b>descriptionParms</b>	Parameters that substitute the description.	String	Optional
<b>appLinkEventId</b>	Event ID for an application event.	String	Optional
<b>appLinkParms</b>	Map of the parameters, which are sent with an event.	String	Optional

## Response Content

On completion, the request returns a JSON object with details about the notification. The response content is shown in [Table 293 on page 408](#).

<i>Table 293. Response content for the send notification request</i>		
<b>Field</b>	<b>Type</b>	<b>Description</b>
apiVersion	String	The version of the Notification Services API.
result	JSONObject	Contains all of the output for each notification destination. It includes 1-3 keys depending on how many destinations the notification is sent to. Each key represents one destination, and its value is a JSONObject which might have messages and return codes.

## Authorization Requirements

Use of the Notification RESTful services API requires the client to be authenticated. For information about client authentication in z/OSMF. See [“Authenticating to z/OSMF” on page 2](#).

You will also need SAF authority, as described in [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

## HTTP status codes

On successful completion, HTTP status code 200 OK is returned and the response body is provided, [Table 293 on page 408](#).

The HTTP status codes and error handling are described in [“Error handling” on page 404](#).

## Escaping special characters

The format of a notification should be a valid JSONObject. If a special character exists it must be escaped. If " exists, it needs to be escaped as \". If \ exists, it needs to be escaped as \\.

## Example

The notification is only sent to assignees' email inbox

### Request

```
POST /zosmf/notifications/new
{"subject": "Test Notification Framework",
 "content": "This is a test.",
 "assignees": "user1@abc.com,zosmfad,user2@abc.com",
 "sendTo": "mail"}
```

### Response

```
{"result":{"mail":{"rc":"Ok","messages":{}}},"apiVersion":"1.0"}
```

Send a notification to user ID's and a group.

### Request

```
POST /zosmf/notifications/new
{  "pluginId": "workflow",
   "taskId": "Workflows",
   "assignees": "zmfuser, zosmfad, z/OSMF Administrators",
   "descriptionBundleURL": "/zosmf/workflow/js/zosmf/",
   "descriptionId": "IZUWF0039I",
   "defaultDescription": "This is a default description.",
   "applLinkId": "IBM.ZOSMF.WORKFLOWS.CREATE_WORKFLOW",
   "applLinkParams": {"workflow_name": "new workflow"}}
data:{
  "event": {
    "dte": "15/12/31",
    "tme": "04:29:57",
    "sys": "SYS1",
    "cat": "2",
    "col": "red",
    "msg": "event 1 (of event list)",
    "lng": "long message",
    "viw": {"workflows":{"key":"13309779173140.992077"}}
  }
}
```

### Response

```
{"apiVersion":"1.0",
 "result":{"mail":{"messages":null,
 "rc":"Ok"},
 "notification":{"messages":null}}}
}
```

## Send a notification and mail from a z/OSMF task or z/OSMF user

This operation is used to send a notification from a z/OSMF task or a z/OSMF user.

### HTTP method and URI path

---

```
POST /zosmf/notifications/new
```

---

### Query Parameters

None

## Description

The content of the notification as well as the mail contains the user input data. This operation does not support application linking. A notification with the same subject and content will be sent to all recipients. If the "attachment" parameter is specified, the attachment will only appear in the recipients' mail.

## Request Content

The value of the post body should be JSON Object-like. See [Table 294 on page 410](#).

<i>Table 294. Request content from a notification that requires user input</i>			
Input	Description	Type	Required or Optional
<b>assignees</b>	The user IDs, group names, or email addresses of all the recipients.	String	Required
<b>subject</b>	Subject of the notification. The allowable length is 1-500.	String	Required
<b>content</b>	Notification body. The allowable length is 0-5000.	String	Optional
<b>sendTo</b>	The assignees will receive the notification in the form of email to their configured email address. Mail is the only supported value of this parameter.	String	Required
<b>attachment</b>	Array of the file paths, up to 5 attachments are allowed.	String	Optional
<b>pluginId</b>	ID of the plug-in where the notification is initiated.	String	Required, only if the notification is from a z/OSMF task.
<b>taskId</b>	ID of the task where the notification is initiated.	String	Required, only if the notification is from a z/OSMF task.

## Response Content

On completion, the request returns a JSON object with details about the notification. The response content is shown in [Table 295 on page 410](#).

<i>Table 295. Response content from a notification that requires user input</i>		
Field	Type	Description
apiVersion	String	The version of the Notification Services API.

Table 295. Response content from a notification that requires user input (continued)

Field	Type	Description
result	JSONObject	Contains all of the output for each notification destination. It includes 1-3 keys depending on how many destinations the notification is sent to. Each key represents one destination, and its value is a JSONObject which might have messages and return codes.

## Authorization Requirements

Use of the Notification RESTful services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

You will also need SAF authority, as described in [Security structures for z/OSMF in IBM z/OS Management Facility Configuration Guide](#).

## HTTP status codes

On successful completion, HTTP status code 200 OK is returned and the response body is provided. See [Table 295 on page 410](#).

The HTTP status codes and error handling are described in [“Error handling” on page 404](#).

## Escaping special characters

The format of a notification should be a valid JSONObject. If a special character exists it must be escaped. If " exists, it needs to be escaped as \". If \ exists, it needs to be escaped as \\\.

## Example

The notification is only sent to assignees' email inbox

### Request

```
POST /zosmf/notifications/new
{"subject":"Test Notification Framework",
 "content":"This is a test.",
 "assignees":"user1@abc.com,zosmfad,user2@abc.com",
 "sendTo":"mail"}
```

### Response

```
{"result":{"mail":{"rc":"Ok","messages":{}}},"apiVersion":"1.0"}
```

Send a notification with an attachment to a user ID.

### Request

```
POST zosmf/notifications/new
{"subject":"Test with unix attachment",
 "content":"See if there is an attachment.",
 "assignees":"zosmfad",
 "attachment":["\\/global/zosmf/data/logs/IZUG0.log\\"]}
```

### Response

```
{"apiVersion":"1.0",
 "result":{"mail":{"messages":{"errorData":[{"messageText":"IZUG615E: The

```

```

connection to the SMTP host \"smtp.gmail.com\" port \"587\" failed with
error type \"ConnectionTimedout\".",
"messageId": "IZUG615E"}]},
"rc": "ConnectionTimedout",
"notification": {"messages": null}}

```

## Software management services

The software management REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. This interface allows a client application to interact with the z/OSMF Software Management task.

Table 296 on page 412 lists the operations that the software management services provide.

Table 296. Operations provided through the software management services.	
Operation	HTTP method and URI path
<a href="#">“List the software instances defined to z/OSMF” on page 414</a>	GET /zosmf/swmgmt/swi
<a href="#">“Retrieve the properties of a software instance” on page 417</a>	GET /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
<a href="#">“List the data sets included in a software instance” on page 423</a>	POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/datasets
<a href="#">“Add a new software instance” on page 428</a>	POST /zosmf/swmgmt/swi
<a href="#">“Export a defined software instance” on page 434</a>	POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/export
<a href="#">“Modify the properties of a software instance” on page 440</a>	PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
<a href="#">“Load the products, features, and FMIDs for a software instance” on page 446</a>	PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/products
<a href="#">“Delete a software instance” on page 452</a>	DELETE /zosmf/swmgmt/swi/<system-nickname>/<swi-name>

Table 296. Operations provided through the software management services. (continued)

Operation	HTTP method and URI path
<a href="#">“List the portable software instances defined to z/OSMF” on page 453</a>	GET /zosmf/swmgmt/pswi
<a href="#">“Retrieve the properties of a portable software instance” on page 455</a>	GET /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>
<a href="#">“Add a new portable software instance” on page 460</a>	POST /zosmf/swmgmt/pswi
<a href="#">“Delete a portable software instance” on page 464</a>	DELETE /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>

## Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. Some errors might also include a returned JSON object that contains the following attributes:

```
{
  "error": {
    "reason": "reason-code",
    "messages": ["message-text"]
  }
}
```

where:

### error

JSON object that contains a reason code and a list of one or more message strings to describe the errors detected while processing the request.

### reason-code

Reason code returned for the request. The value is an integer.

### message-text

Array that contains the text of each message that was issued.

The following HTTP status codes are valid:

**HTTP 200 OK**

Success.

**HTTP 400 Bad request**

The request contained incorrect parameters.

**HTTP 401 Unauthorized**

The submitter of the request did not authenticate to z/OSMF.

**HTTP 403 Forbidden**

The server rejected the request.

**HTTP 404 Not found**

The target of the request was not found.

**HTTP 409 Conflict**

The request could not be completed because there is a conflict with the current state of the resource.

**HTTP 500 Internal server error**

The server encountered an error that prevented it from completing the request.

**HTTP 503 Service unavailable**

The server is currently unavailable to process the request.

## Error logging

Errors from the software management services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## List the software instances defined to z/OSMF

You can use this operation to obtain a list of the software instances that are defined to a z/OSMF instance.

### HTTP method and URI path

---

```
GET /zosmf/swmgmt/swi
```

---

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.

### Standard headers

Use the following standard HTTP header with this request:

**Accept-Language**

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, then English is used.

### Custom headers

None.

### Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, to obtain a list of the software instances that are defined to a z/OSMF instance, the user ID initiating the request must have READ access to the z/OSMF Software Management task. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 413.

If the request is successful, the response also includes the following JSON object:

```
{
  "swilist": [
    {
      "name": "swi-name",
      "system": "system-nickname",
      "description": "swi-description",
      "globalzone": "global-zone",
      "targetzones": ["target-zones"],
      "categories": ["categories"],
      "productinfo": "last-retrieved",
      "lastmodified": "last-modified",
      "modifiedby": "modified-user-ID",
      "created": "date-created",
      "createdby": "created-user-ID",
      "locked": "date-locked",
      "lockedby": "locked-user-ID",
      "swiurl": "swi-URL"
    }
  ]
}
```

where:

### **swilist**

Array that contains each software instance that is defined to z/OSMF.

### **swi-name**

Name of the software instance.

### **system-nickname**

Nickname of the z/OSMF host system that has access to the volumes and data sets where the software instance resides. To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services”](#) on page 465.

### **swi-description**

Description of the software instance.

### **global-zone**

CSI data set that contains the global zone used to manage the software. If the software instance has no global zone, then this property will be null.

### **target-zones**

Comma-separated list of the target zones included in the software instance. If the software instance has no global zone, then this property will be null.

### **categories**

Comma-separated list of the categories to which the software instance is assigned.

**last-retrieved**

Date and time the product, feature, and FMID information was last retrieved for the software instance. This attribute is blank if this information has not been retrieved.

**last-modified**

Date and time in ISO 8601 format that the software instance was last modified.

**modified-user-ID**

User ID of the user who last modified the software instance.

**date-created**

Date and time in ISO 8601 format that the software instance was created.

**created-user-ID**

User ID of the user who created the software instance.

**date-locked**

Date and time in ISO 8601 format that the software instance was locked. This attribute is null if the software instance is not currently locked.

**locked-user-ID**

User ID of the user who locked the software instance. This attribute is null if the software instance is not currently locked.

**swi-URL**

URL that allows you to access the software instance. For example, a client application can use the URL to read a software instance. For more details, see [“Retrieve the properties of a software instance” on page 417](#).

**Example**

In the following example, the GET method is used to retrieve a list of the software instances that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/swmgmt/swi HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 193. Sample request to retrieve a list of software instances*

A sample response is shown in [Figure 194 on page 416](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close

{"swilist": [
  {
    "name": "DB2V9", "system": "PEV174", "description": null,
    "globalzone": "DB2.GLOBAL.CSI", "targetzones": ["DB2TGT"], "categories": null,
    "productinfo": {"retrieved": "2014-08-20T19:23:25+00:00", "lastmodified": "2014-08-20T19:23:25+00:00", "modifiedby": "FRED", "created": "2014-08-20T19:23:25+00:00", "createdby": "BARNEY", "locked": null, "lockedby": null,
    "swiurl": "https://zosmf1.yourco.com/zosmf/swmgmt/swi/PEV174/DB2V9"}
  },
  {
    "name": "zOSV2R1", "system": "PEV174", "description": null,
    "globalzone": "ZOS.GLOBAL.CSI", "targetzones": ["MVST100", "MVST110"],
    "categories": null, "productinfo": {"retrieved": "2014-08-20T19:23:25+00:00", "lastmodified": "2014-08-20T19:23:25+00:00", "modifiedby": "WILMA", "created": "2014-08-20T19:23:25+00:00", "createdby": "BETTY", "locked": null, "lockedby": null,
    "swiurl": "https://zosmf1.yourco.com/zosmf/swmgmt/swi/PEV174/zOSV2R1"}
  }
]}
```

*Figure 194. Sample response from a request to retrieve a list of software instances*

## Retrieve the properties of a software instance

You can use this operation to retrieve the properties of a software instance. The properties include, but are not limited to, the global zone and target zones associated with the software instance and a list of the products, features, FMIDs, and non-SMP/E managed data sets that are included in the software instance.

### HTTP method and URI path

---

```
GET /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
```

---

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

### Standard headers

Use the following standard HTTP header with this request:

#### Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, then English is used.

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, to retrieve the properties of a software instance, the user ID initiating the request must have READ access to both the z/OSMF Software Management task and the software instance. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

### Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 413](#).

If the request is successful, the response also includes the following JSON object:

```
{
  "name": "swi-name",
  "system": "system-nickname",
  "description": "swi-description",
  "globalzone": "global-zone",
  "targetzones": ["target-zones"],
  "categories": ["categories"],
  "productinfo": {
    "retrieved": "last-retrieved",
    "lastmodified": "last-modified",
    "modifiedby": "modified-user-ID",
    "created": "date-created",
    "createdby": "created-user-ID",
    "locked": "date-locked",
    "lockedby": "locked-user-ID",
    "datasets": [
      {
        "dsname": "data-set-name",
        "volume": "volume-serial"
      }
    ],
    "products": [
      {
        "prodname": "product-name",
        "prodid": "product-ID",
        "release": "product-level",
        "vendor": "vendor-name",
        "generalavailability": "general-availability-date",
        "endofservice": "end-of-service",
        "url": "product-URL",
        "productinfo": {
          "fileversion": "file-version",
          "features": [
            {
              "feature": "feature-name",
              "fmids": [
                {
                  "fmid": "fmid-name",
                  "description": "fmid-description",
                  "targetzones": ["fmid-target-zones"]
                }
              ]
            }
          ]
        }
      }
    ]
  },
  "nonsmpeproducts": [
    {
      "prodname": "product-name",
      "prodid": "product-id",
      "release": "product-level",
      "vendor": "vendor-name",
      "url": "product-url",
      "features": ["feature-name"],
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service-date"
    }
  ],
  "workflows": [
    {
      "name": "workflow-name",
      "description": "workflow-description",
      "location": {
        "smptype": "smp-type",
        "smpname": "smp-name",
        "dsname": "workflow-dsname",
        "path": "workflow-path"
      },
      "performonhosts": true | false
    }
  ],
  "datasetproperties": [
    {
      "dddefname": "dddef-name",
      "zone": "zone-name",
      "dsname": "data-set-name",
      "volume": "volume-serial",
      "dstype": "DLIB",
      "properties": [{"key": "value"}]
    }
  ],
  "datasetpropertylabels": [
    {
      "propertyname": "property-name",
      "label": "property-label"
    }
  ],
  "productproperties": [
    {
      "prodid": "product-id",
      "release": "product-level",
      "prodname": "product-name",
      "properties": [{"key": "value"}]
    }
  ]
}
```

where:

**swi-name**

Name of the software instance.

**system-nickname**

Nickname of the z/OSMF host system that has access to the volumes and data sets where the software instance resides. To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

**swi-description**

Description of the software instance.

**global-zone**

CSI data set that contains the global zone used to manage the software. If the software instance has no global zone, then this property will be null.

**target-zones**

Comma-separated list of the target zones included in the software instance. If the software instance has no global zone, then this property will be null.

**categories**

Comma-separated list of the categories to which the software instance is assigned.

**last-retrieved**

Date and time the product, feature, and FMID information was last retrieved for the software instance. This attribute is blank if this information has not been retrieved.

**last-modified**

Date and time in ISO 8601 format that the software instance was last modified.

**modified-user-ID**

User ID of the user who last modified the software instance.

**date-created**

Date and time in ISO 8601 format that the software instance was created.

**created-user-ID**

User ID of the user who created the software instance.

**date-locked**

Date and time in ISO 8601 format that the software instance was locked. This attribute is null if the software instance is not currently locked.

**locked-user-ID**

User ID of the user who locked the software instance. This attribute is null if the software instance is not currently locked.

**datasets**

Array that contains each non-SMP/E managed data set that is included in the software instance.

**data-set-name**

Name of the non-SMP/E managed data set.

**volume-serial**

Volume on which the non-SMP/E managed data set resides.

**products**

Array that contains each product that is included in the software instance.

**product-name**

Name of the product. If any FMIDs are not related to a product and feature, those FMIDs are listed under a product named *No Product*.

**product-ID**

Identifier of the product.

**product-level**

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level.

**vendor-name**

Name of the vendor that provides the product.

**general-availability**

Date this level of the product is available to all users.

**end-of-service**

Last date on which the vendor will deliver standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions.

**product-URL**

URL that links to additional information about the product. This information can include, for example, product life cycle dates, product highlights, planning information, and technical descriptions.

**file-version**

Version of the most recent product information file that was retrieved that contains the corresponding product. The version represents the date that file was created or last updated.

**features**

Array that contains each feature that is included in the product.

**feature-name**

Name of the feature. If any FMIDs are not related to a product and feature, those FMIDs are listed under a feature named *No Feature*.

**fmids**

Array that contains each FMID that is included in the feature.

**fmid-name**

Name of the FMID.

**fmid-description**

Description of the FMID.

**fmid-target-zones**

Name of the target zones where the FMID is installed.

**nonsmpeproducts**

List of products for the software instance that are not managed by SMP/E.

**product-name**

Name of the product, but can be up to 64 characters.

**product-ID**

Identifier for the product, but can be up to 64 characters.

**product-level**

Release level for the product, but can be up to 64 characters.

**vendor-name**

Name of the vendor that provides the product, but can be up to 64 characters.

**product-URL**

A URL that links to additional information about the product, but can be up to 256 characters.

**feature-name**

List of names of features for the product, but can be up to 64 characters.

**general-availability-date**

Date this level of the product is available to all users.

**end-of-service-date**

Last date on which the vendor will deliver standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. Can be any of the following:

**null**

The end of service date is unknown for the product.

**yyyy-mm-ddThh:mm:ssZ**

The known end of service date, in ISO 8601 format.

**NotAnnounced**

The end of service date is not yet announced for the product.

**workflows**

List of workflows for the software instance.

**workflow-name**

Name for the workflow.

**workflow-description**

Description for the workflow.

**location**

Location of the workflow definition file for the workflow.

**smp-type**

The SMP/E element type for a workflow definition file that is managed by SMP/E.

**smp-name**

The SMP/E element name for a workflow definition file that is managed by SMP/E.

**workflow-dsname**

The name of the data set that contains the workflow definition file.

**workflow-path**

The UNIX path for a workflow definition file that is a UNIX file.

**performonhostsyst**

Indicates whether the workflow steps may be performed on the host system or on another system in the same sysplex as the host system.

**true**

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

**false**

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

**datasetproperties**

A list of one or more properties for individual data sets.

**dddefname**

The name of the SMP/E DDDEF entry that describes an SMP/E managed data set.

**zone**

The zone name where the DDDEF entry resides.

**dsname**

The name of the subject data set.

**volume**

The volume of the subject data set.

**dstype**

The usage type of the subject data set. A value of DLIB indicates the data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone.

**properties**

A list of one or more properties for the subject data set, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null.

**datasetpropertylabels**

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for

display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

**propertyname**

The name, or key, of the existing provider defined property.

**label**

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

**productproperties**

A list of one or more properties for individual software products.

**prodid**

The identifier for the subject product.

**release**

The version, release, modification level for the subject product.

**prodname**

The name of the subject product.

**properties**

A list of one or more properties for the subject product, specified as key-value pairs. The keys are strings, and the values are a valid JSON data type such as string, number, Boolean, array, object or null. See [“Provider Defined Properties” on page 995](#) for more information.

## Example

In the following example, the GET method is used to retrieve the properties of software instance DB2V9 on system PEV174.

```
GET /zosmf/swmgmt/swi/PEV174/DB2V9 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 195. Sample request to retrieve the properties of a software instance*

A sample response is shown in [Figure 196 on page 422](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Content-Type: application/json
Content-Language: en
Connection: close

{
  "name": "DB2V9", "system": "PEV174", "description": null,
  "globalzone": "DB2.GLOBAL.CSI", "targetzones": ["DB2TGT"], "categories": null,
  "productinfo": {
    "retrieved": "2014-08-20T19:23:25+00:00", "lastmodified": "2014-08-20T19:23:25+00:00",
    "modifiedby": "FRED", "created": "2014-08-20T19:23:25+00:00",
    "createdby": "BARNEY", "locked": null, "lockedby": null, "datasets": [
      { "dsname": "USER.DB2V9.PROCLIB", "volume": "LV1234" },
      { "dsname": "USER.DB2V9.SAMPLES", "volume": "LV1234" }
    ], "products": [
      { "prodname": "DB2 for z/OS", "prodid": "5635-DB2", "release": "09.01.00",
        "vendor": "IBM", "generalavailability": "20006-06-09T19:23:25+00:00",
        "endofservice": "2014-06-27T19:23:25+00:00", "url": null,
        "productinfo": { "fileversion": "2014-01-01", "features": [
          { "feature": "DB2 Base", "fmids": [
            { "fmid": "HDB9910", "description": "DB2 BASE/TSO", "targetzones": [
              "DB2V9T" ] } ] } ] } ] } ] } }
```

*Figure 196. Sample response from a request to retrieve the properties of a software instance*

## List the data sets included in a software instance

You can use this operation to obtain a list of the data sets that compose a software instance.

### HTTP method and URI path

---

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/datasets
```

---

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

- **datasets** indicates that the data sets included in the software instance are to be obtained.

When you issue this request, z/OSMF analyzes the global, target, and distribution zones included in the software instance to identify the SMP/E managed data sets and the SMP/E managed UNIX data sets that contain the installed software described in those zones. z/OSMF returns a JSON object that lists the properties of each data set it identified, along with the properties of each non-SMP/E managed data set included in the software instance.

### Standard headers

Use the following standard HTTP header with this request:

#### Content-Type

Identifies the type of input content provided by the caller. Use the JSON content type ("Content-Type: application/json") if a JSON document is included as input with this request.

#### Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

### Custom headers

None.

### Request content

If the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. To do so, include the following JSON object in your request:

```
{
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Figure 197. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server

where:

**zosmf-user-ID**

User ID for authenticating with the secondary z/OSMF instance.

**zosmf-password**

Password for authenticating with the secondary z/OSMF instance.

**proxy-user-ID**

User ID for authenticating with the HTTP proxy server.

**proxy-password**

Password for authenticating with the HTTP proxy server.

Include the JSON object in the request only if you are required to authenticate with a secondary z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, the user ID must have READ access to both the Software Management task and the SAF resources required to process the request. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request was accepted. If the request was accepted, the service returns status code 202 Accepted and a JSON object that contains a URL (`{"statusurl": "url"}`). To monitor the status of the list data sets request and to retrieve the results, perform GET requests to the supplied URL. Only the user ID that initiates the list data sets request is authorized to check the status and retrieve the results. One of the following responses is returned from the get status request:

- If the list data sets request is still in progress, an HTTP response code of 200 OK is returned, along with the following JSON object: `{"status": "status", "percentcomplete": "percent-complete"}`.
- If the list data sets request is complete, an HTTP response code of 200 OK is returned, along with the following JSON object:

```
{
  "status": "status",
  "percentcomplete": "percent-complete",
  "swidatasets": {
    "smpmanageddatasets": [ {
      "dsname": "data-set-name",
      "volumes": [ "volume-serial" ],
      "dstype": "data-set-type",
      "recfm": "record-format",
      "lrecl": "logical-record-length",
    } ]
  }
}
```

```

        "blksize": "block-size",
        "tracks": "allocated-tracks",
        "used": "used-tracks-percent",
        "extents": "allocated-extents",
        "dscategory": ["data-set-category"],
        "zones": ["zones"],
        "zonedddefs": [{
            "zone": "zone-name", "dddefs": ["dddef-name"]}],
        "msgs": ["message-text"]
    },
    "smpmanagedunixdatasets": [{
        "dsname": "data-set-name",
        "unixdirs": ["UNIX-directory"],
        "volumes": ["volume-serial"],
        "dstype": "data-set-type",
        "recfm": "record-format",
        "lrecl": "logical-record-length",
        "blksize": "block-size",
        "tracks": "allocated-tracks",
        "used": "used-tracks-percent",
        "extents": "allocated-extents",
        "dscategory": ["data-set-category"],
        "zones": ["zones"],
        "zonedddefs": [{
            "zone": "zone-name", "dddefs": ["dddef-name"]}],
        "msgs": ["message-text"]
    },
    "nonsmpmanagedddatasets": [{
        "dsname": "data-set-name",
        "volumes": ["volume-serial"],
        "dstype": "data-set-type",
        "recfm": "record-format",
        "lrecl": "logical-record-length",
        "blksize": "block-size",
        "tracks": "allocated-tracks",
        "used": "used-tracks-percent",
        "extents": "allocated-extents",
        "dscategory": ["data-set-category"],
        "unixdirs": ["UNIX-directory"],
        "msgs": ["message-text"]
    }
    ]
}

```

where:

#### **status**

Status of the list data sets request. The status is either *running* or *complete*.

#### **percent-complete**

Percentage of the processing that is complete for the list data sets request, expressed as a whole number from 0 to 100.

#### **swidatasets**

Lists all the data sets included in the software instance. A software instance can contain:

- **smpmanagedddatasets**. Array of the SMP/E managed data sets included in the software instance.
- **smpmanagedunixdatasets**. Array of the SMP/E managed UNIX file system data sets included in the software instance.
- **nonsmpmanagedddatasets**. Array of the non-SMP/E managed data sets included in the software instance.

#### **data-set-name**

Name of the data set.

#### **UNIX-directory**

Array of the UNIX directories contained in the data set. If z/OSMF could not identify the UNIX file system data set for any UNIX directories, the data set name for those directories is set to *No Data Set Found*. Typically, this occurs when the UNIX file system data set is not mounted.

#### **volume-serial**

Array of the volume serials for the volumes on which the data set resides. If the data set is migrated, a value of *MIGRAT* is returned.

**data-set-type**

Type of data set. The data set can be one of the following types:

- **HFS**. Hierarchical file system.
- **PDS**. Partitioned data set.
- **PDSE**. Partitioned data set extended.
- **Sequential**. Sequential data set.
- **VSAM**. VSAM data set.
- **ZFS**. zSeries file system.

**record-format**

Record format specified when the data set was allocated. The record format can be any valid combination of the following codes:

- **A**. ASA printer control characters.
- **B**. Blocked records.
- **F**. Fixed-length records.
- **M**. Machine code printer control characters.
- **S**. Standard (for F) or spanned (for V); used only with sequential data sets.
- **T**. Track-overflow feature.
- **U**. Undefined format records.
- **V**. Variable-length records.

**logical-record-length**

Logical record length, in bytes, specified when the data set was allocated.

**block-size**

Block size, in bytes, specified when the data set was allocated.

**allocated-tracks**

Number of tracks allocated to the data set.

**used-tracks-percent**

Percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

**allocated-extents**

Number of extents allocated to the data set.

**dscategory**

List of categories for how the data set is used. Can be one or more of the following:

- **TARGET** - SMP/E managed target library, or SMP/E control data set associated with a target zone.
- **DLIB** - SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- **GLOBAL** - SMP/E control data set associated with the global zone.
- **SMP/E** - SMP/E control data set.
- **WORKFLOW** - Contains one or more workflow definition files for the workflows explicitly defined to the software instance.

**zones**

Array of the SMP/E zones that contain a DDDEF entry for the data set. For an SMPCSI data set, it is an array of the SMP/E zones contained in the data set.

**zonedefs**

Array of SMP/E zones and DDDEF entries for the data set.

**zone-name**

Name of an SMP/E zone that contains one or more DDDEF entries for the data set.

**dddef-name**

Array of DDDEF entry names that identify the data set.

**message-text**

Array of messages returned for the data set.

- If the list data sets request is complete but the results are no longer available, an HTTP response code of 404 Not found is returned. z/OSMF makes the results available for a client application for a finite period of time. When that time elapses, the results are no longer available; in which case, the client must reissue the request.

If the list data sets request cannot be processed, a status code of 4nn or 5nn is returned, indicating that an error has occurred. For more details, see [“Error handling” on page 413](#).

**Example**

In the following example, the POST method is used to retrieve a list of the data sets included in software instance *DB2V9* on system *SYS123*.

```
POST /zosmf/swmgmt/swi/SYS123/DB2V9/datasets HTTP/1.1
Host: sys123.yourco.com
```

*Figure 198. Sample request to list the data sets included in a software instance*

Figure 199 on page 427 provides a sample response, indicating that the list data sets request has been accepted and supplying the URL to use for monitoring the status of that request.

```
HTTP/1.1 202 Accepted
Date: Tues, 21 November 2014 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/dslist/4837290198343"}
```

*Figure 199. Sample response for a list data sets request*

To check the status of the list data sets request, submit the following request:

```
GET /zosmf/swmgmt/statusmonitor/dslist/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

*Figure 200. Sample request to obtain the status of a list data sets request*

Figure 201 on page 427 provides a sample response for when the list data sets request is in progress.

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:19 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"running", "percentcomplete":"65"}
```

*Figure 201. Sample get status response when the list data sets request is in progress*

Figure 202 on page 428 provides a sample response for when the list data sets request is complete.

```

HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete", "percentcomplete":"100", "swidatasets":{
  "smpmanageddatasets":[
    {"dsname":"USERID.SMPE.CSI", "volumes":["SL730C"], "dstype":"VSAM",
      "recfm":null, "lrecl":null, "blksize":null, "tracks":"1509", "used":null,
      "extents":null, "zones":["GLOBAL", "TGT0", "DLB0"], "zonedddefs":null,
      "msgs":null},
    {"dsname":"USERID.SMPE.MACLIB", "volumes":["SL7334"], "dstype":"PDS",
      "recfm":"FB", "lrecl":"80", "blksize":"27920", "tracks":"4", "used":"100",
      "extents":"1", "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["MACLIB", "SMPMTS"]}]}], "msgs":null},
    {"dsname":"USERID.SMPE.MIGLIB", "volumes":["SL8B2D"], "dstype":"PDS",
      "recfm":"U", "lrecl":null, "blksize":"32760", "tracks":"147", "used":"100",
      "extents":"1", "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["MIGLIB"]}]}], "msgs":null},
    {"dsname":"USERID.SMPE.SGIMCLS0", "volumes":["SL7307"], "dstype":"PDS",
      "recfm":"VB", "lrecl":"255", "blksize":"32760", "tracks":"15", "used":"6",
      "extents":"1", "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["SGIMCLS0"]}]}], "msgs":null},
    {"dsname":"USERID.SMPE.SGIMMJPN", "volumes":["MIGRAT"], "dstype":null,
      "recfm":null, "lrecl":null, "blksize":null, "tracks":null, "used":null,
      "extents":null, "zones":null, "zonedddefs":null, "msgs": ["GIM70531E Data set
      USERID.SMPE.SGIMMJPN is migrated and is being recalled."]}],
    ...],
    "smpmanagedunixdatasets":[{"dsname":"USERID.SMPE.ZFS", "unixdirs":
      ["/u/userid/smpe/usr/lpp/smp/IBM/"], "volumes":["ZF3804"], "dstype":"ZFS",
      "recfm":null, "lrecl":null, "blksize":null, "tracks":"22650", "used":null,
      "extents":null, "zones":["TGT0"], "zonedddefs":[{"zone":"TGT0", "dddefs":
        ["SGIMBIN"]}]}], "msgs":null},
    "nonsmpmanageddatasets":null
  ]}

```

Figure 202. Sample get status response when the list data sets request is complete

## Add a new software instance

You can use this operation to add a software instance to z/OSMF.

### HTTP method and URI path

```
POST /zosmf/swmgmt/swi
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.

### Standard headers

Use the following standard HTTP header with this request:

#### Content-Type

Identifies the type of input content provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document included as input with this request.

#### Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

## Custom headers

None.

## Request content

Your request must include a JSON object that describes the software instance to be added, for example:

```
{
  "name": "swi-name",
  "system": "system-nickname",
  "description": "swi-description",
  "globalzone": "global-zone",
  "targetzones": ["target-zones"],
  "categories": ["categories"],
  "datasets": [
    {
      "dsname": "data-set-name",
      "volume": "volume-serial"
    }
  ],
  "products": [
    {
      "prodname": "product-name",
      "prodid": "product-id",
      "release": "product-level",
      "vendor": "vendor-name",
      "url": "product-url",
      "features": ["feature-name"],
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service-date"
    }
  ],
  "workflows": [
    {
      "name": "workflow-name",
      "description": "workflow-description",
      "location": {
        "smptype": "smp-type",
        "smpname": "smp-name",
        "dsname": "workflow-dsname",
        "path": "workflow-path"
      },
      "performonhostsyste": true | false
    }
  ],
  "datasetproperties": [
    {
      "dddefname": "dddef-name",
      "zone": "zone-name",
      "dsname": "data-set-name",
      "volume": "volume-serial",
      "dstype": "DLIB",
      "properties": [{"key": "value"}]
    }
  ],
  "datasetpropertylabels": [
    {
      "propertyname": "property-name",
      "label": "property-label"
    }
  ],
  "productproperties": [
    {
      "prodid": "product-id",
      "release": "product-level",
      "prodname": "product-name",
      "properties": [{"key": "value"}]
    }
  ]
}
```

Figure 203. Adding a software instance: request content

where:

### swi-name

Name of the software instance. The name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ \_ # @ ^). The name is required and must be unique on the system.

### system-nickname

Nickname of the system that has access to the volumes and data sets where the software instance resides. Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems defined to z/OSMF, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

**swi-description**

Description of the software instance. The description is optional and can contain a maximum of 256 characters.

**global-zone**

CSI data set that contains the global zone used to manage the software. If specified, must comply with z/OS data set naming conventions, and must end with *.CSI*.

**target-zones**

Comma-separated list of the target zones to be included in the software instance. A list of 0 or more zone names, each target zone name must be 1-7 characters long; the valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and special characters (@ # \$). The first character must be an alphabetic character.

**categories**

Comma-separated list of the categories to which the software instance is assigned. Each category name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ \_ # @ ^). Assigning the software instance to a category is optional.

**datasets**

Array that contains each non-SMP/E managed data set to be added to the software instance. Adding non-SMP/E managed data sets to the software instance is optional.

**data-set-name**

Name of the non-SMP/E managed data set. You cannot specify data set members or a subset of a data set. A data set name is required if you are adding non-SMP/E managed data sets to the software instance. The data set name must comply with z/OS data set naming conventions

**volume-serial**

Volume on which the non-SMP/E managed data set resides. The volume serial is required if the dataset is not cataloged, and the volume must be accessible by the system where the software instance resides. The volume serial must be 1-6 characters long; the valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and national characters (\$, #, @). This value is optional.

**products**

List of products for the software instance that are not managed by SMP/E. This list is optional.

**product-name**

Name of the product, but can be up to 64 characters.

**product-ID**

Identifier for the product, but can be up to 64 characters.

**product-level**

Release level for the product, but can be up to 64 characters.

**vendor-name**

Name of the vendor that provides the product, but can be up to 64 characters.

**product-URL**

A URL that links to additional information about the product, but can be up to 1023 characters.

**feature-name**

List of names of features for the product, but can be up to 64 characters.

**general-availability-date**

Date this level of the product is available to all users. May be null, or a date value, in ISO 8601 format, yyyy-mm-ddThh:mm:ssZ.

**end-of-service-date**

Last date on which the vendor will deliver standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. Can be any of the following:

**null**

The end of service date is unknown for the product.

**yyyy-mm-ddThh:mm:ssZ**

The known end of service date, in ISO 8601 format.

**NotAnnounced**

The end of service date is not yet announced for the product.

**workflows**

List of workflows for the software instance. This list is optional.

**workflow-name**

Name for the workflow. The name may contain up to 100 characters, but must not include the characters for ampersand ('&'), forward slash ('/'), backward slash ('\'), logical or ('|'), greater than ('>'), or less than ('<'). Embedded blanks are allowed.

**workflow-description**

Description for the workflow. The description is optional, and can contain a maximum of 256 characters.

**location**

Location of the workflow definition file for the workflow. A workflow definition file location is required.

**smp-type**

The SMP/E element type for a workflow definition file that is managed by SMP/E. The SMP/E element type is optional and may be up to 12 uppercase alphanumeric characters. The first character cannot be numeric.

**smp-name**

The SMP/E element name for a workflow definition file that is managed by SMP/E. The SMP/E element name is optional and may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

**workflow-dsname**

The sequential data set name, or partitioned data set name and member for a workflow definition file that is in a data set. The data set name is optional, but if specified must comply with z/OS data set naming conventions. For example, IBMUSR.SM.WRKFLows(WRKFLOW1).

**workflow-path**

The UNIX path for a workflow definition file that is a UNIX file. The UNIX path is optional, but if specified it must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash.
- Can be up to 1023 characters long.

**performonhostsystem**

Indicates whether the workflow steps may be performed on the host system or on another system in the same sysplex as the host system. This property is optional and the default value is true.

**true**

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

**false**

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

**Tip:** Ensure that the description property contains information that helps the user select an appropriate alternative system for the performance of the workflow steps. For example, if the workflow steps are for installation verification procedures (IVPs), the user would select the system on which the newly activated software runs.

**datasetproperties**

A list of one or more properties for individual data sets. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 991 for more information.

To define properties for SMP/E managed data sets, specify the DDDEF entry name that identifies the desired data set or data sets. Specify the zone name only when necessary to identify a unique data set. For example, if there are more than one DDDEF entries with the same name pointing to different data sets, then both a DDDEF entry name and zone name are required to define properties for only one of those data sets. If an SMP/E zone name is not specified then the properties apply to all data sets identified by all DDDEF entries with the same name in all zones that reside in the software instance.

To define properties for a non-SMP/E managed data set, specify the data set name and the volume for the data set as they are used when defining the data set to the software instance. That is, if only a data set name is specified to define the data set to the software instance, then specify only the data set name when defining properties for the data set. If a data set name and volume are specified to define the data set to the software instance, then specify both the data set name and volume to define properties for the data set.

**dddefname**

The name of the SMP/E DDDEF entry that describes one or more SMP/E managed data sets. It may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

**zone**

The zone name where the DDDEF entry resides. It may be up to 7 uppercase alphanumeric, \$, #, or @ characters.

**dsname**

The name of the subject data set. This is used to identify a non-SMP/E managed data set. It may be up to 44 characters and must comply with z/OS data set naming conventions.

**volume**

The volume of the subject data set. This is used to identify a non-SMP/E managed data set where the volume had been specified to identify an uncatalogued data set. It may be up to 6 uppercase alphanumeric, \$, #, or @ characters.

**dstype**

The usage type of the identified data set. The value may be DLIB or the default value of null. A value of DLIB indicates the identified data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone. The dstype value of DLIB must be specified for properties that identify distribution library or related control data sets so Software Management can safely ignore this property specification when a software instance does not contain SMP/E managed distribution libraries or related control data sets.

**properties**

A list of one or more properties for the subject data set, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

**datasetpropertylabels**

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

**propertyname**

The name, or key, of the existing provider defined property.

**label**

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

**productproperties**

A list of one or more properties for individual software products. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow

instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 991 for more information.

A prodid and release must be specified to uniquely identify an SMP/E product.

Prodname, prodid, and release may be specified to uniquely identify a non-SMP/E product.

**prodid**

The identifier for the subject product. This is required to identify an SMP/E managed product. The product identifier may be up to 64 characters.

**release**

The version, release, modification level for the subject product. This is required to identify an SMP/E managed product. The release level may be null, or up to 64 characters.

**prodname**

The name of the subject product. This is required to identify a non-SMP/E managed product. The product name may be null, or up to 64 characters.

**properties**

A list of one or more properties for the subject product, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task, and CONTROL access to the SAF resources corresponding to the software instance to be added. If the specified categories are implicitly added during this software instance add operation, the user ID must also have CONTROL access to the SAF resources corresponding to the specified categories. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 413.

## Example

In the following example, the POST method is used to add a software instance to the z/OSMF instance that has a host name of *pev174.yourco.com*.

```

POST /zosmf/swmgmt/swi HTTP/1.1
Host: pev174.yourco.com Content-Type: application/json Accept-Language: en { "name":"Hooli",
"system":"PEV174", "description":"Fictitious software product from Hooli",
"globalzone":"IBMUSER.HOOLLI.CSI", "targetzones":["TGTZ"], "datasets":
[
  { "dsname":"IBMUSER.HOOLLI.PROCLIB", "volume":"LV1234" },
  { "dsname":"IBMUSER.HOOLLI.PARMLIB", "volume":"LV1234" } ], "products":[ { "prodname":"Hooli
Express", "release":"2.1", "vendor":"Hooli Software" } ], "workflows":[ { "name":"Setup Hooli",
"description":"Setup and configure fictitious software product Hooli", "location":
"dsname":"IBMUSER.HOOLLI.WORKFLOW(SETUP)" } ], "datasetproperties":
[
  { "dddefine":"HOOLMOD", "properties":[{"hooli-LnkLst":"yes"}, {"hooli-ispfType":"LMOD"}]},
  { "dddefine":"HOOLMSG", "properties":[{"hooli-ispfType":"MESSAGE"}]},
  { "dddefine":"HOOLPNL", "properties":[{"hooli-ispfType":"PANEL"}]},
  { "dddefine":"HOOLSKL", "properties":[{"hooli-ispfType":"SKELETON"}]},
  { "dddefine":"HOOLTBL", "properties":[{"hooli-ispfType":"TABLE"}]},
  { "dddefine":"HOOLREXX", "properties":[{"hooli-ispfType":"EXEC"}]},
  { "propertyname":"hooli-LnkLst", "label":"LNKLST Eligible", "datasetpropertylabels":
  { "propertyname":"hooli-ispfType", "label":"ISPF Element Type" } ] }

```

Figure 204. Sample request to add a software instance

## Export a defined software instance

A portable software instance is a set of portable files that represents the content of a z/OSMF software instance. An Export action on a software instance is used to create a portable software instance. You can use the POST method to perform an Export action on a software instance that is defined to z/OSMF, which generates a portable software instance descriptor file and JCL that when executed creates the archive files for a portable software instance, and store those files in a UNIX directory on the system where the software instance being exported resides.

### HTTP method and URI path

```
POST /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/export
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be exported. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

- **/export** indicates JCL to perform an export action for the software instance to be generated.

### Standard headers

Use the following standard HTTP headers with this request:

#### Accept-Language

Identifies the preferred language for messages that can be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, then English is used.

#### Content-Type

Identifies the type of input content that is provided by the caller. The JSON content type("Content-Type: application/json") is used for the JSON document, if any, included as input with this request.

## Custom headers

None.

## Request content

The request must include a JSON document that identifies properties that are required to perform the operation. For example:

```
{
  "packagedir": "UNIX-directory",
  "jcldataset": "data-set-name",
  "includedlibs": "yes | no",
  "jobstatement": ["jclrecord"],
  "unixdatasets": [{
    "dsname": "data-set-name",
    "mountpoint": "UNIX-path"
  }],
  "workdsnprefix": "data-set-name-prefix",
  "workvolume": "volume-serial",
  "workstorclas": "storage-class",
  "zosmfuid": "user-id",
  "zosmfpw": "password",
  "proxyuid": "user-id",
  "proxypw": "password"
}
```

where:

### **packagedir**

UNIX directory to contain the files for the portable software instance.

Must be a UNIX directory with valid UNIX directory name syntax:

- Must be an absolute pathname (must start with slash).
- Must end with a slash.
- Can be up to 1023 characters long.

### **jcldataset**

Partitioned data set to contain the portable software instance descriptor file and the generated JCL.

The data set name must comply with z/OS data set naming conventions.

### **includedlibs**

If the software instance contains SMP/E managed software, this property indicates whether the distribution libraries and DLIB zones are to be included in the portable software instance.

This is an optional property. Can be null, "yes" or "no", but the default value is "yes".

### **jobstatement**

List of JCL cards for the JOB statement to be used in the generated JCL for the export operation.

This is an optional property. Can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"\* and the job name must be 1 to 8 characters. If no JOB statement is provided, the default is exactly://IZUD01EX JOB (ACCOUNT),NAME'.

### **unixdatasets**

List of UNIX file system data sets in the software instance that are currently not mounted, and therefore, cannot be identified by z/OSMF by referencing the UNIX directories that are defined in the SMP/E target zone DDDEF entries.

This is an optional property, required only if the software instance describes SMP/E managed software, and if any of the UNIX file system data sets containing that software are currently not mounted.

### **dsname**

Name of a UNIX file system data set.

The data set name must comply with z/OS data set naming conventions.

**mountpoint**

Mount point for the UNIX file system data set, if the data set were mounted.

Must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash, unless a slash is the only character. That is, "/" (root) is valid, but "/mountpoint/" is not.
- Can be up to 1023 characters long.

**workdsnprefix**

Data set name prefix for the work data sets that are created, used, and then deleted in the generated Export JCL.

z/OSMF uses a work zFS data set to contain temporary UNIX directory space. The prefix can contain up to 26 characters, including periods, and must comply with z/OS data set naming conventions. This value is optional.

**Notes:**

- If one or more of the workdsnprefix, workvolume, or workstorclas values are specified, then the generated JCL creates, mounts, uses, unmounts, and deletes a work zFS data set. If none of those three values are specified, then the generated JCL does not create and use a work zFS data set and instead uses /tmp as the temporary UNIX directory space.
- If you specify workdsnprefix but do not provide values for either workvolume or workstorclas, then the volume and storclas are not specified in the generated JCL on the IDCAMS DEFINE CLUSTER command to create the zFS data set.
- If workdsnprefix is not specified but either workvolume or workstorclas is provided, then a default data set name prefix of *z/OSMF-ID.WRKDSN* is used, where *z/OSMF-ID* is the logged-in user ID that is used to authenticate to the z/OSMF server.

**workvolume**

Volume for the work data sets that are created, used, and then deleted in the generated Export JCL.

The volume serial value must contain 1-6 characters. Valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and national characters (\$, #, @). This value is optional.

**Notes:**

- See [“workdsnprefix” on page 436](#) for notes that also apply here.
- The workvolume and workstorclas properties are mutually exclusive.

**workstorclas**

Storage class for the work data sets that are created, used, and then deleted in the generated Export JCL.

The storage class value can contain up to eight characters. Valid characters are alphabetic characters (A-Z, a-z), numeric characters (0-9), and national characters (\$, #, @). This value is optional.

**Notes:**

- See [“workdsnprefix” on page 436](#) for notes that also apply here.
- The workvolume and workstorclas properties are mutually exclusive.

**zosmfuid**

Userid for authenticating with a remote z/OSMF instance.

This is an optional property.

**zosmfpw**

Password for authenticating with a remote z/OSMF instance.

This is an optional property

**proxyuid**

Userid for authenticating with an HTTP proxy.

This is an optional property.

#### **proxypw**

Password for authenticating with an HTTP proxy.

This is an optional property.

The request content is required, but some properties are optional. For example, if the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. Therefore, you can need to specify the remote z/OSMF userid, password, and proxy userid and password.

## **Required authorizations**

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation that uses the z/OSMF Software Management task. That is, to export a software instance, the user ID must have READ access to the Software Management task and READ access to the SAF resources for the software instance being exported. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## **Expected response**

Generating JCL to export a software instance is an asynchronous operation. Therefore, on completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted and a JSON document containing a URL for the status monitor for the request. The client performs GET requests to the supplied URL to monitor the status of the operation and to obtain the result set. For example:

```
{
  "statusurl": "url"
}
```

where:

#### **statusurl**

Indicates the URL that provides status for the software instance export request.

On subsequent GET requests to the status monitor URL:

- If the operation is not yet complete, an HTTP response code of 200 OK is returned, along with a JSON document containing status information for the operation.
- If the operation has completed, then an HTTP response code of 200 OK is returned, along with a JSON document containing status information and the desired result set.
- If the request has expired, then an HTTP response code of 404 Not found is returned. That is, when the operation has completed, the z/OSMF server holds the result set for a finite length of time. After that time has passed, the result set is said to expire and will no longer be available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status",
  "percentcomplete": "percent",
  "jcl": ["data-set-name(member-name)"]
}
```

**status**

The status of the export request. The status can be either "running" or "complete".

**percentcomplete**

The percentage of the processing that is complete for this export software instance request, expressed as a whole number from 0 to100.

**jcl**

Ordered list of generated jobs for the export action. Each job is a unique member in the JCL data set. The values are fully qualified data set and member names of the form "data-set-name(member-name)".

See “Error handling” on page 413 for the error response document containing a reason code and a list of one or more message strings to describe the errors that are detected during processing of a request.

**Example**

The following request generates an Export job for the software instance that is named *DB2V9* on *SYS123*.

```
POST /zosmf/swmgmt/swi/SYS123/DB2V9/export HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "packagedir": "/u/userid/exportDir/",
  "jcldataset": "USERID.SMJCL.CNTL",
  "includedlibs": "yes",
  "jobstatement": ["//EXPORT    JOB (123456), 'USER', NOTIFY=&SYSUID,",
  "// MSGCLAS=H", "/*"]
}
```

*Figure 205. Sample request*

A sample response is as follows:

```
HTTP/1.1 202 Accepted
Date: Tues, 21 November 2014 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl": "https://sys123.yourco.com/zosmf/swmgmt/statusmonitor/export/4837290198343"}
```

*Figure 206. Sample response*

The above response indicates the request to generate an Export job for the software instance has been accepted, and the status monitor URL is provided. A subsequent GET request to the status monitor URL is as follows:

```
GET /zosmf/swmgmt/statusmonitor/export/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

A sample response is as follows:

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:19 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status": "running", "percentcomplete": "65"}
```

*Figure 207. Sample response*

The above response indicates the operation to generate an Export job for the software instance is still running and 65% complete. A final request to the status monitor URL is as follows:

```
GET /zosmf/swmgmt/statusmonitor/export/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

A sample response is as follows:

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:27 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete", "percentcomplete":"100",
"jcl":["USERID.SMJCL.CNTL(IZUD01EX)"]}
}
```

*Figure 208. Sample response*

## Usage notes

The POST method to generate JCL that exports a software instance requires the subject software instance to be already defined to z/OSMF. If the software instance is not already defined, you can use the POST method to Add a new software instance to z/OSMF, followed by the PUT method to Load the products, features, and FMIDs for the new software instance if that software instance contains SMP/E managed software. Then, you can use the POST method to Export the software instance. For example:

Add the software instance:

```
POST /zosmf/swmgmt/swi HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "name": "DB2V9",
  "system": "sys123",
  "description": "DB2",
  "globalzone": "DB2.GLOBAL.CSI",
  "targetzones": ["DB2TGT"],
}
```

Load the products, features, and FMIDs for the software instance:

```
PUT /zosmf/swmgmt/swi/sys123/DB2/products HTTP/1.1
Host: sys123.yourco.com
```

Create JCL to export the software instance:

```
POST /zosmf/swmgmt/swi/sys123/DB2/export HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
Accept-Language: en
{
  "packagedir": "/u/userid/exportDir/",
  "jcldataset": "USERID.SMJCL.CNTL",
  "includedlibs": "yes",
  "jobstatement": ["//EXPORT JOB (123456), 'USER', NOTIFY=&SYSUID, ",
  "// MSGCLAS=H", "/*"]
}
```

The response from the POST method to create JCL to export the software instance specifies the members in the JCL data set that contain the generated jobs to perform the export action. You can use the z/OS jobs REST interface to submit and obtain the status of the export job. For example:

Submit the export job:

```
PUT /zosmf/restjobs/jobs HTTP/1.1
Host: sys123.yourco.com
Content-Type: application/json
{
```

```
"file": "'/USERID.SMJCL.CNTL(IZUD01EX)'"
}
```

The response to this PUT method provides the job ID and the job name for the submitted job.

Obtain status for the export job:

```
GET /zosmf/restjobs/jobs/IZUD01EX/job-id HTTP/1.1
Host: sys123.yourco.com
```

## Sample REXX exec

A sample REXX exec that is named IZUDXEXP is provided in the SYS1.SAMPLIB data set to illustrate a program that uses Software management services to do the following:

- Add a software instance.
- Load the SMP/E Products, Features, and FMIDs for the software instance.
- Generate JCL to export the software instance.
- Run the generated JCL to create a portable software instance.

The sample REXX exec uses the HTTP REXX client of the z/OS Web Enablement Toolkit to perform HTTP operations to the z/OSMF server.

## Modify the properties of a software instance

You can use this operation to modify the properties of a software instance that is defined in z/OSMF, including changing the global zone, target zones, or non-SMP/E managed data sets associated with the software instance. The modify operation updates only the definition of the software instance in z/OSMF. The physical data sets that compose the software instance are not affected.

### HTTP method and URI path

---

```
PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
```

---

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be modified. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

### Standard headers

Use the following standard HTTP header with this request:

#### Content-Type

Identifies the type of input content provided by the caller. The JSON content type ("Content-Type: application/json") is used for the JSON document included as input with this request.

#### Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

## Custom headers

None.

## Request content

Your request must include a JSON object that describes all the properties of the software instance to be modified, for example:

```
{
  "name": "swi-name",
  "system": "system-nickname",
  "description": "swi-description",
  "globalzone": "global-zone",
  "targetzones": ["target-zones"],
  "categories": ["categories"],
  "datasets": [
    {
      "dsname": "data-set-name",
      "volume": "volume-serial"
    }
  ]
  "products": [{
    "prodname": "product-name",
    "prodid": "product-id",
    "release": "product-level",
    "vendor": "vendor-name",
    "url": "product-url",
    "features": ["feature-name"],
    "generalavailability": "general-availability-date",
    "endofservice": "end-of-service-date"
  }],
  "workflows": [{
    "name": "workflow-name",
    "description": "workflow-description",
    "location": {
      "smptype": "smp-type",
      "smpname": "smp-name",
      "dsname": "workflow-dsname",
      "path": "workflow-path"
    },
    "performonhostsyste": true | false
  }],
  "datasetproperties": [{
    "dddefname": "dddef-name",
    "zone": "zone-name",
    "dsname": "data-set-name",
    "volume": "volume-serial",
    "dstype": "DLIB",
    "properties": [{"key": "value"}]
  }],
  "datasetpropertylabels": [{
    "propertyname": "property-name",
    "label": "property-label"
  }],
  "productproperties": [{
    "prodid": "product-id",
    "release": "product-level",
    "prodname": "product-name",
    "properties": [{"key": "value"}]
  }],
}
```

Figure 209. Modifying a software instance: request content

where:

### swi-name

Name of the software instance. The name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ \_ # @ ^). The name is required and must be unique on the system.

### system-nickname

Nickname of the system that has access to the volumes and data sets where the software instance resides. Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems defined to z/OSMF, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

**swi-description**

Description of the software instance. The description is optional and can contain a maximum of 256 characters.

**global-zone**

CSI data set that contains the global zone used to manage the software. The data set name is required, must comply with z/OS data set naming conventions, and must end with *.CSI*.

**target-zones**

Comma-separated list of the target zones to be included in the software instance. At least one target zone is required. Each target zone name must be 1-7 characters long; the valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), and special characters (@ # \$). The first character must be an alphabetic character.

**categories**

Comma-separated list of the categories to which the software instance is assigned. Each category name can contain up to 30 non-blank characters, including alphanumeric characters (A-Z, a-z, and 0-9), mathematical symbols (< > - = | \), punctuation marks (? ! : ' " /), and special characters (\$ \_ # @ ^). Assigning the software instance to a category is optional.

**datasets**

Array that contains each non-SMP/E managed data set to be added to the software instance. Adding non-SMP/E managed data sets to the software instance is optional.

**data-set-name**

Name of the non-SMP/E managed data set. You cannot specify data set members or a subset of a data set. A data set name is required if you are adding non-SMP/E managed data sets to the software instance. The data set name must comply with z/OS data set naming conventions

**volume-serial**

Volume on which the non-SMP/E managed data set resides. The volume serial is required if the data set is not cataloged, and the volume must be accessible by the system where the software instance resides. The volume serial must be 6 characters long; the valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9).

**products**

List of products for the software instance that are not managed by SMP/E. This list is optional.

**product-name**

Name of the product, but can be up to 64 characters.

**product-ID**

Identifier for the product, but can be up to 64 characters.

**product-level**

Release level for the product, but can be up to 64 characters.

**vendor-name**

Name of the vendor that provides the product, but can be up to 64 characters.

**product-URL**

A URL that links to additional information about the product, but can be up to 1023 characters.

**feature-name**

List of names of features for the product, but can be up to 64 characters.

**general-availability-date**

Date this level of the product is available to all users. May be null, or a date value, in ISO 8601 format, yyyy-mm-ddThh:mm:ssZ.

**end-of-service-date**

Last date on which the vendor will deliver standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. Can be any of the following:

**null**

The end of service date is unknown for the product.

**yyyy-mm-ddThh:mm:ssZ**

The known end of service date, in ISO 8601 format.

**NotAnnounced**

The end of service date is not yet announced for the product.

**workflows**

List of workflows for the software instance. This list is optional.

**workflow-name**

Name for the workflow. The name may contain up to 100 characters, but must not include the characters for ampersand ('&'), forward slash ('/'), backward slash ('\'), logical or ('|'), greater than ('>'), or less than ('<'). Embedded blanks are allowed.

**workflow-description**

Description for the workflow. The description is optional, and can contain a maximum of 256 characters.

**location**

Location of the workflow definition file for the workflow. A workflow definition file location is required.

**smp-type**

The SMP/E element type for a workflow definition file that is managed by SMP/E. The SMP/E element type is optional and may be up to 12 uppercase alphanumeric characters. The first character cannot be numeric.

**smp-name**

The SMP/E element name for a workflow definition file that is managed by SMP/E. The SMP/E element name is optional and may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

**workflow-dsname**

The sequential data set name, or partitioned data set name and member for a workflow definition file that is in a data set. The data set name is optional, but if specified must comply with z/OS data set naming conventions. For example, IBMUSR.SM.WRKFLWS(WRKFLOW1).

**workflow-path**

The UNIX path for a workflow definition file that is a UNIX file. The UNIX path is optional, but if specified it must have valid UNIX file name syntax:

- Must be an absolute pathname (must start with slash).
- Must not end with a slash.
- Can be up to 1023 characters long.

**performonhostsystem**

Indicates whether the workflow steps may be performed on the host system or on another system in the same sysplex as the host system. This property is optional and the default value is true.

**true**

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

**false**

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

**Tip:** Ensure that the description property contains information that helps the user select an appropriate alternative system for the performance of the workflow steps. For example, if the workflow steps are for installation verification procedures (IVPs), the user would select the system on which the newly activated software runs.

**datasetproperties**

A list of one or more properties for individual data sets. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 991 for more information.

To define properties for SMP/E managed data sets, specify the DDDEF entry name that identifies the desired data set or data sets. Specify the zone name only when necessary to identify a unique data set. For example, if there are more than one DDDEF entries with the same name pointing to different data sets, then both a DDDEF entry name and zone name are required to define properties for only one of those data sets. If an SMP/E zone name is not specified then the properties apply to all data sets identified by all DDDEF entries with the same name in all zones that reside in the software instance.

To define properties for a non-SMP/E managed data set, specify the data set name and the volume for the data set as they are used when defining the data set to the software instance. That is, if only a data set name is specified to define the data set to the software instance, then specify only the data set name when defining properties for the data set. If a data set name and volume are specified to define the data set to the software instance, then specify both the data set name and volume to define properties for the data set.

**dddefname**

The name of the SMP/E DDDEF entry that describes one or more SMP/E managed data sets. It may be up to 8 uppercase alphanumeric, \$, #, or @ characters.

**zone**

The zone name where the DDDEF entry resides. It may be up to 7 uppercase alphanumeric, \$, #, or @ characters.

**dsname**

The name of the subject data set. This is used to identify a non-SMP/E managed data set. It may be up to 44 characters and must comply with z/OS data set naming conventions.

**volume**

The volume of the subject data set. This is used to identify a non-SMP/E managed data set where the volume had been specified to identify an uncatalogued data set. It may be up to 6 uppercase alphanumeric, \$, #, or @ characters.

**dstype**

The usage type of the identified data set. The value may be DLIB or the default value of null. A value of DLIB indicates the identified data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone. The dstype value of DLIB must be specified for properties that identify distribution library or related control data sets so Software Management can safely ignore this property specification when a software instance does not contain SMP/E managed distribution libraries or related control data sets.

**properties**

A list of one or more properties for the subject data set, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

**datasetpropertylabels**

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

**propertyname**

The name, or key, of the existing provider defined property.

**label**

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

**productproperties**

A list of one or more properties for individual software products. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow

instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 991 for more information.

A prodid and release must be specified to uniquely identify an SMP/E product.

Prodname, prodid, and release may be specified to uniquely identify a non-SMP/E product.

**prodid**

The identifier for the subject product. This is required to identify an SMP/E managed product. The product identifier may be up to 64 characters.

**release**

The version, release, modification level for the subject product. This is required to identify an SMP/E managed product. The release level may be null, or up to 64 characters.

**prodname**

The name of the subject product. This is required to identify a non-SMP/E managed product. The product name may be null, or up to 64 characters.

**properties**

A list of one or more properties for the subject product, specified as key-value pairs. The keys are strings, and values are a valid JSON data type such as string, number, Boolean, array, object or null. Keys may not start with *izud-* to ensure no collisions with Software Management provided variables and properties.

## Usage considerations

The SMP/E global zone and target zones define the installed products, features, and FMIDs for a software instance. If you modify these properties for a software instance, perform a PUT method to load the products, features, and FMIDs for the updated software instance. Doing so refreshes the product, feature, and FMID information known to z/OSMF for the subject software instance. For instructions, see [“Load the products, features, and FMIDs for a software instance”](#) on page 446.

For other usage considerations, see [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

To submit a modify software instance request through the software management services, the user ID initiating the request requires the same authorizations as when performing a modify operation using the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task. The user ID must also have either CONTROL or UPDATE access to the SAF resources corresponding to the software instance to be modified, as follows:

- The name, system, and categories properties are used to create the SAF resource names for the software instance; therefore, to modify any of these properties, the user ID must have CONTROL access to the existing SAF resource names and to the new SAF resource names.
- If a global zone or categories will be implicitly added to z/OSMF through the modify operation, the user ID must have CONTROL access to the SAF resources corresponding to the specified global zone and categories.
- To modify any other property, the user ID must have UPDATE access.

For information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 413.

## Example

In the following example, the PUT method is used to modify the description property for software instance *Hooli* on system *PEV174*.

```
PUT /zosmf/swmgmt/swi/PEV174/Hooli HTTP/1.1
Host: pev174.yourco.com Content-Type: application/json Accept-Language: en { "name": "Hooli",
"system": "PEV174", "description": "Fictitious software product from Hooli",
"globalzone": "IBMUSER.HOOLI.CSI", "targetzones": ["TGTZ"], "datasets":
[
  { "dsname": "IBMUSER.HOOLI.PROCLIB", "volume": "LV1234" },
  { "dsname": "IBMUSER.HOOLI.PARMLIB", "volume": "LV1234" } ], "products": [
  { "prodname": "Hooli Express", "release": "2.1", "vendor": "Hooli Software" } ], "workflows": [
  { "name": "Setup Hooli", "location": "Setup and configure fictitious software product Hooli",
    "description": "Setup and configure fictitious software product Hooli",
    "dsname": "IBMUSER.HOOLI.WORKFLOW(SETUP)" } ] }, "datasetproperties":
[
  { "dddefname": "HOOLMOD", "properties": [ { "hooli-lnklst": "yes" }, { "hooli-ispfType": "LMOD" } ] },
  { "dddefname": "HOOLMSG", "properties": [ { "hooli-ispfType": "MESSAGE" } ] },
  { "dddefname": "HOOLPNL", "properties": [ { "hooli-ispfType": "PANEL" } ] },
  { "dddefname": "HOOLSKL", "properties": [ { "hooli-ispfType": "SKELETON" } ] },
  { "dddefname": "HOOLTBL", "properties": [ { "hooli-ispfType": "TABLE" } ] },
  { "dddefname": "HOOLREXX", "properties": [ { "hooli-ispfType": "EXEC" } ] } ], "datasetpropertylabels":
[
  { "propertyname": "hooli-lnklst", "label": "LNKLST Eligible" },
  { "propertyname": "hooli-ispfType", "label": "ISPF Element Type" } ] }
```

Figure 210. Sample request to modify a software instance

Figure 211 on page 446 provides a sample response, indicating that the update was successful.

```
HTTP/1.1 200 OK
Date: Tues, 22 July 2014 18:53:27 +0000GMT
```

Figure 211. Sample response for a modify software instance request

## Load the products, features, and FMIDs for a software instance

You can use this operation to analyze the SMP/E global zone and target zones for a software instance to identify the installed products, features, and FMIDs in the instance, and to load that information into the z/OSMF Software Management task database.

### HTTP method and URI path

```
PUT /zosmf/swmgmt/swi/<system-nickname>/<swi-name>/products
```

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be retrieved. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

- **products** indicates that the products, features, and FMIDs included in the software instance are to be obtained and loaded into the Software Management task database.

When you issue this request, z/OSMF searches the SMP/E global zone and target zones associated with the software instance and gathers information about the software instance and its products, features, and FMIDs. z/OSMF returns this information in a JSON object, and stores this information in the Software Management task database.

## Standard headers

Use the following standard HTTP header with this request:

### Content-Type

Identifies the type of input content provided by the caller. Use the JSON content type ("Content-Type: application/json") if a JSON document is included as input with this request.

### Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

## Custom headers

None.

## Request content

If the software instance does not reside in the same sysplex as the primary z/OSMF instance, you might be required to authenticate with the secondary z/OSMF instance that is running in the sysplex where the software instance resides. In addition, if the primary z/OSMF instance must navigate an HTTP proxy server to connect with the secondary z/OSMF instance, you might also be required to authenticate with that HTTP proxy server. To do so, include the following JSON object in your request:

```
{
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Figure 212. Request content to authenticate with a secondary z/OSMF instance and an HTTP proxy server

where:

### zosmf-user-ID

User ID for authenticating with the secondary z/OSMF instance.

### zosmf-password

Password for authenticating with the secondary z/OSMF instance.

### proxy-user-ID

User ID for authenticating with the HTTP proxy server.

### proxy-password

Password for authenticating with the HTTP proxy server.

Include the JSON object in the request only if you are required to authenticate with a secondary z/OSMF instance or an HTTP proxy server. Otherwise, omit the JSON object.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

To submit requests through the software management services, the user ID initiating the request requires the same authorizations as when performing an analogous operation using the z/OSMF Software Management task. That is, to retrieve the product, feature, and FMID information, the user ID must have READ access to the Software Management task and UPDATE access to the SAF resources for the software instance being updated. For information about access controls for the Software Management task, see

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request was accepted. If the request was accepted, the service returns status code 202 Accepted and a JSON object that contains a URL (`{"statusurl": "url"}`). To monitor the status of the retrieve product, feature, and FMID information request and to obtain the results, perform GET requests to the supplied URL. Only the user ID that initiates the retrieve product, feature, and FMID information request is authorized to check the status and obtain the results. One of the following responses is returned from the get status request:

- If the retrieve product, feature, and FMID information request is still in progress, an HTTP response code of 200 OK is returned, along with the following JSON object: `{"status": "status"}`.
- If the retrieve product, feature, and FMID information request is complete, an HTTP response code of 200 OK is returned, along with the following JSON object:

```
{
  "status": "status",
  "swi": {
    "name": "swi-name",
    "system": "system-nickname",
    "description": "swi-description",
    "globalzone": "global-zone-name",
    "targetzones": ["target-zone-name"],
    "categories": ["category-name"],
    "productinfo": {"retrieved": "date-retrieved",
                    "lastmodified": "date-modified",
                    "modifiedby": "modified-user-ID",
                    "created": "date-created",
                    "createdby": "created-user-ID",
                    "locked": "date-locked",
                    "lockedby": "locked-user-ID",
                    "datasets": [{
                      "dsname": "data-set-name", "volume": "volume-serial"
                    }
                    ]
    },
    "products": [{
      "prodname": "product-name",
      "prodid": "product-ID",
      "release": "version-release-modification",
      "vendor": "vendor-name",
      "generalavailability": "general-availability-date",
      "endofservice": "end-of-service-date",
      "url": "product-URL",
      "productinfo": {"fileversion": "product-file-version",
                      "features": [{
                        "feature": "feature-name",
                        "fmids": [{
                          "fmid": "fmid-name",
                          "description": "fmid-description",
                          "targetzones": ["fmid-target-zone-name"]
                        }
                        ]
                      }
                      ]
    }
    ]
  }
}
```

where:

### status

Status of the retrieve product, feature, and FMID information request. The status is either *running* or *complete*.

### swi

JSON object to describe a software instance.

### swi-name

Name of the software instance.

**system-nickname**

Nickname of the z/OSMF host system that has access to the global zone CSI data set included in the software instance. To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

**swi-description**

Description of the software instance.

**global-zone-name**

Name of the CSI data set that contains the global zone used to manage the software.

**target-zone-name**

Array of the target zones in the specified global zone that describe the software.

**category-name**

Array of the categories to which the software instance is assigned.

**date-retrieved**

Date and time in ISO 8601 format that the product, feature, and FMID information for the software instance was last retrieved from the CSI data set. For example, 2014-08-20T19:23:25+00:00Z.

**date-modified**

Date and time in ISO 8601 format that the software instance was last modified. For example, 2014-08-20T19:23:25+00:00Z.

**modified-user-ID**

User ID of the user who last modified the software instance.

**date-created**

Date and time in ISO 8601 format that the software instance was created. For example, 2014-08-20T19:23:25+00:00Z.

**created-user-ID**

User ID of the user who created the software instance.

**date-locked**

Date and time in ISO 8601 format that the software instance was locked for an impending update. For example, 2014-08-20T19:23:25+00:00Z. If null, the software instance is not locked.

**locked-user-ID**

User ID of the user who locked the software instance.

**datasets**

Array of the non-SMP/E managed data sets that are included in the software instance.

**data-set-name**

Name of the data set.

**volume-serial**

Volume serial for the volume where the data set resides.

**products**

Array of the products included in the software instance.

**product-name**

Name of the product.

**product-ID**

Identifier for the product.

**version-release-modification**

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level.

**vendor-name**

Name of the vendor that provides the product.

**general-availability-date**

Date and time in ISO 8601 format that a version or release of the product became available to all users. For example, 2014-08-20T19:23:25+00:00Z.

**end-of-service-date**

Date and time in ISO 8601 format that service support ends for the product. For example, 2014-08-20T19:23:25+00:00Z.

**product-URL**

URL that links to additional information about the product. This information can include, for example, product life cycle dates, product highlights, planning information, and technical descriptions.

**product-file-version**

Version of the most recent product information file that supplied information about the product. The version represents the date and time in ISO 8601 format that file was created or last updated. For example, 2014-08-20T19:23:25+00:00Z.

**features**

Array of the features contained in the software instance.

**feature-name**

Name of the feature.

**fmids**

Array of the FMIDs contained in the software instance.

**fmid-name**

Name of the FMID.

**fmid-description**

Description of the FMID.

**fmid-target-zone-name**

Array of the target zones where the FMID is installed.

- If the retrieve product, feature, and FMID information request is complete but the results are no longer available, an HTTP response code of 404 Not found is returned. z/OSMF makes the results available for a client application for a finite period of time. When that time elapses, the results are no longer available; in which case, the client must reissue the request.

If the retrieve product, feature, and FMID information request cannot be processed, a status code of 4nn or 5nn is returned, indicating that an error has occurred. For more details, see [“Error handling” on page 413](#).

**Example**

In the following example, the PUT method is used to retrieve the product, feature, and FMID information for software instance *DB2V9* on system *SYS123*.

```
PUT /zosmf/swmgmt/swi/SYS123/DB2V9/products HTTP/1.1
Host: sys123.yourco.com
```

*Figure 213. Sample request to retrieve the product, feature, and FMID information for a software instance*

Figure 214 on page 451 provides a sample response, indicating that the retrieve product, feature, and FMID information request has been accepted and supplying the URL to use for monitoring the status of that request.

```
HTTP/1.1 202 Accepted
Date: Tues, 21 November 2014 18:53:04 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://sys123.yourco.com/zosmf/swmgt/statusmonitor/prodload
/4837290198343"}
```

*Figure 214. Sample response for a retrieve product, feature, and FMID information request*

To check the status of the retrieve product, feature, and FMID information request, submit the following request:

```
GET /zosmf/swmgt/statusmonitor/prodload/4837290198343 HTTP/1.1
Host: sys123.yourco.com
```

*Figure 215. Sample request to obtain the status of a retrieve product, feature, and FMID information request*

Figure 216 on page 451 provides a sample get status response, indicating that the retrieve product, feature, and FMID information request is in progress.

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:19 +00005GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"running"}
```

*Figure 216. Sample get status response when the retrieve product, feature, and FMID information request is in progress*

Figure 217 on page 451 provides a sample get status response, indicating that the retrieve product, feature, and FMID information request is complete.

```
HTTP/1.1 200 OK
Date: Tues, 21 November 2014 18:53:36 +00006GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete", "swi":{"
  "name":"DB2V9", "system":"PEV174", "description":null,
  "globalzone":"DB2.GLOBAL.CSI", "targetzones":["DB2TGT"], "categories":null,
  "productinfo":{"retrieved":"2014-08-20T19:23:25Z", "lastmodified":"2014-08-
20T19:23:25Z", "modifiedby":"FRED", "created":"2014-08-20T19:23:25Z",
  "createdby":"BARNEY", "locked":null, "lockedby":null, "datasets":
  [{"dsname":"USER.DB2V9.PROCLIB", "volume":"LV1234"},
  {"dsname":"USER.DB2V9.SAMPLES", "volume":"LV1234"}]}, "products":
  [{"prodname":"DB2 for z/OS", "prodid":"5635-DB2", "release":"09.01.00",
  "vendor":"IBM", "generalavailability":"20006-06-09T19:23:25Z",
  "endofservice":"2014-06-27T19:23:25Z", "url":null,
  "productinfofileversion":"2014-01-01", "features":[{"feature":"DB2 Base",
  "fmids":[{"fmid":"HDB9910", "description":"DB2 BASE/TSO", "targetzones":
  ["DB2V9T"]}]}]}]}
```

*Figure 217. Sample get status response when the retrieve product, feature, and FMID information request is complete*

## Delete a software instance

You can use this operation to remove a software instance definition from z/OSMF. The delete operation removes only the definition of the software instance from z/OSMF. The physical data sets that compose the software instance are not affected.

### HTTP method and URI path

---

```
DELETE /zosmf/swmgmt/swi/<system-nickname>/<swi-name>
```

---

where:

- **zosmf/swmgmt** identifies the software management services.
- **swi** informs the service that the request is for the software instance object.
- **<system-nickname>/<swi-name>** further qualifies the request and indicates the specific software instance to be deleted. A software instance is uniquely identified by its name (*swi-name*) and the nickname (*system-nickname*) of the z/OSMF host system that has access to the volumes and data sets where the software instance resides.

To obtain information about the specified system, you can use the z/OSMF topology services. For more details, see [“Topology services” on page 465](#).

### Standard headers

Use the following standard HTTP header with this request:

#### Accept-Language

Identifies the preferred language for messages that may be returned to the caller. Acceptable values are "Accept-Language: en" (English) and "Accept-Language: ja" (Japanese). Any other language value is ignored and English is used instead. In addition, if the header is not specified, English is used.

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### Required authorizations

To submit a delete software instance request through the software management services, the user ID initiating the request requires the same authorizations as when performing a remove operation using the z/OSMF Software Management task. That is, the user ID must have READ access to the Software Management task, and CONTROL access to the SAF resources corresponding to the software instance to be deleted. For information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

### Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 413](#).

## Example

In the following example, the DELETE method is used to delete software instance *DB2V9* on system *PEV174*.

```
DELETE /zosmf/swmgmt/swi/PEV174/DB2V9 HTTP/1.1
Host: pev174.yourco.com
```

*Figure 218. Sample request to delete a software instance*

Figure 219 on page 453 provides a sample response, indicating that the delete operation was successful.

```
HTTP/1.1 200 OK
Date: Tues, 22 July 2014 18:53:27 +0000GMT
```

*Figure 219. Sample response for a delete software instance request*

## List the portable software instances defined to z/OSMF

You can use this operation to obtain a list of the portable software instances that are defined to a z/OSMF instance.

### HTTP method and URI path

```
GET /zosmf/swmgmt/pswi
```

where:

#### **zosmf/swmgmt**

Identifies the software management services.

#### **pswi**

Informs the service that the request is for a portable software instance object.

### Standard headers

Use the following standard HTTP header with this request:

#### **Accept-Language**

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

Certain authorizations are required to submit a request through the software management services to list portable software instances. The user ID that initiates the request requires the same authorizations as the ones needed to perform a list operation with the z/OSMF Software Management task. The user ID needs READ access to the Software Management task. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

## Expected response

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more information about errors, see [“Error handling” on page 413](#).

If the request is successful, the response also includes the following JSON object:

```
{
  "pswilst": [
    {
      "name": "pswi-name",
      "system": "system-nickname",
      "description": "pswi-description",
      "directory": "UNIX-directory",
      "categories": ["categories"],
      "lastmodified": "last-modified",
      "modifiedby": "modified-user-ID",
      "created": "date-created",
      "createdby": "created-user-ID",
      "locked": "date-locked",
      "lockedby": "locked-user-ID"
    }
  ]
}
```

where:

### **pswilst**

Array that contains each portable software instance that is defined to z/OSMF.

#### **pswi-name**

Name of the portable software instance.

#### **system-nickname**

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 465](#).

#### **pswi-description**

Description of the portable software instance.

#### **UNIX-directory**

UNIX directory that contains the portable software instance files.

#### **categories**

Comma-separated list of the categories to which the portable software instance is assigned.

#### **last-modified**

Date and time, in ISO 8601 format, that the portable software instance was last modified.

#### **modified-user-ID**

User ID of the user who last modified the portable software instance.

#### **date-created**

Date and time, in ISO 8601 format, that the portable software instance was created.

#### **created-user-ID**

User ID of the user who created the portable software instance.

**date-locked**

Date and time, in ISO 8601 format, that the portable software instance was locked. This attribute is null if the portable software instance is not currently locked.

**locked-user-ID**

User ID of the user who locked the portable software instance. This attribute is null if the portable software instance is not currently locked.

**Example**

In the following example, the GET method is used to retrieve a list of the portable software instances that are defined on the z/OSMF host system zosmf1.yourco.com.

```
GET /zosmf/swmgmt/pswi HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 220. Sample request to retrieve a list of portable software instances*

A sample response is shown in [Figure 221 on page 455](#).

```
HTTP/1.1 200 OK
Date: Tues, 05 June 2020 18:53:27 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close

{"pswlist":[{"name":"DB2V9", "system":"PEV174", "description":null,
"directory":"/u/zosmfuser/DB2V9", "categories":null,
"lastmodified":"2020-06-01T19:23:25+00:00", "modifiedby":"FRED",
"created":"2020-01-03T12:00:10+00:00", "createdby":"BARNEY", "locked":null,
"lockedby":null}]}
```

*Figure 221. Sample response from a request to retrieve a list of portable software Required authorizations*

**Retrieve the properties of a portable software instance**

You can use this operation to retrieve the properties of a portable software instance.

**HTTP method and URI path**

```
GET /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>
```

where:

**zosmf/swmgmt**

Identifies the software management services.

**pswi**

Informs the service that the request is for a portable software instance object.

**<system-nickname>**

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located. In combination with *<pswi-name>*, it further qualifies the request and indicates the specific portable software instance to retrieve.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 465](#).

### **<pswi-name>**

Name of the portable software instance. In combination with <system-nickname>, it further qualifies the request and indicates the specific portable software instance to retrieve.

## **Standard headers**

Use the following standard HTTP header with this request:

### **Accept-Language**

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

## **Custom headers**

None.

## **Request content**

None.

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## **Required authorizations**

Certain authorizations are required to submit a request through the software management services to read a portable software instance. The user ID that initiates the request requires the same authorizations as the ones needed to perform a read operation with the z/OSMF Software Management task. The user ID needs READ access to both the Software Management task and the SAF resources that correspond to the portable software instance to retrieve. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task in IBM z/OS Management Facility Configuration Guide](#).

## **Expected response**

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more information about errors, see [“Error handling” on page 413](#).

If the request is successful, the response also includes the following JSON object:

```
{
  "name": "pswi-name",
  "system": "system-nickname",
  "description": "pswi-description",
  "directory": "UNIX-directory",
  "categories": ["categories"],
  "serverxml": ["remote-server-xml"],
  "clientxml": ["client-options-xml"],
  "jcldataset": "jcl-data-set-name",
  "lastmodified": "last-modified",
  "modifiedby": "modified-user-ID",
  "created": "date-created",
  "createdby": "created-user-ID",
  "locked": "date-locked",
  "lockedby": "locked-user-ID",
  "products": [{
    "prodname": "product-name",
    "productid": "product-ID",
```

```

        "release": "release-level",
        "vendor": "vendor-name",
        "generalavailability": "general-availability-date",
        "endofservice": "end-of-service",
        "url": "product-URL",
        "productinfofileversion": "file-version",
        "features": [
            "feature-name"
        ]
    },
    "nonsmpeproducts": [{
        "prodname": "product-name",
        "productid": "product-id",
        "release": "release-level",
        "vendor": "vendor-name",
        "generalavailability": "general-availability-date",
        "endofservice": "end-of-service-date",
        "url": "product-url",
        "productinfofileversion": "file-version",
        "features": [
            "feature-name"
        ]
    }
  ]
}

```

where:

**pswi-name**

Name of the portable software instance.

**system-nickname**

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services” on page 465](#).

**pswi-description**

Description of the portable software instance.

**UNIX-directory**

UNIX directory that contains the portable software instance files.

**categories**

Comma-separated list of the categories to which the portable software instance is assigned.

**remote-server-xml**

Information from when the Add Portable Software Instance from Download Server action was used to define the portable software instance. This information identifies the download server and the location of the portable software instance files on that server.

The GIMGTPKG program uses this XML to download the portable software instance files. For more information, see [Content of SERVER data set in z/OS SMP/E Commands](#).

**client-options-xml**

Information from when the Add Portable Software Instance from Download Server action was used to define the portable software instance. This information identifies the local z/OS server, which is needed to download the portable software instance files from the download server.

The GIMGTPKG program uses this XML to download the portable software instance files. For more information, see [Content of CLIENT data set in z/OS SMP/E Commands](#).

**jcl-data-set-name**

Information from when the Add Portable Software Instance from Download Server action was used to define the portable software instance. This value lists the name of the data set that contains the generated JCL that was used to download the portable software instance files from a remote download server.

**last-modified**

Date and time, in ISO 8601 format, that the portable software instance was last modified.

**modified-user-ID**

User ID of the user who last modified the portable software instance.

**date-created**

Date and time, in ISO 8601 format, that the portable software instance was created.

**created-user-ID**

User ID of the user who created the portable software instance.

**date-locked**

Date and time, in ISO 8601 format, that the portable software instance was locked. This attribute is null if the portable software instance is not currently locked.

**locked-user-ID**

User ID of the user who locked the portable software instance. This attribute is null if the portable software instance is not currently locked.

**products**

Array that contains each product that is included in the portable software instance.

**product-name**

Name of the product. If any FMIDs are not related to a product and feature, those FMIDs are listed under a product that is named "No Product".

**product-ID**

Identifier of the product.

**release-level**

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level.

**vendor-name**

Name of the vendor that provides the product.

**general-availability-date**

Date on which this level of the product is available to all users.

**end-of-service**

Last date on which the vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions.

**product-URL**

URL that links to additional information about the product. This information can include, for example, product lifecycle dates, product highlights, planning information, and technical descriptions.

**file-version**

Version of the most recently retrieved product information file that contains the corresponding product. The version represents the date on which the file was created or last updated.

**features**

Array that contains each feature that is included in the product.

**feature-name**

Name of the feature. If any FMIDs are not related to a product and feature, those FMIDs are listed under a feature that is named "No Feature".

**nonsmpeproducts**

List of products for the portable software instance that are not managed by SMP/E.

**product-name**

Name of the product. The value can contain up to 64 characters.

**product-ID**

Identifier for the product. The value can contain up to 64 characters.

**release-level**

Release level for the product. The value can contain up to 64 characters.

**vendor-name**

Name of the vendor that provides the product. The value can contain up to 64 characters.

**product-URL**

A URL that links to additional information about the product. The value can contain up to 256 characters.

**feature-name**

List of names of features for the product. The value can contain up to 64 characters.

**general-availability-date**

Date on which this level of the product is available to all users.

**end-of-service-date**

Last date on which the vendor delivers standard support services for this level of the product. This date is the general end of service date. It does not account for lifecycle extensions. The following values are acceptable:

**NotAnnounced**

The end of service date is not yet announced for the product.

**null**

The end of service date is unknown for the product.

**yyyy-mm-ddThh:mm:ssZ**

The known end of service date, in ISO 8601 format.

**Example**

In the following example, the GET method is used to retrieve the properties of portable software instance Hooli on system PEV174.

```
GET /zosmf/swmgmt/pswi/PEV174/Hooli HTTP/1.1
Host: pev174.yourco.com
```

*Figure 222. Sample request to retrieve the properties of a portable software instance*

A sample response is shown in [Figure 223 on page 459](#).

```
HTTP/1.1 200 OK
Date: Tues, 05 June 2020 18:53:27 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close

{
  "name": "DB2V9", "system": "PEV174", "description": null, "directory":
  "/u/fred/DB2V9", "categories": null, "lastmodified": "2014-08-20T19:23:25+00:00",
  "modifiedby": "FRED", "created": "2014-08-20T19:23:25+00:00", "createdby": "BARNEY",
  "locked": null, "lockedby": null, "products": [{"prodname": "DB2 for z/OS",
  "productid": "5635-DB2", "release": "09.01.00", "vendor": "IBM",
  "generalavailability": "20006-06-09T19:23:25+00:00", "endofservice": "2014-06-
  27T19:23:25+00:00", "url": null, "productinfofileversion": "2014-01-01",
  "features": [{"feature": "DB2 Base"}]}, {"nonsmpeproducts": [{"prodname":
  "Acme Software", "productid": "5990-ACME", "release": "100", "vendor":
  "Acme Software Products", "generalavailability": "2020-01-01T00:00:00Z",
  "endofservice": "2021-06-01T00:00:00Z", "url": "http://acmesoftware.com",
  "features": [{"Runtime"}]}]},
}
```

*Figure 223. Sample response from a request to retrieve the properties of a portable software instance*

## Add a new portable software instance

You can use this operation to add a portable software instance to z/OSMF.

### HTTP method and URI path

---

```
POST /zosmf/swmgmt/pswi
```

---

where:

#### **zosmf/swmgmt**

Identifies the software management services.

#### **pswi**

Informs the service that the request is for a portable software instance object.

### Standard headers

Use the following standard HTTP headers with this request:

#### **Content-Type**

Identifies the type of input content that is provided by the caller. The following value is acceptable:

- Content-Type: application/json (JSON)

The JSON content type is used for the JSON document that is included as input with this request.

#### **Accept-Language**

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

### Custom headers

None.

### Request content

Your request must include a JSON object that describes the portable software instance to add and the configuration information for the target software instance. An example of the request content is shown in Figure 224 on page 460.

---

```
{
  "name": "pswi-name",
  "system": "system-nickname",
  "description": "pswi-description",
  "directory": "UNIX-directory",
  "categories": ["pswi-category"],
  "zosmfuid": "zosmf-user-ID",
  "zosmfpw": "zosmf-password",
  "proxyuid": "proxy-user-ID",
  "proxypw": "proxy-password"
}
```

Figure 224. Adding a portable software instance: request content

---

where:

#### **pswi-name**

Name of the portable software instance. The name is required and must be unique on the system. The name can contain up to 30 non-blank characters, including the following alphanumeric characters,

mathematical symbols, punctuation marks, and special characters: 0-9 A-Z a-z ! " # \$ ' - / : < = > ? | @ \ ^ \_

**system-nickname**

Nickname of the system that has access to the volumes and data sets where the portable software instance is located. Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is required.

To manage the systems that are defined to z/OSMF, you can use the z/OSMF topology services. For more information, see [“Topology services”](#) on page 465.

**pswi-description**

Description of the portable software instance. The description can contain a maximum of 256 characters.

**UNIX-directory**

UNIX directory to contain the files for the portable software instance. Ensure that the directory name is valid according to the following UNIX directory name syntax:

- Must be an absolute path name.
- Must start and end with a forward slash.
- Can contain up to 1023 characters.

**pswi-category**

Comma-separated list of the categories to which the portable software instance is assigned. Each category name can contain up to 30 non-blank characters, including the following alphanumeric characters, mathematical symbols, punctuation marks, and special characters: 0-9 A-Z a-z ! " # \$ ' - / : < = > ? | @ \ ^ \_

**zosmf-user-ID**

User ID for authenticating with a remote z/OSMF instance.

**zosmf-password**

Password for authenticating with a remote z/OSMF instance.

**proxy-user-ID**

User ID for authenticating with an HTTP proxy.

**proxy-password**

Password for authenticating with an HTTP proxy.

The request content is required, but some properties are optional. For example, consider whether the UNIX directory is not in the same sysplex as the primary z/OSMF instance. You might need to authenticate with the secondary z/OSMF instance that is running in the sysplex where the source software instance is located. In addition, consider whether the primary z/OSMF instance must access an HTTP proxy server to connect with the secondary z/OSMF instance. You might also need to authenticate with that HTTP proxy server. Therefore, you might need to specify the remote z/OSMF user ID and password and the proxy user ID and password.

**Usage considerations**

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

**Required authorizations**

Certain authorizations are required to submit a request through the software management services to add a portable software instance. The user ID that initiates the request requires the same authorizations as the ones needed to perform an add operation with the z/OSMF Software Management task. The user ID needs READ access to the Software Management task and CONTROL access to the SAF resources that correspond to the portable software instance to add. If the specified categories are implicitly added during this portable software instance add operation, the user ID must also have CONTROL access to the SAF resources that correspond to the specified categories. For more information about access controls for

the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## Expected response

Validating the portable software instance files is an asynchronous operation. On completion of the initial POST request, the z/OSMF Software Management REST interface returns an HTTP response code of 202 Accepted. The response also includes the following JSON object:

```
{
  "statusurl": "url"
}
```

where:

### statusurl

Indicates the URL that provides status for the portable software instance add request. The client issues GET requests to the supplied URL to monitor the status of the operation and to obtain the result set.

Subsequent GET requests to the status monitor URL obtain the status of the operation. The following values are possible:

### 200 OK

The operation is either not yet complete or is complete with no errors. A JSON document is included that contains status information for the operation. It indicates either *running* or *complete*.

### 4nn or 5nn

An error occurred. For more information, see [“Error handling”](#) on page 413. This document contains a reason code and a list of one or more message strings that describe the errors that are detected during request processing.

### 404 Not found

The request expired. When the operation completes, the z/OSMF server retains the result set for a finite length of time. After that time passes, the result set expires and is no longer available for the client to obtain.

The response to GET requests on the status monitor URL includes the following JSON document:

```
{
  "status": "status"
}
```

where:

### status

The status of the Add portable software instance request. The status is either *running* or *complete*.

## Example

In the following example, the POST method is used to add a portable software instance to the z/OSMF instance that has a hostname of `pev174.yourco.com`.

```
POST /zosmf/swmgmt/pswi HTTP/1.1
Host: pev174.yourco.com
Content-Type: application/json
{
  "name": "Hooli",
  "system": "PEV174",
  "description": "Fictitious software product from Hooli",
  "directory": "/u/ibmusr/hoolipswi/"
}
```

*Figure 225. Sample request to add a portable software instance*

An example response is as follows:

```
HTTP/1.1 202 Accepted
Date: Fri, 5 June 2020 18:53:04 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"statusurl":"https://\pev174.yourco.com\zosmf\swmgmt\statusmonitor\addpswi\4837290198343"}
```

*Figure 226. Response to sample request to add a portable software instance*

Figure 226 on page 463 indicates that the request to validate and create a portable software instance entry was accepted, and the status monitor URL is provided.

A subsequent GET request to the status monitor URL is as follows:

```
GET /zosmf/swmgmt/statusmonitor/addpswi/4837290198343 HTTP/1.1
Host: pev174.yourco.com
```

*Figure 227. Sample GET request to the status monitor URL*

An example response is as follows:

```
HTTP/1.1 200 OK
Date: Fri, 5 June 2020 18:54:04 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"running"}
```

*Figure 228. Response to sample GET request to the status monitor URL*

Figure 228 on page 463 indicates that the request to validate and create a portable software instance entry is still running.

A final request to the status monitor URL is as follows:

```
GET /zosmf/swmgmt/statusmonitor/addpswi/4837290198343 HTTP/1.1
Host: pev174.yourco.com
```

*Figure 229. Sample final request to the status monitor URL*

An example response is as follows:

```
HTTP/1.1 200 OK
Date: Fri, 5 June 2020 18:55:04 +00004GMT
Content-Type: application/json
Content-Language: en
Connection: close
{"status":"complete"}
```

*Figure 230. Response to sample final request to the status monitor URL*

Figure 230 on page 463 indicates that the request to validate and create a portable software instance completed with no errors.

## Delete a portable software instance

You can use this operation to remove a portable software instance definition from z/OSMF. The delete operation removes only the definition of the portable software instance from z/OSMF. The UNIX directory that contains the portable software instance files is not affected.

### HTTP method and URI path

---

```
DELETE /zosmf/swmgmt/pswi/<system-nickname>/<pswi-name>
```

---

where:

#### **zosmf/swmgmt**

Identifies the software management services.

#### **pswi**

Informs the service that the request is for a portable software instance object.

#### **<system-nickname>**

Nickname of the z/OSMF host system that has access to the volumes and data sets where the portable software instance is located. In combination with *<pswi-name>*, it further qualifies the request and indicates the specific portable software instance to delete.

To obtain information about the specified system, you can use the z/OSMF topology services. For more information, see [“Topology services”](#) on page 465.

#### **<pswi-name>**

Name of the portable software instance. In combination with *<system-nickname>*, it further qualifies the request and indicates the specific portable software instance to delete.

### Standard headers

Use the following standard HTTP header with this request:

#### **Accept-Language**

Identifies the preferred language for any messages that are returned to the caller. The following values are acceptable:

- Accept-Language: en (English)
- Accept-Language: ja (Japanese)

If any other language value is specified or if the header is omitted, then English is used.

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

### Required authorizations

Certain authorizations are required to submit a request through the software management services to delete a portable software instance. The user ID that initiates the request requires the same authorizations as the ones needed to perform a remove operation with the z/OSMF Software Management task. The user ID needs READ access to the Software Management task and CONTROL access to the SAF

resources that correspond to the portable software instance to delete. For more information about access controls for the Software Management task, see [Creating access controls for the Software Management task](#) in *IBM z/OS Management Facility Configuration Guide*.

## Expected response

On completion, the service returns an HTTP response with a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For more information about errors, see [“Error handling” on page 413](#).

## Example

In the following example, the DELETE method is used to delete portable software instance Hooli from system PEV174.

```
DELETE /zosmf/swmgmt/pswi/PEV174/Hooli HTTP/1.1
Host: pev174.yourco.com
```

*Figure 231. Sample request to delete a portable software instance*

Figure 232 on page 465 provides a sample response that indicates that the delete operation was successful.

```
HTTP/1.1 200 OK
Date: Tues, 05 June 2020 18:53:27 +00004GMT
```

*Figure 232. Sample response for a delete portable software instance request*

## Topology services

The topology services is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with the groups, sysplexes, central processor complexes (CPCs), and systems that are defined to z/OSMF, as described in this topic.

Table 297 on page 465 lists the operations that the topology services provide.

Table 297. Operations provided through the topology services.	
Operation	HTTP method and URI path
<a href="#">“List the systems defined to z/OSMF” on page 467</a>	GET /zosmf/resttopology/systems
<a href="#">“List the groups defined to z/OSMF” on page 470</a>	GET /zosmf/resttopology/groups
<a href="#">“List the systems included in a group” on page 471</a>	GET /zosmf/resttopology/systems/groupName/<groupName>

Table 297. Operations provided through the topology services. (continued)

Operation	HTTP method and URI path
<a href="#">“List the sysplexes defined to z/OSMF” on page 474</a>	GET /zosmf/resttopology/sysplexes
<a href="#">“List the systems included in a sysplex” on page 476</a>	GET /zosmf/resttopology/systems/sysplexName/<sysplexName>
<a href="#">“List the systems included in a CPC” on page 478</a>	GET /zosmf/resttopology/systems/cpcName/<cpcName>

## Required authorizations

To submit requests through the topology services, your user ID requires authorization to the Systems task provided in z/OSMF. Ensure that your user ID has READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.SETTINGS.SYSTEMS.VIEW. By default, users with user IDs connected to the IZUADMIN and IZUUSER security groups can access the topology services.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data. The following JSON object is used by all topology services for returning data about the requested operation:

```
{
  "items": "item-list",
  "numRows": "total-items"
}
```

where:

### item-list

Array that contains the items that were retrieved. The attributes provided in the array depend on the requested operation.

### total-items

Number of items retrieved.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

### HTTP 200 OK

Success.

### HTTP 400 Bad request

The request contained incorrect parameters.

### HTTP 401 Unauthorized

The submitter of the request did not authenticate to z/OSMF or is not authorized to use the topology services.

**HTTP 403 Forbidden**

The server rejected the request.

**HTTP 404 Bad URL**

The target of the request (a URL) was not found.

**HTTP 405 Method not allowed**

The service does not support the HTTP method specified for the request.

**HTTP 500 Internal server error**

A programming error occurred.

**Error logging**

Errors from the topology services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## List the systems defined to z/OSMF

You can use this operation to obtain a list of the systems that are defined to a z/OSMF instance.

**HTTP method and URI path**

---

```
GET /zosmf/resttopology/systems
```

---

where:

- **zosmf/resttopology** identifies the topology services.
- **systems** informs the service that the request is to retrieve a list of the systems that are defined to the z/OSMF instance.

**Standard headers**

Use the following standard HTTP header with this request:

Content-Type: application/json

**Custom headers**

None.

**Request content**

None.

**Usage considerations**

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

**Required authorizations**

See [“Required authorizations”](#) on page 466.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 466](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

### **system-nickname**

Unique name assigned to the system definition.

### **system-name**

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

### **sysplex-name**

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

### **group-names**

Comma-separated list of the groups to which the system is assigned.

### **url**

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

### **zos-level**

Version and release of the z/OS image installed on the system. The version and release has the format z/OS VxxRyy where V stands for version, xx is the version number, R stands for release, and yy is the release number. For example, z/OS V2R1.

### **JES-member-name**

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

### **JES-type**

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

### **CPC-name**

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

### **CPC-serial**

Serial number of the CPC.

### **proxy-name**

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

**server-name**

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

**total-items**

Number of system definitions that were retrieved.

**Example**

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/resttopology/systems HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 233. Sample request to retrieve a list of systems*

A sample response is shown in [Figure 234 on page 469](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    },
    {
      "systemNickName": "sys2",
      "systemName": "sys2",
      "sysplexName": "plex2",
      "groupNames": "production",
      "url": "https://zosmf2.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY2",
      "jesType": "JES3",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs-sftp"
    },
    {
      "systemNickName": "sys3",
      "systemName": "sys3",
      "sysplexName": "plex3",
      "groupNames": "test",
      "url": "https://zosmf3.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY3",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "3"
}
```

*Figure 234. Sample response from a request to retrieve a list of systems*

## List the groups defined to z/OSMF

You can use this operation to obtain a list of the groups that are defined to a z/OSMF instance.

### HTTP method and URI path

---

```
GET /zosmf/resttopology/groups
```

---

where:

- **zosmf/resttopology** identifies the topology services.
- **groups** informs the service that the request is to retrieve a list of the groups that are defined to the z/OSMF instance.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### Required authorizations

See [“Required authorizations” on page 466](#).

### Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 466](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "name": "group-name",
      "description": "group-description",
      "systemNickNames": "system-nicknames"
    }
  ],
  "numRows": "total-items"
}
```

where:

#### **group-name**

Name of the groups defined to z/OSMF. A value of <None> indicates that one or more systems are not assigned to a group.

**group-description**

Description of the group.

**system-nicknames**

Comma-separated list of the systems assigned to the group. Each system is identified by its nickname.

**total-items**

Number of groups that were retrieved.

**Example**

In the following example, the GET method is used to retrieve a list of the groups that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/resttopology/groups HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 235. Sample request to retrieve a list of groups*

A sample response is shown in [Figure 236 on page 471](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "name": "development",
      "description": "This group contains the systems used by development.",
      "systemNickNames": "sys1"
    },
    {
      "name": "production",
      "description": "This group contains the systems that are in production.",
      "systemNickNames": "sys2"
    },
    {
      "name": "test",
      "description": "This group contains the systems that are used for testing code.",
      "systemNickNames": "sys1,sys3"
    }
  ],
  "numRows": "3"
}
```

*Figure 236. Sample response from a request to retrieve a list of groups*

## List the systems included in a group

You can use this operation to obtain a list of the systems that are included in a group.

### HTTP method and URI path

```
GET /zosmf/resttopology/systems/groupName/<groupName>
```

where:

- **zosmf/resttopology** identifies the topology services.
- **systems/groupName** informs the service that the request is to retrieve a list of the systems that are defined to a specific group.
- **<groupName>** identifies the group for which to obtain the list of systems. If the group name contains a number sign (#), encode the number sign as %23. For example, if the group name is *test#systems*, specify *test%23systems*. Otherwise, the service will truncate *#systems*, and use *test* as the group name.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 466](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error has occurred. For more details, see [“Error handling” on page 466](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

### **system-nickname**

Unique name assigned to the system definition.

### **system-name**

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

### **sysplex-name**

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

### **group-names**

Comma-separated list of the groups to which the system is assigned.

**url**

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

**zos-level**

Version and release of the z/OS image installed on the system. The version and release has the format *z/OS VxxRyy* where *V* stands for version, *xx* is the version number, *R* stands for release, and *yy* is the release number. For example, *z/OS V2R1*.

**JES-member-name**

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

**JES-type**

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

**CPC-name**

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

**CPC-serial**

Serial number of the CPC.

**proxy-name**

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

**server-name**

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

**total-items**

Number of system definitions that were retrieved.

**Example**

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance with host name *zosmf1.yourco.com* and that are assigned to the group *test*.

```
GET /zosmf/resttopology/systems/groupName/test HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 237. Sample request to retrieve a list of systems included in a group*

A sample response is shown in [Figure 238 on page 474](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    },
    {
      "systemNickName": "sys3",
      "systemName": "sys3",
      "sysplexName": "plex3",
      "groupNames": "test",
      "url": "https://zosmf3.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY3",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "2"
}

```

Figure 238. Sample response from a request to retrieve a list of systems included in a group

## List the sysplexes defined to z/OSMF

You can use this operation to obtain a list of the sysplexes that are defined to a z/OSMF instance.

### HTTP method and URI path

```
GET /zosmf/resttopology/sysplexes
```

where:

- **zosmf/resttopology** identifies the topology services.
- **sysplexes** informs the service that the request is to retrieve a list of the sysplexes that are defined to the z/OSMF instance.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Required authorizations”](#) on page 466.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 466.

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "sysplexName": "sysplex-name",
      "systemNickNames": "system-nicknames"
    }
  ],
  "numRows": "total-items"
}
```

where:

### **sysplex-name**

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility. A value of *<Not Specified>* indicates that one or more systems are not assigned to a sysplex.

### **system-nicknames**

Comma-separated list of the systems assigned to the sysplex. Each system is identified by its nickname.

### **total-items**

Number of sysplexes that were retrieved.

## Example

In the following example, the GET method is used to retrieve a list of the sysplexes that are defined to the z/OSMF instance that has a host name of *zosmf1.yourco.com*.

```
GET /zosmf/resttopology/sysplexes HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 239. Sample request to retrieve a list of sysplexes*

A sample response is shown in [Figure 240 on page 476](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close
```

```
{
  "items": [
    {
      "sysplexName": "plex1",
      "systemNickNames": "sys1"
    },
    {
      "sysplexName": "plex2",
      "systemNickNames": "sys2"
    },
    {
      "sysplexName": "plex3",
      "systemNickNames": "sys3"
    }
  ],
  "numRows": "3"
}
```

*Figure 240. Sample response from a request to retrieve a list of sysplexes*

## List the systems included in a sysplex

You can use this operation to obtain a list of the systems that are included in a sysplex.

### HTTP method and URI path

---

```
GET /zosmf/resttopology/systems/sysplexName/<sysplexName>
```

---

where:

- **zosmf/resttopology** identifies the topology services.
- **systems/sysplexName** informs the service that the request is to retrieve a list of the systems that are included in a specific sysplex.
- **<sysplexName>** identifies the sysplex for which to obtain the list of systems.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

### Required authorizations

See [“Required authorizations”](#) on page 466.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 466](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

### **system-nickname**

Unique name assigned to the system definition.

### **system-name**

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

### **sysplex-name**

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

### **group-names**

Comma-separated list of the groups to which the system is assigned.

### **url**

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

### **zos-level**

Version and release of the z/OS image installed on the system. The version and release has the format z/OS VxxRyy where *V* stands for version, *xx* is the version number, *R* stands for release, and *yy* is the release number. For example, z/OS V2R1.

### **JES-member-name**

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

### **JES-type**

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

### **CPC-name**

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

### **CPC-serial**

Serial number of the CPC.

### **proxy-name**

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

**server-name**

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

**total-items**

Number of system definitions that were retrieved.

**Example**

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance with host name *zosmf1.yourco.com* and that are included in sysplex *plex1*.

```
GET /zosmf/resttopology/systems/sysplexName/plex1 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 241. Sample request to retrieve a list of systems included in a sysplex*

A sample response is shown in [Figure 242 on page 478](#).

```
HTTP/1.1 200 OK
Date: Thu, 15 Jan 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "",
      "cpcSerial": "",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "1"
}
```

*Figure 242. Sample response from a request to retrieve a list of systems included in a sysplex*

## List the systems included in a CPC

You can use this operation to obtain a list of the systems that are included in a central processor complex (CPC).

### HTTP method and URI path

```
GET /zosmf/resttopology/systems/cpcName/<cpcName>
```

where:

- **zosmf/resttopology** identifies the topology services.
- **systems/cpcName** informs the service that the request is to retrieve a list of the systems that are included in a specific CPC.
- **<cpcName>** identifies the CPC for which to obtain the list of systems.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 466](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error has occurred. For more details, see [“Error handling” on page 466](#).

If the request was successful, the response also includes the following JSON object:

```
{
  "items": [
    {
      "systemNickName": "system-nickname",
      "systemName": "system-name",
      "sysplexName": "sysplex-name",
      "groupNames": "group-names",
      "url": "url",
      "zosVR": "zos-level",
      "jesMemberName": "JES-member-name",
      "jesType": "JES-type",
      "cpcName": "CPC-name",
      "cpcSerial": "CPC-serial",
      "httpProxyName": "proxy-name",
      "ftpDestinationName": "server-name"
    }
  ],
  "numRows": "total-items"
}
```

where:

### **system-nickname**

Unique name assigned to the system definition.

### **system-name**

Name specified for the system on the SYSNAME parameter in the IEASYSxx parmlib member.

### **sysplex-name**

Name of the sysplex where the z/OS system is a member. The name is the value specified for the SYSPLEX parameter of the cross-system coupling facility (XCF) couple data set format utility.

### **group-names**

Comma-separated list of the groups to which the system is assigned.

**url**

URL used to access the z/OSMF instance that resides in the same sysplex as the system identified by the **systemName** attribute. Or, the URL used to access the application server that is hosting the server-side code for the plug-ins your enterprise imported into z/OSMF.

**zos-level**

Version and release of the z/OS image installed on the system. The version and release has the format *z/OS VxxRyy* where *V* stands for version, *xx* is the version number, *R* stands for release, and *yy* is the release number. For example, *z/OS V2R1*.

**JES-member-name**

JES2 multi-access spool (MAS) member name or JES3 complex member name that is assigned to the primary job entry subsystem (JES) that is running on the system.

**JES-type**

Type for the primary job entry subsystem running on the system. The type is either JES2 or JES3.

**CPC-name**

Name specified for the central processor complex (CPC) at the support element (SE) of that processor complex.

**CPC-serial**

Serial number of the CPC.

**proxy-name**

Name of the HTTP proxy definition that specifies the settings required to access the system through an HTTP or SOCKS proxy server.

**server-name**

Name of the server definition that specifies the settings required to access the FTP or SFTP server that is running on the system.

**total-items**

Number of system definitions that were retrieved.

**Example**

In the following example, the GET method is used to retrieve a list of the systems that are defined to the z/OSMF instance with host name *zosmf1.yourco.com* and that are included in CPC *CPC1*.

```
GET /zosmf/resttopology/systems/cpcName/CPC1 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 243. Sample request to retrieve a list of systems included in a CPC*

A sample response is shown in [Figure 244 on page 481](#).

```

HTTP/1.1 200 OK
Date: Thu, 15 Feb 2015 05:39:28 +0000GMT
Connection: close

{
  "items": [
    {
      "systemNickName": "sys1",
      "systemName": "sys1",
      "sysplexName": "plex1",
      "groupNames": "test,development",
      "url": "https://zosmf1.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY1",
      "jesType": "JES2",
      "cpcName": "CPC1",
      "cpcSerial": "30104",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    },
    {
      "systemNickName": "sys10",
      "systemName": "sys10",
      "sysplexName": "plex2",
      "groupNames": "production",
      "url": "https://zosmf10.yourco.com/zosmf/",
      "zosVR": "z/OS V2R1",
      "jesMemberName": "SY10",
      "jesType": "JES2",
      "cpcName": "CPC1",
      "cpcSerial": "30104",
      "httpProxyName": "No Proxy",
      "ftpDestinationName": "IBM-testcase-mvs"
    }
  ],
  "numRows": "2"
}

```

Figure 244. Sample response from a request to retrieve a list of systems included in a CPC

## TSO/E address space services

TSO/E address space services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with TSO/E address spaces on a z/OS system.

Table 298 on page 481 lists the operations that the TSO/E address space services provide.

Table 298. Operations provided through the TSO/E address space services.	
Operation	HTTP method and URI path
<b>“Start or reconnect to a TSO/E address space” on page 485</b>	POST /zosmf/tsoApp/tso?parms
<b>“Start an application in a TSO/E address space” on page 490</b>	POST /zosmf/tsoApp/app/servletKey/appKey
<b>“Receive messages from a TSO/E address space” on page 509</b>	GET /zosmf/tsoApp/tso/servletKey

Table 298. Operations provided through the TSO/E address space services. (continued)

Operation	HTTP method and URI path
<a href="#">“Receive messages from an application” on page 510</a>	GET /zosmf/tsoApp/app/servletKey/appKey
<a href="#">“Send messages to a TSO/E address space” on page 504</a>	PUT /zosmf/tsoApp/tso/servletKey?[readReply=true false]
<a href="#">“Send messages to an application” on page 506</a>	PUT /zosmf/tsoApp/app/servletKey/appKey
<a href="#">“Ping a TSO/E address space” on page 507</a>	PUT /zosmf/tsoApp/tso/ping/servletKey
<a href="#">“End a TSO/E address space” on page 511</a>	DELETE /zosmf/tsoApp/tso/servletKey?[tsoforcecancel=true false]

## How to use the Swagger interface

You can use the Swagger interface to display information about the TSO/E address space services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

## Required authorizations

Generally, your z/OSMF user ID requires the same authorizations for using the TSO/E address space services as when you perform these operations through a TSO/E session on the z/OS system. For example, to start an application in a TSO/E address space, your user ID must be authorized to operate that application.

In addition, to use TSO/E address space services, you must have:

- READ access to the *account* resource in class ACCTNUM, where *account* is the value that is specified in the COMMON\_TSO ACCT option in parmlib.
- READ access to the CEA.CEATSO.TSOREQUEST resource in class SERVAUTH.
- READ access to the *proc* resource in class TSOPROC, where *proc* is the value that is specified with the COMMON\_TSO PROC option in parmlib.
- READ access to the <SAF\_PREFIX>.\*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF\_PREFIX>.IzuManagementFacilityTsoServices.izuUsers resource name in the EJBROLE class.

You must also ensure that the z/OSMF started task user ID, which is IZUSVR by default, has READ access to the CEA.CEATSO.TSOREQUEST resource in class SERVAUTH.

To create a TSO/E address space on a remote system, you require the following authorizations:

- You must be authorized to the SAF resource profile that controls the ability to send data to the remote system (*systemname*), as indicated:

```
CEA.CEATSO.FLOW.systemname
```

- To flow data between different systems in the sysplex, you must be authorized to do so by your external security manager, such as a RACF database with sysplex-wide scope. For example, to flow data between System A and System B, you must have access to the following resource profiles:

```
CEA.CEATSO.FLOW.SYSTEMA
CEA.CEATSO.FLOW.SYSTEMB
```

The TSO/E address space services authority might already be defined if you are using z/OS data set and file REST services, as those services require similar authority.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. The following JSON object is used by all TSO/E address space services for returning data and status about the requested operations. The attributes that are provided in the JSON object depend on the requested operation.

```
{
  "servletKey": "servlet-key",
  "ver": "structure-version",
  "queueID": "message queue ID",
  "remoteSys": "remote system",
  "ceatsoconn": "CEA connection handler",
  "tsoData": "TSO/E-messages",
  "appData": "application-messages",
  "timeout": "timeout-indicator",
  "reused": "reconnected-indicator",
  "msgData": "z/OSMF-messages",
  "messages": "unexpected z/OSMF-messages"
}
```

Where:

### servletKey

Unique identifier for the servlet entry. It maps to the TSO/E address space ID and provides additional information about the address space. To communicate with the TSO/E address space, the client must provide the servlet key.

### ver

Version of the TSO/E address space services and the JSON object structure that is used for this request. The version sequence starts at 0100, and is incremented only if the services or the JSON structure changes. In your application, check the value of the returned structure and verify that your application is compatible with the current API.

### queueID

When the TSO/E address space interface starts a new TSO/E session, it also creates a new z/OS UNIX message queue to enable communication between the client and the TSO/E address space. This value is the identifier for the z/OS UNIX message queue.

### remoteSys

System name of the remote system on which the TSO/E address space is to be started.

### ceatsoconn

100-byte binary key (in hexadecimal), which is used by callers to perform subsequent operations with a remote TSO/E address space. To use this value with CEA APIs, the caller must first convert it to raw binary. This hexadecimal string includes 200 characters, where 2 characters represent 1 byte.

### tsoData

TSO/E messages that were received during the request. The *tsoData* attribute is included in the JSON object only if TSO/E messages were received.

The value that is returned in the *tsoData* attribute is a JSON object that describes the messages that were received. For example, the TSO/E message JSON format has the following syntax:

```
{"message-type":{"VERSION":"JSON-version","data-type":"data-value"}}
```

where:

**message-type**

Keyword that identifies the type of TSO/E message. The value can be TSO MESSAGE, TSO PROMPT, or TSO RESPONSE.

**JSON-version**

A four-digit number that identifies the JSON version that is used to format the message.

**data-type**

Keyword that describes the type of data that is included in the data-value variable. The value can be DATA, HIDDEN, or ACTION.

Example: `{"TSO RESPONSE": {"VERSION": "0100", "DATA": "ALLOC DA"}}`

**appData**

Messages that are received from an application that is running in a TSO/E address space during the request. The appData attribute is included in the JSON object only if messages were received from an application and no TSO/E messages were received during the request.

**timeout**

Indicator of whether the request timed out while it waited for a response. The value is "true" if the request timed out. Otherwise, the value is "false".

If the service creates a new TSO/E address space, the service attempts to read the initial startup TSO/E messages. If no messages are received in the allotted time, this value is set to "true".

If the service reconnects the user to an existing TSO/E address space, no startup messages are expected; therefore, the service does not wait for any startup TSO/E messages.

**reused**

Indicator of whether the service connected the user to an existing TSO/E session instead of a new session. The reused attribute is included in the JSON object only if the appsessionid parameter is provided for the start TSO/E address space request. The value that is returned for the reused attribute is "true" if a TSO/E address space with that appsessionid exists. Otherwise, the value is "false".

**msgData**

z/OSMF messages received during the request. The messages attribute is included in the JSON object only if an error occurred during the request. The message ID and message text are provided for each z/OSMF message received.

**messages**

z/OSMF messages received for unexpected errors.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

**HTTP 200 OK**

Success.

**HTTP 400 Bad request**

Request contained incorrect parameters.

**HTTP 401 Unauthorized**

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the TSO/E address space services.

**HTTP 404 Bad URL**

Target of the request (a URL) was not found.

**HTTP 500 Internal server error**

Programming error.

## Error logging

Errors from the TSO/E address space services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Usage Notes

- Establish the SYSCALL environment in your REXX script.

If you want to run a REXX script and get the response from the TSO/E REST APIs, establish the SYSCALL environment in your REXX script with a SYSCALL request that begins with ADDRESS SYSCALL. For example,

```
if syscalls('ON')>3 then
do
  say 'Unable to establish the SYSCALL environment'
  return
end
```

For more information on how to establish the SYSCALL environment, see [Establishing the SYSCALL environment in z/OS Using REXX and z/OS UNIX System Services](#)

- Things to consider when you use the TSO/E command ALLOCATE:

If you plan to use ALLOCATE to create a data set or file with the APIs provided, you need to use the TSO/E command FREE to release it after you finish working on it. Otherwise, the TSO/E address space locks the data set or file it creates. This is important if you are using the API PUT /zosmf/tsoApp/{version}/tsowith cmdState=stateless. Ensure that the ALLOCATE and FREE commands are invoked in pairs.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Start or reconnect to a TSO/E address space

You can use this operation to start a new TSO/E address space or to reconnect to a dormant TSO/E address space.

### HTTP method and URI path

---

```
POST /zosmf/tsoApp/tso?<parms>
```

---

Where:

- zosmf/tsoApp** identifies the TSO/E address space services.
- tso** informs the service that the request is for a TSO/E address space.
- <parms>** qualifies the request with one or more of the parameters that are described in [Table 299 on page 486](#).

When the TSO/E address space interface starts a new TSO/E session, it also creates a new z/OS UNIX message queue to enable communication between the client and the TSO/E address space.

When the interface reconnects to a dormant TSO/E address space, the interface reuses the session resources, including the z/OS UNIX message queue.

**Note:** A dormant TSO/E address space is an address space that is deactivated for communication through its z/OS UNIX message queue, but remains available at a TSO/E READY prompt for time.

## Supported parameters

Table 299. Supported parameters for the start and reconnect TSO/E session requests		
Parameter	Required	Description
<b>proc</b>	Yes	Name of the TSO/E logon procedure to use to log in to the TSO/E address space.
<b>chset</b>	Yes	Character set to use for the caller's TSO/E address space. This value is used by the applications running in the TSO/E address space to convert messages and responses from UTF-8 to EBCDIC. The default character set, which is 697 decimal, are used if zero is specified as the value.
<b>cpage</b>	Yes	Code page to use for the caller's TSO/E address space. This value is used by the applications running in the TSO/E address space to convert messages and responses from UTF-8 to EBCDIC. The default code page, which is 1047 decimal, is used if zero is specified as the value.
<b>rows</b>	Yes	Number of rows to be displayed on the screen. The default number of rows, which is 24, is used if zero is specified as the value.
<b>cols</b>	Yes	Number of columns to be displayed on the screen. The default number of columns, which is 80, is used if zero is specified as the value.
<b>acct</b>	No	TSO/E user account number.
<b>ugrp</b>	No	Name of the TSO/E user group.
<b>rsize</b>	No	Region size to use for the TSO/E address space.
<b>appsessid</b>	No (for new), Yes (for reconnect)	Identifier that uniquely identifies the TSO/E address space. This parameter is optional when starting a new TSO/E address space, and it is required when reconnecting to an existing TSO/E address space. If an address space with the specified identifier does not exist, a new TSO/E address space is created and assigned the identifier that is specified.
<b>system</b>	No	System on which the TSO/E address space is to be created or reconnected. Specify the system name. This parameter is optional; if not specified, the request is processed on the local system.
<b>apptag</b>	No	Identifies the application that is responsible for creating the TSO/E address space. This value is used by CEA. The value of apptag is 2 - 8 characters (A-Z, a-z, 0-9) and cannot begin with a digit. The value is case-sensitive.  This parameter is optional; if not specified, the default value is IZUTSOAP.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 482](#).

In addition, only the z/OSMF user that started the TSO/E address space is authorized to use the z/OS UNIX message queue that is associated with that address space.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 484](#).

The response also includes a JSON object with additional information about the results of the request. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example: Creating a new TSO/E address space on the local system

To create a new address space with the following settings on the local system, submit the request that is depicted in [Figure 245 on page 487](#):

- Procedure name: IKJACCNT
- Character set: 697
- Code page: 1047
- Screen rows: 204
- Screen columns: 160
- Region size: 50000
- Account number: DEFAULT

```
POST /zosmf/tsoApp/tso?proc=IKJACCNT&chset=697&cpage=1047&rows=204
&cols=160&rsz=50000&acct=DEFAULT HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 245. Sample request to create a new TSO/E address space on the local system*

A sample response is shown in [Figure 246 on page 488](#).

Figure 246. Sample response from create TSO/E address space request on the local system

- Screen columns: 160
- Region size: 50000
- Account number: DEFAULT
- Application Session ID: sdsf\_23715376543765

```
POST /zosmf/tsoApp/tso?proc=IKJACCNT&chset=697&cpage=1047&rows=204
&cols=160&rsz=50000&acct=DEFAULT&appid=sdsf_23715376543765 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 249. Sample request to reconnect to an existing TSO/E address space on the local system*

A sample response is shown in [Figure 250 on page 489](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-55-aaakaaac","queueID":"4","sessionID":"0x37","ver":"0100","reused":false,"timeout":false}
```

*Figure 250. Sample response from a reconnect to TSO/E address space request on the local system*

### **Example: Reconnecting to an existing TSO/E address space on a remote system**

To reconnect to the TSO/E address space associated with application session ID *sdsf\_23715376543765* on a remote system (SYS2), specify the following settings, and submit the request that is depicted in [Figure 249 on page 489](#):

- Procedure name: IKJACCNT
- Character set: 697
- Code page: 1047
- Screen rows: 204
- Screen columns: 160
- Region size: 50000
- Account number: DEFAULT
- System: SYS2
- Application Session ID: sdsf\_23715376543765

```
POST /zosmf/tsoApp/tso?proc=IKJACCNT&chset=697&cpage=1047&rows=204
&cols=160&rsz=50000&acct=DEFAULT&system=SYS2&appid=sdsf_23715376543765 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 251. Sample request to reconnect to an existing TSO/E address space on a remote system*

A sample response is shown in [Figure 250 on page 489](#).

Figure 252. Sample response from a request to reconnect to a TSO/E address space on a remote system

You can use this operation to start an application in a TSO/E address space.

POST /zosmf/tsoApp/app/<servletKey>/<appKey>

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **app** informs the service that the request is for an application running in a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space in which to start the application.
- **<appKey>** identifies the application to be started.

Use the following standard HTTP header with this request:

Content-Type: application/json

None.

To start an application in a TSO/E address space, your request must include a JSON object that contains the application start command, plus any input values that the application requires on start-up. The following example shows a possible JSON object structure. Note, however, that command requirements can vary. For the specific requirements for the start command, refer to the documentation for the command.

Figure 253. Starting an application: example of the request content

In the example, the first three parameters are required to establish communication between the REST-caller and the command. The rest is optional. This order is not enforced.

The parameters are further described, as follows:

TSO/E command that is used to start the application. For example, the ISFWEB parameter is used to start the SDSF application.

## &1 and &2

Variables for passing the command processor output message type ID and the command processor input message type ID.

## {queueid}

Identifier for the z/OS UNIX message queue, which is used to enable communication between the client and the TSO/E address space. If you use the TSO/E address space services to start a new TSO/E session, the message queue ID is returned in the HTTP response body, along with other values. See [“Content type used for HTTP request and response data”](#) on page 483.

## {user-parm-list}

Optional list of application-specific parameters.

## Processing overview

When the client requests to start an application in a TSO/E address space, the API completes the following actions:

- Assigns the command processor input and output message types to use for communication with the application to be started. The message types will be used only for the application identified by the **appKey**.
- Replaces the variables in the command with the assigned command processor message types.
- Sends the TSO/E command to the TSO/E address space identified by the servlet key.
- Attempts to read a TSO/E or application response message. If no messages are received in the time allotted, a timeout indication is returned. Any TSO/E messages are prioritized over application messages. Typically, when a caller receives a TSO/E message while attempting to receive application messages, the caller processes the TSO/E messages, then attempts to retrieve the queued application messages.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Required authorizations”](#) on page 482.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 484.

The response also includes a JSON object that contains the application response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data”](#) on page 483.

## Example

The application *BkApp001* is a TSO/E command processor. To start this application in the TSO/E address space associated with servlet key *ZOSMFAD-71-aabcaaaaf*, submit the request shown in [Figure 254](#) on page 491.

```
POST /zosmf/tsoApp/app/ZOSMFAD-71-aabcaaaaf/BkApp001 HTTP/1.1
Host: zosmf1.yourco.com

{"startcmd": "ISFWEB &1 &2 12345"}
```

*Figure 254. Sample request to start an application in a TSO/E address space*

In this example, the request content includes three variables:

- Command processor output message type ID is passed in placeholder variable `&1`
- Command processor input message type ID is passed in placeholder variable `&2`
- Message queue ID is included as the third parameter `12345`.

Thus, if the start command is **ISFWEB** and the message queue ID was 12345, your request would specify the following values to start the application:

```
ISFWEB &1 &2 12345
```

Before the variables are passed to the command, `&1` is automatically resolved by z/OSMF to the message type that the application uses to send messages to the client (the output message) and `&2` is automatically resolved to the message type that is used by the application to receive messages from the client (the input message).

A sample response is shown in [Figure 255 on page 492](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaaaf","ver":"0100",
"appData":...,"timeout":false}
```

*Figure 255. Sample response from a start an application in a TSO/E address space request*

## Issue a TSO/E command with z/OSMF REST API

You can use this operation to issue a TSO/E command and get a correspond response.

### HTTP method and URI path

```
PUT /zosmf/tsoApp/{version}/tso
```

Where:

- `/zosmf/tsoAPP` identifies the TSO/E address space services.
- `version` identifies the version of the TSO/E REST API service. The following value is valid: `v1`.
- `/tso` informs the service that the request is for a TSO/E address space.

### Query parameters

None.

### Standard headers

Use the following standard HTTP header with this request: `Content-Type: application/json`.

## Request body

*Table 300. Supported parameters for the TSO/E command with z/OSMF REST API.*

The request content is expected to contain a JSON object. See [Table 300 on page 493](#) for a description of the fields.

Parameter	Required or Optional	Description
<b>tsoCmd</b>	Required	Specifies the command to issue.
<b>system</b>	Optional	Name of the system in the same sysplex that the command is routed to. The default is the local system.
<b>maxWaitTime</b>	Optional	<p>Specify the max amount of time in seconds that the Issue command API will continue to get responses. The API tries to get as many responses as possible until the amount of time elapsed or until the TSO PROMPT is received. The TSO PROMPT indicates that the TSO address space is ready for more input from the client. The input can either be a new command or the next part of the previous command. After the TSO PROMPT is received, there is no further response for a TSO command.</p> <ul style="list-style-type: none"><li>• If you do not specify the value for maxWaitTime, the API gets as many messages as possible until any of the three things happen:<ul style="list-style-type: none"><li>– The TSO PROMPT is received.</li><li>– The API returns if there is no new response during last 5 seconds.</li><li>– 30 seconds elapses.</li></ul></li><li>• The minimal value is 1 second.</li><li>• The maximum value is 300, which means 5 mins.</li></ul>

Table 300. Supported parameters for the TSO/E command with z/OSMF REST API.

The request content is expected to contain a JSON object. See [Table 300 on page 493](#) for a description of the fields.

(continued)

Parameter	Required or Optional	Description
<b>cmdState</b>	Optional	<p>Specify the state of the TSO/E command you want to issue:</p> <p><b>stateless</b>  The tsoCmd is a stateless TSO/E command. You can use a single 'Issue TSO/E command REST API' to fulfill the request.</p> <p><b>Note:</b> for a stateless command, all the command responses are returned in the API response. You cannot perform any further action against the TSO/E address space, which is used to issue the command. This is the default. For a single user, you can issue up to 45 stateless command APIs concurrently. Case is not significant.</p> <p><b>stateful</b>  The tsoCmd is a stateful TSO/E command, which means you need to issue a series of TSO/E commands that are related to each other. You need to issue them one after another by the Issue command REST API. Case is not significant.</p> <p>The address space serves these stateful commands, which are identified by the servletKey and kept alive for 10 minutes. If there is no further Issue stateful command or get response request from the user during that time, the address space is released.</p> <p>For a single user, you can issue up to 45 stateful command APIs concurrently. The API is differentiated by the servletKey. It is active for 10 minutes, and is counted as 1 of the 45 during that 10 minutes. This is true even if you do not issue any stateful commands/get response API in the TSO/E address space that is identified by the servletKey.</p>
<b>servletKey</b>	Optional	<p>Unique identifier for a stateful TSO/E command entry. It maps to the TSO/E address space in which the stateful TSO/E command is issued. You can get servletKey from the response of the Issue TSO/E command API. The servletKey is only valid if you previously issued a stateful TSO/E command with Issue Command REST API. For the first command of a serial of stateful TSO/E commands, you do not specify servletKey. You get the value of servletKey from the response JSON. The servletKey is invalid if you do not specify cmdState as stateful.</p>

*Table 300. Supported parameters for the TSO/E command with z/OSMF REST API.*

The request content is expected to contain a JSON object. See [Table 300 on page 493](#) for a description of the fields.

(continued)

Parameter	Required or Optional	Description
<b>keyword</b>	Optional	Specifies a regular expression that you want to detect in the command response.  For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key: a[\s]+regular

**Note:**

- If you use a stateful Issue command API, you can use the servletKey to lock a TSO/E address space for following stateful commands/get response for 600 seconds. The address space, which is identified by servletKey will be released after 600 seconds if there's no further Issue stateful command/get response request from user.
- z/OSMF TSO/E services use CEA to create and maintain TSO/E address spaces. The default value of MAXSESSPERUSER for CEA is 10, which means CEA can create and maintain up to 10 TSO/E address spaces for a single user. The default value of MAXSESSIONS is 50, which means CEA can create and maintain up to 50 TSO/E address spaces on the system. You can issue command **f cea,d,parms** to check these values on your system. The maximum value for MAXSESSPERUSER is 99, and for MAXSESSIONS it is 2000. z/OSMF TSO/E services can maintain up to 45 TSO/E address spaces for stateless request, and up to 45 TSO/E address spaces for stateful request per user. You need to change the MAXSESSPERUSER to 99 and update the MAXSESSIONS to achieve 45/45 address spaces.
- For more information about how to change a CEA parameter, see [Working with TSO/E address spaces started by CEA in z/OS MVS Programming: Callable Services for High-Level Languages](#) and [CEAPRMxx \(common event adapter parameters\)](#) in [z/OS MVS Initialization and Tuning Reference](#).

## Required authorizations

To issue the TSO/E command and get response by the TSO/E address space services, you must have:

- – READ access to resource account in class ACCTNUM, where account is the value that is specified in the COMMON\_TSO ACCT option in parmlib.
- READ access to resource proc in class TSOPROC, where proc is the value that is specified with the COMMON\_TSO PROC option in parmlib.

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request is complete. Status code 200 indicates success. A status code of 4nn or 5nn indicates an error. For more information, see [“Error handling” on page 484](#).

*Table 301. Response content for a successful issue command request*

Field name	Description
cmdResponse	Command response in a JSON array.

Table 301. Response content for a successful issue command request (continued)	
Field name	Description
servletKey	Unique identifier for the servlet entry. It maps to the TSO/E address space in which the TSO/E command is issued. servletKey is returned only when cmdState is stateful.
tsoPromptReceived	Whether the TSO PROMPT sign is received in the command response: <b>Y</b> TSO PROMPT is received. <b>N</b> TSO PROMPT is not received yet.
keywordDetected	The result of the response detection request. This is returned when the keyword is specified. The values are: <b>Y</b> Matching record in the response was found. <b>N</b> Matching record in the response was not found.

If a failure occurs, the response body contains a JSON object that describes the error.

Table 302. Response JSON object for an unsuccessful issue command request	
Field name	Description
returnCode	Identifies the category of error.
returnCode	Identifies the specific error.
reason	Text that describes the cause of the error.

## HTTP status codes

For a successful request, HTTP status code 200 is returned, and the response body is provided, as described in [Table 301 on page 495](#).

For unsuccessful requests, the service returns the status codes that are described in [Table 302 on page 496](#).

Table 303. Response codes for unsuccessful issue command requests				
HTT P Stat us	Return code	Reaso n code	Reason	Description
500	8	4	An error occurred in the TSO/E address space. The error description: %s	The request failed because an error occurred. The error description is provided in the message text: %s. To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.

Table 303. Response codes for unsuccessful issue command requests (continued)

HTTP Status	Return code	Reason code	Reason	Description
500	8	12	The system cannot get the local node name and cannot start the TSO/E address space.	The attempt to prepare a TSO/E address space failed. The z/OSMF TSO/E service failed to retrieve the local node name of the system. Retry the request. If the problem persists, contact your z/OSMF administrator.
400	8	13	Unsupported Encoding Exception: %s	The system cannot support %s encoding. For more information, check the z/OSMF logs. Retry the request. If the problem persists, contact the IBM Support Center and provide the error details.
400	8	14	An error occurred when parsing the TSO/E response data.	The z/OSMF TSO/E service failed to parse the TSO/E response data. For more information, check the z/OSMF logs. Retry the request. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	16	An I/O Exception occurred when parsing the issue command request JSON body.	The z/OSMF TSO/E service failed to parse the issue command request JSON body. For more information, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	17	The TSO/E API cannot recognize the json field: %s	The JSON field %s in the request body, is not a supported field.
500	8	18	The tsoCmd cannot be null.	The JSON field “tsoCmd” in the request body cannot be null.
500	8	19	The maxWaitTime must be a positive integer between 1 to 300, cannot be %d.	The value of the “maxWaitTime” in the request JSON body must be a positive integer 1 - 300 .
500	8	21	The servletKey cannot be null.	The JSON field “servletKey” in the request body cannot be null.

Table 303. Response codes for unsuccessful issue command requests (continued)

HTT P Stat us	Return code	Reason code	Reason	Description
500	8	22	The maximum number of TSO/E address spaces that are allowed for the current user has been reached.	Refer to message IZUG1127E for a detailed explanation.  Display the active TSO/E address spaces and remove or cancel any address spaces that the user no longer needs.  To display the active TSO/E address spaces, enter the command <b>D TS,ALL</b> from the operator console.  To cancel a TSO/E address space, issue the <b>C u=user-ID,a=ASID</b> command from the operator console, where user-ID is the user's TSO/E ID and ASID is the address space identifier.
500	8	23	No TSO/E address space exist for servletKey:%s. Check your servletKey.	No TSO/E address space exists for servletKey:%s. Check your servletKey in the request JSON body.
500	8	25	Your attempt to start a TSO/E address space for your command has timed out. Try again later.	Your attempt to start a TSO/E address space for your command timed out. Try again later.
500	8	26	The TSO/E address space is temporarily unavailable. Refer to IZUG1113E for details and try again later.	The TSO/E address space is temporarily unavailable. Refer to IZUG1113E for details and try again later.
500	8	27	The TSO/E address space cannot be created, %s.	The TSO/E address space cannot be created. The context of the error is provided in the message text: %s. Refer to IZUG1117E for details.
500	8	28	System I/O exception.	System I/O exception. To obtain more details about the error, check the z/OSMF logs.
500	8	29	The TSO/E address space could not be created because an error occurred with the logon procedure or the user settings.	Verify that the logon procedure exists and is valid. For more information, see message IZUG1121E.
500	8	30	The TSO/E address space for the request cannot be found.	The TSO/E address space for the request cannot be found. For more information, check the z/OSMF logs.

Table 303. Response codes for unsuccessful issue command requests (continued)

HTT P Stat us	Return code	Reaso n code	Reason	Description
500	8	31	The maximum number of TSO/E address spaces for the system has been reached.	<p>Refer to IZUG1105E for a detailed explanation.</p> <p>Display the active TSO/E address spaces and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command D TS, ALL from the operator console.</p> <p>To cancel a TSO/E address space, enter the command <b>C u=user-ID,a=ASID</b> from the operator console, where user-ID is the user's TSO/E ID and ASID is the address space identifier.</p>
500	8	32	Failed to create a TSO/E address space. TSO/E user account number has not been defined for use, %s	<p>Refer to message IKJ56486I for a detailed explanation.</p> <p>One of the following scenarios occurred:</p> <ul style="list-style-type: none"> <li>• The specified account number is not defined to the RACF database. <ul style="list-style-type: none"> <li>– The RACF administrator must first define the account number as a RACF resource and then give the user access that uses the PERMIT command. However, if the procedure is not in the procedure library, the logon attempt continues to fail.</li> </ul> </li> <li>• The RACF class ACCTNUM is not active. <ul style="list-style-type: none"> <li>– The RACF administrator must activate the RACF class.</li> </ul> </li> </ul>
500	8	33	Failed to create a TSO/E address space. TSO/E user account number has not been authorized for the user, %s	<p>Refer to message IKJ56487I for a detailed explanation.</p> <p>The specified account number is defined to the RACF database. However, this particular user ID is not allowed to use it.</p>
500	8	34	Failed to create a TSO/E address space. TSO/E user account number is invalid: %s	<p>Refer to message IKJ56702I for a detailed explanation. The specified account number is incorrect.</p>

Table 303. Response codes for unsuccessful issue command requests (continued)

HTT P Stat us	Return code	Reason code	Reason	Description
500	8	35	Failed to create a TSO/E address space. Following messages returned by system: %s	z/OSMF TSO/E service failed to create TSO/E address space. The context of the error is provided in the message text:  %s. To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	36	Failed to create a TSO/E address space. %s	z/OSMF TSO/E service failed to create TSO/E address space due to the region size exceeds the limit size. The context of the error is provided in the message text: %s.  To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	8	37	The system you specified is incorrect.	Specify your request with the correct name of the target system and try again.
500	8	38	The maximum number of TSO/E address spaces that are allowed for the current user to issue a stateless command has been reached.	For a single user, you can have up to 45 active TSO/E address space to issue a stateless command.
500	8	39	The maximum number of TSO/E address spaces that are allowed for the current user to issue a stateful command has been reached.	For a single user, you can have up to 45 active TSO/E address space to issue stateful command.

## Example

In the following example, the PUT method is used to issue a TSO/E command **TIME** and get corresponding response. On completion, the command response is returned to the user.

```
Request:
PUT https:// your.company.com/zosmf/tsoApp/v1/tso

Request body:
{
  "tsoCmd" : "TIME"
}

Response:
HTTP/1.1 200 OK
{
  "cmdResponse": [
    {
      "message": "TIME-07:35:12 AM. CPU-00:00:00 SERVICE-21 SESSION-00:00:01 APRIL 6,2021"
    },
  ],
}
```

```

    {
      "message": "READY "
    }
  ],
  "tsoPromptReceived": "Y"
}

```

## Get the response to a command that was issued with the TSO/E REST API

This API gets a response for the stateful TSO/E command. This means you set cmdState to stateful for a previous Issue command REST API.

### HTTP method and URI path

```
GET /zosmf/tsoApp/{version}/tso
```

Where:

- zosmf/tsoApp identifies the TSO/E address space services.
- {version} identifies the version of the TSO/E REST API service. The following value is valid: v1.
- tso informs the service that the request is for a TSO/E address space.

### Query parameters

Table 304.		
Parameter	Required or Optional	Description
servletKey	Required	A unique identifier for the servlet entry. It maps to the TSO/E address space in which the TSO/E stateful command is issued. You can get a servletKey from the response of the Issue TSO/E command API.
maxWaitTime	Optional	<p>Specifies the max amount of time in seconds that the Get response API continue to get responses. The API tries to get as many responses as possible until a certain amount of time elapses or the TSO PROMPT is received. TSO PROMPT indicates that the TSO/E address space is ready for more input from the client. The input can either be a new command or the next part of the previous command. There's no further response for a TSO/E command after the TSO PROMPT is received.</p> <ul style="list-style-type: none"> <li>• If you do not specify the value for maxWaitTime, the API gets as many messages as possible until any of the following three things happen:             <ul style="list-style-type: none"> <li>– The TSO PROMPT is received.</li> <li>– The API returns no new response during the last 5 seconds.</li> <li>– 30 seconds elapses.</li> </ul> </li> <li>• The minimal value is 1 second.</li> <li>• The maximum value is 300, which means 5 mins.</li> </ul>

Table 304. (continued)		
Parameter	Required or Optional	Description
keyword	Optional	Specifies a regular expression that you want to detect in the command response.  For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key: a[\s]+regular
sourceCmd	Optional	Specifies whether the API returns the TSO/E command, which was issued previously by Issue command request. Case is not significant.  <b>Y</b> Return tsoCmd in the response JSON.  <b>N</b> Do not return tsoCmd in the response JSON. This is the default.

## Required authorizations

See the Required authorizations section of [“Issue a TSO/E command with z/OSMF REST API”](#) on page 492.

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request is complete. Status code 200 indicates success. A status code of 4nn or 5nn indicates an error.

Table 305. Response content for a successful get response request	
Field	Description
cmdResponse	Command response in a JSON array.
tsoPromptReceived	Whether the TSO PROMPT sign is received in the command response:  <b>Y</b> TSO PROMPT is received.  <b>N</b> TSO PROMPT is not received yet.
keywordDetected	Result of the response detection request. Returned when keyword is specified. The values are:  <b>Y</b> Matching record in the response was found.  <b>N</b> Matching record in the response was not found.
sourceCmd	The TSO/E command, which was issued by the Issue TSO/E command request previously. Returned when sourceCmd is Y.

If a failure occurs, the response body contains a JSON object that describes the error.

Table 306. Response JSON object for an unsuccessful issue command request	
Field	Description
returnCode	Identifies the category of the error.
reasonCode	Identifies the specific error.
reason	Text that describes the cause of the error.

## HTTP status codes

For a successful request, the HTTP status code 200 is returned and the response body is provided, as is described in [Table 305 on page 502](#).

For unsuccessful requests, see the HTTP status codes section within [“Issue a TSO/E command with z/OSMF REST API” on page 492](#).

## Example

In this example, you start a REXX script with the stateful PUT `/zosmf/tsoApp/{version}/tso API` and get part of the script output first. You then get the rest of the output with GET `/zosmf/tsoApp/{version}/tso API`.

- REXX script example: ZOSMF.JINGHUA.REXX(RXSAY)

```
— /* REXX */
/******
/*
/* Invoke like
/* exec 'ZOSMF.JINGHUA.REXX(RXSAY)' '10 1'
/*
/******
parse arg loopCnt sleepTime
do i = 1 to loopCnt
  say i "of" loopCnt
  CALL SYSCALLS('ON') /*ENABLE USS-CALLS*/
  ADDRESS SYSCALL
  "SLEEP" sleepTime /*SLEEP FOR ?? SECONDS*/
  CALL SYSCALLS 'OFF'
end
```

- Issue command

```
— Request:
PUT https:// your.company.com/zosmf/tsoApp/v1/tso

Request body:
{
  "tsoCmd" : "exec 'ZOSMF.JINGHUA.REXX(RXSAY)' '10 1'",
  "maxWaitTime" : "1",
  "cmdState" : "stateful"
}
Response:
HTTP/1.1 200 OK
{
  "cmdResponse": [
    {
      "message": "1 of 10"
    },
    {
      "message": "2 of 10"
    }
  ],
  "servletKey": "ZOSMFT1-31-aaauaaad",
  "tsoPromptReceived": "N"
}
```

- Get response.

– GET /zosmf/tsoApp/v1/tso?servletKey=ZOSMFT1-31-aaauaad&maxWaitTime=15

Response body:

```
{
  "cmdResponse": [
    {
      "message": "3 of 10"
    },
    {
      "message": "4 of 10"
    },
    {
      "message": "5 of 10"
    },
    {
      "message": "6 of 10"
    },
    {
      "message": "7 of 10"
    },
    {
      "message": "8 of 10"
    },
    {
      "message": "9 of 10"
    },
    {
      "message": "10 of 10"
    },
    {
      "message": "READY "
    }
  ],
  "tsoPromptReceived": "Y"
}
```

## Send messages to a TSO/E address space

You can use this operation to send TSO/E messages to a TSO/E address space.

### HTTP method and URI path

---

```
PUT /zosmf/tsoApp/tso/<servletKey>?[readReply=true|false]
```

---

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space to which the message will be sent.
- **[readReply]** is an optional parameter that indicates whether the service should send the message and immediately check for a response (default) or just send the message. To immediately check for a response, omit the parameter or set its value to *true*. Otherwise, set its value to *false*.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

## Request content

Your request must include a JSON object that describes the message to be sent. For example, the TSO/E message JSON format has the following syntax:

```
{ "message-type": { "VERSION": "JSON-version", "data-type": "data-value" } }
```

where:

### message-type

Keyword that identifies the type of TSO/E message. The value can be TSO MESSAGE, TSO PROMPT, or TSO RESPONSE.

### JSON-version

A four-digit number that identifies the JSON version used to format the message.

### data-type

Keyword that describes the type of data included in the *data-value* variable. The value can be DATA, HIDDEN, or ACTION.

Example: { "TSO RESPONSE": { "VERSION": "0100", "DATA": "ALLOC DA" } }

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

In addition, note that the API will attempt to read response TSO/E messages after the input message is sent. If no TSO/E messages are received after a predetermined time period, a timeout indication will be returned.

## Required authorizations

See [“Required authorizations”](#) on page 482.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 484.

The response also includes a JSON object that contains the TSO/E response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data”](#) on page 483.

## Example

To send a TSO/E message to the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaaaf* and read the response TSO/E messages, submit the request depicted in [Figure 256](#) on page 505.

```
PUT /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaaaf HTTP/1.1
Host: zosmf1.yourco.com

{ "TSO RESPONSE": { "VERSION": "0100", "DATA": "TIME" } }
```

*Figure 256. Sample request to send a message to a TSO/E address space*

A sample response is shown in [Figure 257](#) on page 506.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaaaf","ver":"0100","tsoData":[{"TSO MESSAGE":
{"VERSION":"0100","DATA":"TIME-12:09:07 PM. CPU-00:00:00 SERVICE-92319
SESSION-00:00:13 OCTOBER 12,2011"}]}],"timeout":false}
```

*Figure 257. Sample response from send message to TSO/E address space request*

## Send messages to an application

You can use this operation to send messages to an application running in a TSO/E address space.

### HTTP method and URI path

---

```
PUT /zosmf/tsoApp/app/<servletKey>/<appKey>
```

---

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **app** informs the service that the request is for an application running in a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space where the application is running.
- **<appKey>** identifies the application to which to send messages.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

Your request must include a JSON object that contains the application message to be sent.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, note that the API will attempt to read application and TSO/E response messages after the application input message is sent. If no messages are received in the time allotted, a timeout indication will be returned.

Any TSO/E messages are prioritized over application messages. Typically, when a caller receives a TSO/E message while attempting to receive application messages, the caller processes the TSO/E messages, then attempts to retrieve the queued application messages.

### Required authorizations

See [“Required authorizations” on page 482](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 484](#).

The response also includes a JSON object that contains the application response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

To send a message to application *BkApp001*, which is running in the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 258 on page 507](#).

```
PUT /zosmf/tsoApp/app/ZOSMFAD-71-aabcaaf/BkApp001 HTTP/1.1
Host: zosmf1.yourco.com

{...}
```

*Figure 258. Sample request to send a message to an application*

A sample response is shown in [Figure 259 on page 507](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","appData":[...],"timeout":false}
```

*Figure 259. Sample response from send message to an application request*

## Ping a TSO/E address space

You can use this operation to ping a TSO/E address space. Doing so at regular intervals helps to ensure that the TSO/E address space remains active for the client. Otherwise, the server can end the TSO/E address space without warning.

### HTTP method and URI path

```
PUT /zosmf/tsoApp/tso/ping/<servletKey>
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **ping** informs the service to ping the specified TSO/E address space.
- **<servletKey>** identifies the TSO/E address space for the service to ping.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

Each TSO/E address space has an idle application time that the TSO/E address space services interface uses to determine whether the client application that is associated with the address space is active. If the idle application time is 10 minutes, the client application is considered to be inactive. In which case, the API ends all the TSO/E address spaces associated with the client application.

To prevent TSO/E address spaces from ending because of idle application time, callers can issue a ping request at least once every 5 minutes. Doing so informs the TSO/E address space services interface that the client application is still active, and causes the interface to reset the idle application time for all the TSO/E address spaces associated with the client application.

## Required authorizations

See [“Required authorizations”](#) on page 482.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 484.

The response also includes a JSON object that contains all the attributes in the JSON structure except the message data. For more details, see [“Content type used for HTTP request and response data”](#) on page 483.

## Example

To ping the TSO/E address space identified by servlet key `ZOSMFAD-71-aabcaaaaf`, submit the request depicted in Figure 260 on page 508.

```
PUT /zosmf/tsoApp/tso/ping/ZOSMFAD-71-aabcaaaaf HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 260. Sample request to ping a TSO/E address space*

A sample response is shown in Figure 261 on page 508.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaaaf","ver":"0100","timeout":false}
```

*Figure 261. Sample response from ping TSO/E address space request*

## Receive messages from a TSO/E address space

You can use this operation to receive messages from a TSO/E address space.

### HTTP method and URI path

---

```
GET /zosmf/tsoApp/tso/<servletKey>
```

---

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space from which to receive messages.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

In addition, note that the API will attempt to read TSO/E messages. If no TSO/E messages are received after 15 seconds, a timeout indication will be returned.

### Required authorizations

See [“Required authorizations”](#) on page 482.

### Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 484.

The response also includes a JSON object that contains the TSO/E response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data”](#) on page 483.

### Example

To read TSO/E messages from the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaaaf*, submit the request depicted in [Figure 262](#) on page 509.

```
GET /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaaaf HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 262. Sample request to receive a message from a TSO/E address space*

A sample response is shown in [Figure 263 on page 510](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","tsoData":
[{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> LOGON proc version = 04/28/2011"}}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":{" "}}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> System Name = DCEIMGNE"}}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> System Suffix = NE"}}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> SYSPLEX Name = CFCIMGNE"}}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":"--> SYSRES Volume = SD1131"}}},
{"TSO MESSAGE":{"VERSION":"0100","DATA":{" "}}}
,"timeout":false}
```

*Figure 263. Sample response from receive message from a TSO/E address space request*

## Receive messages from an application

You can use this operation to receive messages from an application running in a TSO/E address space.

### HTTP method and URI path

```
GET /zosmf/tsoApp/app/<servletKey>/<appKey>
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **app** informs the service that the request is for an application running in a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space where the application is running.
- **<appKey>** identifies the application to which to send messages.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, note that the API will attempt to read application and TSO/E response messages. If no messages are received in the time allotted, a timeout indication will be returned.

Any TSO/E messages are prioritized over application messages. Typically, when a caller receives a TSO/E message while attempting to receive application messages, the caller processes the TSO/E messages, then attempts to retrieve the queued application messages.

## Required authorizations

See [“Required authorizations” on page 482](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 484](#).

The response also includes a JSON object that contains the application response messages, or a timeout indication. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

To receive TSO/E or application messages from application *BkApp001*, which is running in the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 264 on page 511](#).

```
GET /zosmf/tsoApp/app/ZOSMFAD-71-aabcaaf/BkApp001 HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 264. Sample request to receive messages from an application*

A sample response is shown in [Figure 265 on page 511](#).

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","appData":[...],"timeout":false}
```

*Figure 265. Sample response for request to receive messages from an application*

## End a TSO/E address space

You can use this operation to end a TSO/E address space or place it in a dormant state as a candidate for reconnection.

## HTTP method and URI path

```
DELETE /zosmf/tsoApp/tso/<servletKey>?[tsoforcecancel=true|false]
```

where:

- **zosmf/tsoApp** identifies the TSO/E address space services.
- **tso** informs the service that the request is for a TSO/E address space.
- **<servletKey>** identifies the TSO/E address space to be ended or placed in a dormant state.
- **[tsoforcecancel]** is an optional parameter that indicates whether to use the CANCEL or LOGOFF command to end the TSO/E address space. The parameter can have one of the following values:
  - **True:** The CANCEL command will be issued and the TSO/E address space will not be placed in a dormant state.
  - **False** (default): The LOGOFF command will be issued. If the CEA reconnect feature is enabled in your installation, the TSO/E address space will be placed in a dormant state. Otherwise, the TSO/E session will end.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Required authorizations

See [“Required authorizations”](#) on page 482.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 484.

The response also includes a JSON object with additional information about the results of the request. For more details, see [“Content type used for HTTP request and response data”](#) on page 483.

## Example: Logging off a TSO/E address space

To use the LOGOFF command to end the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 266](#) on page 512.

```
DELETE /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaf HTTP/1.1
Host: zosmf1.yourco.com
```

*Figure 266. Sample request to logoff a TSO/E address space*

A sample response is shown in [Figure 267](#) on page 512.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","timeout":false,"reuse":false}
```

*Figure 267. Sample response for logoff a TSO/E address space request*

## Example: Canceling a TSO/E address space

To use the CANCEL command to end the TSO/E address space identified by servlet key *ZOSMFAD-71-aabcaaf*, submit the request depicted in [Figure 268](#) on page 513.

```
DELETE /zosmf/tsoApp/tso/ZOSMFAD-71-aabcaaf?tsoforcecancel=true HTTP/1.1
Host: zosmf1.yourco.com
```

Figure 268. Sample request to cancel a TSO/E address space

A sample response is shown in Figure 269 on page 513.

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2011 05:39:28 +0000GMT
Connection: close

{"servletKey":"ZOSMFAD-71-aabcaaf","ver":"0100","timeout":false,"reuse":true}
```

Figure 269. Sample response for a cancel TSO/E address space request

## WLM resource pooling services

The WLM resource pooling services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. The WLM resource pooling services provide a programming interface for WLM policy elements. You can work with WLM policy elements in the context of IBM Cloud Provisioning and Management for z/OS.

With the WLM resource pooling services, you can provision and deprovision WLM policy elements, dynamically construct a new service definition, and install the service definition.

Table 307 on page 513 lists the operations that the WLM resource pooling services provide.

Table 307. Operations provided through the WLM resource pooling services.

Operation	HTTP method and URI path
<a href="#">“Prime a WLM resource pool” on page 514</a>	POST /zosmf/zwlm/rest/wrps
<a href="#">“Delete a WLM resource pool” on page 516</a>	DELETE /zosmf/zwlm/rest/wrps/wrpid
<a href="#">“Construct a WLM service definition” on page 517</a>	PUT /zosmf/zwlm/rest/policy/inspolicy
<a href="#">“Construct a WLM service definition with remove and install” on page 519</a>	PUT /zosmf/zwlm/rest/policy/inspolicy

### Required authorizations

The user’s z/OS user ID must have READ access to the following resource profile in the ZMFAPLA class: <SAF-prefix>.ZOSMF.RESOURCE\_POOL.WLM.*domainid*, where *domainid* is the identifier of the domain of systems.

### Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. The HTTP status codes are described in the topics for the individual services.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

**HTTP 200 OK**

Request was processed successfully.

**HTTP 204 No Content**

Request was processed successfully.

**HTTP 400 Bad request**

Request could not be processed because it contains a syntax error or an incorrect parameter.

**HTTP 401 Unauthorized**

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

**HTTP 500 Internal server error**

Server encountered an error. See the response body for a JSON object with information about the error.

## Error logging

Errors from the WLM resource pooling services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Prime a WLM resource pool

Use this operation to create a record for a WLM resource pool.

### HTTP method and URI path

---

```
POST /zosmf/zwlm/rest/wips
```

---

### Query parameters

None.

### Description

This operation creates a WRP record in a persistent file for WRP data. In addition, it causes a new WRP record to be displayed on the WLM Resource Pool page of the z/OSMF Workload Management task. The actual provisioning for the report class, and the definition and installation of the service definition, occurs only when the WRP definition is completed by the WLM administrator in z/OSMF.

On successful completion of a prime a WLM resource pool request, a response body that describes the request is returned.

For the properties that you can specify on the request body, see [“Request content” on page 514](#).

For a description of the response content, see [“Response content” on page 515](#).

### Request content

The request content is expected to contain a JSON object. See [Table 308 on page 515](#) for a description of the fields.

Table 308. Request content for the prime WLM resource pool request		
Field name	Required or Optional	Description
<b>cloud-info</b>	Required	Specifies the attributes of the cloud: <b>domain-name</b> Name of the domain of systems <b>domain-id</b> Generated identifier of the domain of systems <b>tenant-name</b> Name of the tenant for the domain <b>tenant-id</b> Generated identifier of the tenant <b>template-name</b> Name of the software services template
<b>wrp-data</b>	Required	Specifies the attributes of the WLM resource pool: <b>wrp-name</b> Name of the WLM resource pool <b>service-level-agreements</b> Array of service-level agreements, in the form "sla-name":"level", for example: <pre>"service-level-agreements": [   {"SlaOne":"GOLD"},   {"SlaTwo":"SILVER"} ]</pre> <b>report-class-name</b> Name of the report class

## Authorization requirements

See [“Required authorizations”](#) on page 513.

## HTTP status codes

For a successful request, the response body is provided, as described in [“Request content”](#) on page 514.

For a list of status codes, see [“Error handling”](#) on page 513.

## Response content

On successful completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 309](#) on page 515.

Table 309. Response content for a successful prime WLM resource pool request	
Field name	Description
<b>status</b>	Status of the request.
<b>return-code</b>	Return code of the request.
<b>message</b>	Message issued for the request.
<b>wrp-id</b>	Identifier of the WLM resource pool
<b>state</b>	State of the request.

## Example HTTP interaction

1. The example in [Figure 270 on page 516](#) shows a request to prime a WLM resource pool.

```
POST https://host:port/zosmf/zwlm/rest/wrps
{
  "cloud-info":{
    "domain-name":"DOMAIN1",
    "domain-id":"12345ABC",
    "tenant-name":"Joey",
    "tenant-id":"IZU$000",
    "template-name":"CICSBasic"
  }

  "wrp-data" {
    "wrp-name":"WRP1",
    "service-level-agreements":[
      {"sla-name":"GOLD"},
      {"sla-name":"SILVER"}
    ],
    "report-class-name":"Joey00"
  }
}
```

Figure 270. Sample request to issue a prime WRP request

The following is the response body for the request.

```
{
  "status":"success",
  "return-code":"0",
  "message":null,
  "wrp-id":"1090f34e-0a5a-4506-b553-91e932a46f3e",
  "wrp-name":"WRP1",
  "state":"initialized"
}
```

Figure 271. Sample response body

## Delete a WLM resource pool

Use this operation to delete the record for a WLM resource pool.

### HTTP method and URI path

```
DELETE /zosmf/zwlm/rest/wrps/wrpid
```

In this request, *wrpid* is the identifier of the WLM resource pool.

### Query parameters

None.

### Description

This operation:

- Removes a WLM resource pool record from the persistent file for WLM resource pools
- Deprovisions the report class in the current installed service definition if it is not referenced by any other classification rule
- Deletes the WLM resource pool record from the WLM Resource Pool page of the z/OSMF Workload Management task.

On successful completion, a response body that describes the request is returned.

For a description of the response content, see [“Response content” on page 517](#).

## Request content

None.

## Authorization requirements

See [“Required authorizations” on page 513](#).

## HTTP status codes

For a successful request, the response body is provided, as described in [“Request content” on page 517](#).

For a list of status codes, see [“Error handling” on page 513](#).

## Response content

On completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 310 on page 517](#).

Table 310. Response content for a delete WLM resource pool request	
Field name	Description
<b>status</b>	Status of the request.
<b>return-code</b>	Return code of the request.
<b>message</b>	Message issued for the request.

## Example HTTP interaction

1. The example in [Figure 272 on page 517](#) shows a request to delete a WLM resource pool.

```
DELETE https://host:port/zosmf/zwlm/rest/wrps/1090f34e-0a5a-4506-b553-91e932a46f3e
```

*Figure 272. Sample request to issue a delete WLM resource pool request*

The following is the response body for the successful request.

```
{
  "status": "success",
  "return-code": "0",
  "message": null
}
```

*Figure 273. Sample response body*

## Construct a WLM service definition

Use this operation to construct a new service definition based on the current installed service definition.

## HTTP method and URI path

```
PUT /zosmf/zwlm/rest/policy/inspolicy
```

## Query parameters

None.

## Description

This operation constructs a new service definition based on the current installed service definition. A description for the report class is generated based on the domain name and domain ID.

On successful completion, a response body that describes the request is returned.

For the properties that you can specify on the request body, see [“Request content” on page 518](#).

For a description of the response content, see [“Response content” on page 518](#).

## Request content

The request content is expected to contain a JSON object. See [Table 311 on page 518](#) for a description of the fields.

Table 311. Request content for the construct a WLM service definition request		
Field name	Required or Optional	Description
<b>cloud-info</b>	Required	Specifies attributes of the cloud: <b>WRP-ID</b> Identifier of the WLM resource pool.
<b>provision-data</b>	Required	Specifies the attributes of the WLM service definition: <b>classification-rules</b> Array of attributes, in the form "attribute-name":"value", for example: <pre>"classification-rules":   [     {"service-level-agreement":"Gold",      "qualifier-value":"CICSL00",      "report-class": "optional"}   ]</pre>

## Authorization requirements

See [“Required authorizations” on page 513](#).

## HTTP status codes

For a successful request, the response body is provided, as described in [“Request content” on page 518](#).

For a list of status codes, see [“Error handling” on page 513](#).

## Response content

On successful completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 312 on page 518](#).

Table 312. Response content for a successful construct a WLM service definition request	
Field name	Description
<b>status</b>	Status of the request.
<b>messages</b>	Message issued for the request.

Table 312. Response content for a successful construct a WLM service definition request (continued)

Field name	Description
<b>result</b>	<p>Result for the request.</p> <p><b>classification-rules</b> Array of attributes, in the form "cl-rule-id"."value", for example:</p> <pre>"classification-rules":   [     {"cl-rule-id": "id"}   ]</pre>

## Example HTTP interaction

1. The example in [Figure 274 on page 519](#) shows a request to construct a service definition based on the current installed definition.

```
PUT https://host:port/zosmf/zwlm/rest/policy/inspolicy
{
  "cloud-info": {
    "WRP-ID": "e5697dd6-88da-43f8-8f89-
bbdf9537b296",
  },
  "provision-data": {
    classification-rules: [
      { "service-level-agreement": "Gold",
        "qualifier-value": "CICSL00",
        "report-class": "optional" }
    ]
  }
}
```

Figure 274. Sample request to construct a service definition based on the current installed definition

The following is the response body for the request.

```
{
  "state": "success",
  "return-code": "0",
  "message": null,
  "result": {
    "classification-rules": [
      { "cl-rule-id": "1090f34e-0a5a-4506-b553-91e932a46f3e" }
    ]
  }
}
```

Figure 275. Sample response body

## Construct a WLM service definition with remove and install

Use this operation to construct a new service definition by removing the classification rule, then installing the new service definition.

### HTTP method and URI path

```
PUT /zosmf/zwlm/rest/policy/inspolicy
```

### Query parameters

None.

## Description

This operation constructs a new service definition by removing the classification rule, then installing the new service definition.

On successful completion, a response body that describes the request is returned.

For the properties that you can specify on the request body, see [“Request content” on page 520](#).

For a description of the response content, see [“Response content” on page 520](#).

## Request content

The request content is expected to contain a JSON object. See [Table 313 on page 520](#) for a description of the fields.

Table 313. Request content for the construct a WLM service definition request		
Field name	Required or Optional	Description
<b>cloud-info</b>	Required	Specifies cloud-related attributes: <b>WRP-ID</b> Identifier of the WLM resource pool
<b>deprovision-data</b>	Required	Specifies the attributes of the WLM service definition: <b>classification-rules</b> Array of attributes, in the form "attribute-name":"value", for example: <pre>"classification-rules":   [     {"cl-rule-id":"value"}   ]</pre>

## Authorization requirements

See [“Required authorizations” on page 513](#).

## HTTP status codes

For a successful request, the response body is provided, as described in [“Request content” on page 520](#).

For a list of status codes, see [“Error handling” on page 513](#).

## Response content

On successful completion, the service returns a response body, which contains a JSON object. For a description of fields in the JSON object, see [Table 314 on page 520](#).

Table 314. Response content for a successful construct a WLM service definition request	
Field name	Description
<b>state</b>	State of the request.
<b>messages</b>	Message issued for the request.

## Example HTTP interactions

1. The example in [Figure 276 on page 521](#) shows a request to construct a service definition by removing the classification rule, then installing the new service definition.

```
PUT https://host:port/zosmf/zwlm/rest/policy/inspolicy
{
  "cloud-info": {
    "WRP-ID": "e5697dd6-88da-43f8-8f89-
bbdf9537b296",
  }
  "deprovision-data": {
    "classification-rules": [
      {
        "cl-rule-id": "1090f34e-0a5a-4506-b553-91e932a46f3e"
      }
    ]
  }
}
```

Figure 276. Sample request to construct a service definition by removing the classification rule, then installing the new service definition

The following is the response body for the request.

```
{
  "state": "success",
  "message": null,
}
```

Figure 277. Sample response body

## RMF metering services

The Resource Measurement Facility REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. This interface allows a client application to interact with z/OSMF.

Table 315 on page 521 lists the operations that the RMF services provide.

Table 315. Operations provided through the RMF services	
Operation	HTTP method and URI path
<b>“Get metered data” on page 522</b>	GET /zosmf/izur/meterdata
<b>“Get DDS server time data” on page 525</b>	GET /zosmf/izur/rest/ddstime

### Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document” on page 640](#).

The following HTTP status codes are valid:

#### HTTP 200 OK

Request was processed successfully.

#### HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

### HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

### HTTP 404 Not found

Requested resource does not exist.

### HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

## Error logging

Errors from the z/OSMF notifications services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Get metered data

Use this operation to retrieve your metered data.

### HTTP method and URI path

```
GET /zosmf/izur/meterdata
```

### Query parameters

Table 316. Query parameters		
Parameter	Description	Rule
TRG	Tenant resource group name.	Must contain one to eight characters, starting with an alphabetic character.
TRGLACS	Long-term average of CPU service (millions of service units) consumed by a tenant resource group. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGCP	Service units on general purpose processors consumed by a tenant resource group per second. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGAAP	Service units on zAAPs consumed by a tenant resource group per second. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGIIP	Service units on zIIPs consumed by a tenant resource group per second. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGCPN	General purpose processor consumption in number of CPs. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.

<i>Table 316. Query parameters (continued)</i>		
Parameter	Description	Rule
TRGAAPN	zAAP processor consumption in number of CPs. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
TRGIIPN	zIIP processor consumption in number of CPs. This parameter is optional.	Must contain one to eight characters, starting with an alphabetic character.
date	Specifies the start and end date/time of the RMF reporting interval.	Must be in the format of: startDate: yyyyymmdd endDate: yyyyymmdd
timeofday	The start time and end time based on the date of the operation to retrieve your metered data.	Must be in the format of: startTime: hhmm endTime: hhmm
duration	The length of the RMF reporting interval.	Must be in the format of: duration: hhmm

## Description

This operation retrieves your metered data.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in metered data being retrieved, and a response body is returned. See [Figure 279 on page 524](#).

## Request content

None.

## Authorization requirements

Provide authorization level. For example, "User" or "Superuser". If there is no authorization level, specify "None".

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

<i>Table 317. HTTP error response codes for a get metered data request</i>	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request body is missing a field.
<b>HTTP 404 Not found</b>	The requested metered data does not exist.
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

### Example HTTP interaction

In Figure 278 on page 524, a request is submitted to retrieve your metered data.

```
GET https://zosmfhost:1035/zosmf/izur/meterdata?TRG=TRG1&TRGLACS=TRGLACS1&TRGCP=TRGCP1&TRGAAP=TRGAAP1&TRGIIP=TRGIIP1&TRGCPN=TRGCPN1&TRGAAPN=TRGAAPN1&TRGIIPN=TRGIIPN1&date=20170724,20170724&timeofday=0500,2400&duration=0100
```

Figure 278. Sample request to get metered data

The following Figure 279 on page 524, is the response body for the example GET metered data request.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="include/ddssml-pp.xsl"?>
<ddssml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="include/ddssml.xsd">
  <server>
    <name>RMF-DDS-Server</name>
    <version>Z0SV2R2</version>
    <functionality>3206</functionality>
    <platform>z/OS</platform>
  </server>
  <postprocessor><metric id="OVW"><description>RMF Overview Report</description>
    <type>Interval</type></metric><version><smf-data>z/OS V2R2</smf-data><rmf-report>
    z/OS V2R2</rmf-report></version><resource><resname>TRX1</resname><restype>SYSTEM</restype>
    </resource><time-data><display-start locale="en-us">07/31/2017-09.34.41</display-start>
    <display-end locale="en-us">07/31/2017-09.44.34</display-end><report-interval
    unit="hours">000:09:51</report-interval><cycle unit="milliseconds">1000</cycle>
    </time-data>
    <segment id="1"><name>Overview Report</name>
    <part id="2">
      <var-list id="3"><var><name>Number of Intervals</name><value>1</value></var><var>
      <name>Total Length of Intervals</name><value>00.09.51</value></var>
      </var-list>
      <table id="4">
        <column-headers><col type="T">Date (mm/dd)</col><col type="T">Time (hh.mm.ss)</col>
        <col type="T">Interval Length (hh.mm.ss)</col><col condition="TRGCP" qualifier="TRGRMFN1"
        type="N">TRGCP1</col>
        <col condition="TRGLACS" qualifier="TRGRMFN1" type="N">TRGLACS1</col><col condition="TRGAAP"
        qualifier="TRGRMFN1" type="N">TRGAAP1</col><col condition="TRGIIP" qualifier="TRGRMFN1"
        type="N">TRGIIP1</col></column-headers>
        <row refno="1"><col>07/31</col><col>09.34.41</col><col>00.09.51</col><col>0</col><col>0</col>
        <col>0</col></row>
      </table></part></segment>
    </postprocessor>
    <postprocessor><metric id="OVW"><description>RMF Overview Report</description><type>Interval</type>
    </metric><version><smf-data>z/OS V2R2</smf-data><rmf-report>z/OS V2R2</rmf-report></version><resource>
    <resname>T2</resname><restype>SYSTEM</restype>
    </resource><time-data><display-start locale="en-us">07/31/2017-09.35.06</display-start>
    <display-end locale="en-us">07/31/2017-09.44.34</display-end><report-interval unit="hours">000:09:26
    </report-interval><cycle unit="milliseconds">1000</cycle>
    </time-data>
    <segment id="1"><name>Overview Report</name>
    <part id="2">
      <var-list id="3"><var><name>Number of Intervals</name><value>1</value></var><var>
      <name>Total Length of Intervals</name><value>00.09.26</value></var>
      </var-list>
      <table id="4">
        <column-headers><col type="T">Date (mm/dd)</col><col type="T">Time (hh.mm.ss)</col>
        <col type="T">Interval Length (hh.mm.ss)</col><col condition="TRGCP" qualifier="TRGRMFN1"
        type="N">TRGCP1</col>
        <col condition="TRGLACS" qualifier="TRGRMFN1" type="N">TRGLACS1</col><col condition="TRGAAP"
        qualifier="TRGRMFN1" type="N">TRGAAP1</col><col condition="TRGIIP" qualifier="TRGRMFN1"
        type="N">TRGIIP1</col></column-headers>
        <row refno="1"><col>07/31</col><col>09.35.06</col><col>00.09.26</col><col>0</col><col>0</col>
        <col>0</col></row>
      </table></part></segment>
    </postprocessor>
  </ddssml>
```

Figure 279. Sample response

# Get DDS server time data

Use this operation to retrieve your DDS server time data.

## HTTP method and URI path

GET /zosmf/izur/rest/ddstime

## Description

This operation retrieves your DDS server time data.

On successful completion, HTTP status code 200 (OK) is returned, indicating that the request resulted in a DDS server time being retrieved, and a response body is returned. See [Figure 281 on page 526](#).

## Request content

None.

## Authorization requirements

Provide authorization level. For example, "User" or "Superuser". For no authorization level, specify "None".

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code that is indicated and associated error message.

Table 318. HTTP error response codes for a get DDS server time data request

HTTP error status code	Description
HTTP 400 Bad request	The request body is missing a field.
HTTP 404 Not found	The requested DDS server time does not exist.
HTTP 500 Internal server error	The server encountered an error. See the response body for a JSON object with information about the error.

## Response content

## Example HTTP interaction

In [Figure 280 on page 525](#), a request is submitted to retrieve DDS server time data.

GET https://zosmfhost:1035/zosmf/izur/rest/ddstime

Figure 280. Sample request to get DDS server time data

The following [Figure 281 on page 526](#), is the response body for the example GET DDS server time data.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="include/ddsm1-m3.xsl"?>
<ddsm1 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="include/ddsm1.xsd">
  <server>
    <name>RMF-DDS-Server</name>
    <version>Z0SV2R3</version>
    <functionality>3211</functionality>
    <platform>z/OS</platform>
  </server>
  <report>
    <metric id="CFOVER">
      <description>CFOVER (Coupling Facility Overview)</description>
      <format>report</format>
      <numcols>33</numcols>
    </metric>
    <resource>
      <reslabel>,PLEX1,SYSPLEX</reslabel>
      <restype>SYSPLEX</restype>
      <reslabelurl>%2Cplex1%2CSysplex</reslabelurl>
    </resource>
    <time-data>
      <local-start>20170927030820</local-start>
      <local-end>20170927031000</local-end>
      <utc-start>20170927070820</utc-start>
      <utc-end>20170927071000</utc-end>
      <local-prev>20170927030730</local-prev>
      <local-next>20170927031050</local-next>
      <display-start locale="en-us">09/27/2017 03:08:20</display-start>
      <display-end locale="en-us">09/27/2017 03:10:00</display-end>
      <gatherer-interval unit="seconds">100</gatherer-interval>
      <data-range unit="seconds">100</data-range>
    </time-data>
    <caption><var><name>CFOHPNAM</name><value>CTTEST1</value></var><var><name>CFOHPACD</name><value>03/08/01</value>
    </var><var><name>CFOHPACT</name><value>19.13.33</value></var><var><name>CFOHPREF</name><value>N0</value></var></caption>
    <row refno="1"><col>TESTCFNM</col> <col>MDEV</col> <col>001</col> <col>8</col> <col>N/A</col> <col>0kay</col>
    <col>YES</col> <col>0.0</col> <col>1</col> <col>N/A</col> <col>N/A</col> <col>N/A</col> <col>1.0</col> <col>2.5</col>
    <col>6815744</col> <col>0</col> <col>100</col> <col>6815744</col> <col>0</col> <col>262144</col> <col>0</col> <col>1</col>
    <col>4</col> <col>0</col> <col>0FF</col> <col>0FF</col> <col>0</col> <col>0</col> <col>0.0</col> <col>0</col> <col>0</col>
    <col>0.0</col> <col>0</col></row>
    <column-headers> <col type="T">CFOPNAM</col> <col type="T">CFOPMOD</col> <col type="N">CFOPVER</col> <col type="N">CFOPLVL
    </col> <col type="T">CFOPDYND</col> <col type="T">CFOPSTAT</col> <col type="T">CFOPVOL</col> <col type="N">CFOPUTIP</col>
    <col type="N">CFOPDEF</col> <col type="T">CFOPPEDE</col> <col type="T">CFOPPSHR</col> <col type="T">CFOPPGWT</col>
    <col type="N">CFOPPEFF</col> <col type="N">CFOPREQR</col> <col type="N">CFOPTSD</col> <col type="N">CFOPTSF</col>
    <col type="N">CFOPUTIS</col> <col type="N">CFOPTCS</col> <col type="N">CFOPFCS</col> <col type="N">CFOPDTS</col>
    <col type="N">CFOPDTUS</col> <col type="N">CFOPSYSC</col> <col type="N">CFOPSTCI</col> <col type="N">CFOPSTCO</col>
    <col type="T">CFOPMNT</col> <col type="T">CFOPRCV</col> <col type="N">CFOPSCMS</col> <col type="N">CFOPSCMA</col>
    <col type="N">CFOPSCMU</col> <col type="N">CFOPAUGS</col> <col type="N">CFOPAUGA</col> <col type="N">CFOPAUGU</col>
    <col type="N">CFOPMSC</col> </column-headers>
  </report>
</ddsm1>

```

Figure 281. Sample response

## z/OS console services

The z/OS console services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. The z/OS console services provide a programming interface for performing z/OS console operations.

With the z/OS console services, you can issue system commands and work with both solicited messages (messages that were issued in response to the command) and unsolicited messages (other messages that might or might not have been issued in response to the command). z/OS console services establish an extended MCS (EMCS) console, which is then used to issue commands and receive messages.

Table 319 on page 526 lists the operations that the z/OS console services provide.

Table 319. Operations provided through the z/OS console services.	
Operation	HTTP method and URI path
<b>“Issue a command from a system console” on page 528</b>	PUT /zosmf/restconsoles/consoles/consolename PUT /zosmf/restconsoles/consoles/defcn
<b>“Get a command response” on page 545</b>	GET /zosmf/restconsoles/consoles/console-name/solmsgs/Ckey-number GET /zosmf/restconsoles/consoles/defcn/solmsgs/Ckey-number

Table 319. Operations provided through the z/OS console services. (continued)

Operation	HTTP method and URI path
<b><u>“Get the detect result for unsolicited messages” on page 551</u></b>	GET /zosmf/restconsoles/consoles/ <i>consolename</i> /detections/ <i>Dkey-number</i> GET /zosmf/restconsoles/consoles/defcn/detections/ <i>Dkey-number</i>
<b><u>“Get messages from logs” on page 558</u></b>	GET /zosmf/restconsoles/log

## Using the Swagger interface

You can use the Swagger interface to display information about the z/OS console services REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

## Required authorizations

Your user ID must have the same authority when issuing a command with the z/OS console services as when issuing a command through a console on a z/OS system.

The required authority is:

- READ access to the MVS.MCSOPER.*consolename* resource in the OPERCMDS class, where *consolename* is the name of the EMCS console that is used to issue the command
- READ access to the CONSOLE resource in the TSOAUTH class
- READ access to the <SAF\_PREFIX>.\*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF\_PREFIX>.IzuManagementFacilityRestConsoles.izuUsers resource name in the EJBROLE class.

z/OS console services use z/OSMF TSO/E address space services to create a TSO address space as the host for an EMCS console. To use TSO/E address space services, you must have:

- READ access to resource *account* in class ACCTNUM, where *account* is the value specified in the COMMON\_TSO ACCT option in parmlib
- READ access to resource CEA.CEATSO.TSOREQUEST in class SERVAUTH
- READ access to resource *proc* in class TSOPROC, where *proc* is the value specified with the COMMON\_TSO PROC option in parmlib.

You must also ensure that the z/OSMF started task user ID, which is IZUSVR by default, has READ access to resource CEA.CEATSO.TSOREQUEST in class SERVAUTH.

The TSO/E address space services authority might already be defined if you are using z/OS data set and file REST services, as those services require similar authority.

## Configuration

z/OS console services use the TSO CONSOLE command to establish an EMCS console, which allows you to issue system commands and retrieve the messages that are issued in response. Console attributes such as ROUTCODE and AUTH affect the messages that the EMCS console can receive and the commands that the console can issue. When you use the z/OS console services, be sure that the EMCS console that is established has the desired attributes. For information, see [Extended MCS consoles](#) in *z/OS MVS Planning: Operations*.

In addition, be aware that messages can be suppressed due to settings in the active MPFLSTxx member of parmlib. If a message associated with a command response is suppressed, a REST API call that attempts to detect that message will fail.

To control the parameters that z/OS console services use when creating a TSO address space as the host for an EMCS console, use parmlib option COMMON\_TSO ACCT(IZUACCT) REGION(50000)

PROC(IZUFPROC). Configure this setting before z/OS console services are to be used. Otherwise, default values are used with z/OS console services.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. The HTTP status codes are described in the topics for the individual services.

In addition, a JSON object describes the error.

## Error logging

Errors from the z/OS console services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Issue a command from a system console

Use this operation to issue a command by using a system console.

### HTTP method and URI path

---

```
PUT /zosmf/restconsoles/consoles/consolename
PUT /zosmf/restconsoles/consoles/defcn
```

---

Where:

#### ***consolename***

Is the name of the EMCS console that is used to issue the command. The name must be 2 - 8 characters long, and cannot be defcn, which is reserved.

#### ***defcn***

Indicates that the name of the console that is used to issue the command is generated by the REST Console API, by adding CN to the logon user ID. For example, if the logon user ID is CJOEY, the console name is CJOEYCN. If the user ID is longer than 6 characters, the user ID is truncated. For example, if the user ID is ZOSMFAD, the console name is ZOSMFACN.

## Query parameters

None.

## Description

This operation issues a command, based on the properties that are specified in the request body. On successful completion, HTTP status code 200 is returned. A JSON object typically contains the command response.

When a command is issued synchronously, the console API attempts to get the solicited messages immediately after the command is issued. If there are no messages available within a certain time interval, approximately 3 seconds when your system workload is not high, the API returns "cmd-response": "" in the response body.

A value for cmd-response of the empty string, "", usually means that there is no command response. However, it is also possible that the command response arrived after 3 seconds. If that is the case, you can use the cmd-response-url field in the response body to retrieve the command response. You might do this several times to ensure that all messages that are related to the command are retrieved.

Alternatively, you might examine unsolicited messages, that is, additional messages that are not part of the command response. To do so, you can issue the command with option `unsol-key` to detect a keyword in the unsolicited messages.

For the properties that you can specify on the request body, see [“Request content” on page 529](#).

For a description of the response content, see [“Response content” on page 540](#).

The request body can include properties that you can use to control console attributes, such as 'auth', 'routcode', 'mscope', 'storage', and 'auto'. Alternatively, you can use System Authorization Facility or SAF to control console attributes. The RACF ADDUSER command with the OPERPARM parameter sets console attributes when a user establishes an EMCS console. Using the ADDUSER command to control console attributes requires that you know in advance the name of the EMCS console that the z/OS Console service uses. The console name is either a name that you specify on the Issue Command service, or a name that the service generates, as described in [“Issue a command from a system console” on page 528](#). For example, if user CJOEY plans to accept the default console name, CJOEYCN, this user can issue this RACF command to set console attributes for the console:

```
ADDUSER CJOEYCN OPERPARM(AUTH(MASTER) ROUTCODE(ALL))
```

**Note:** To use the properties 'auth', 'routcode', 'mscope', 'storage', or 'auto' in your request, your user ID requires at least READ access to resource CONOPER in class TSOAUTH.

## Request content

The request content is expected to contain a JSON object. See [Table 320 on page 529](#) for a description of the fields.

Table 320. Request content for the issue command request		
Field name	Required or Optional	Description
<b>cmd</b>	Required	Specifies the command to issue.
<b>sol-key</b>	Optional	<p>Specifies a keyword that you want to detect in solicited messages, that is, the command response. Case is not significant.</p> <p>This value can be a string or a regular expression. To use a regular expression, you must also set the solKeyReg property.</p>
<b>unsol-key</b>	Optional	<p>Specifies a keyword that you want to detect in unsolicited messages. Case is not significant.</p> <p>This value can be a string or a regular expression. To use a regular expression, you must also set the unsolKeyReg property.</p> <p>Message suppression can prevent the return of an unsolicited message. To determine whether a particular message ID is suppressed through the message processing facility on your system, enter the following command to list the MPF settings: <b>D MPF.</b></p>
<b>detect-time</b>	Optional	<p>Indicates how long the console attempts to detect the value of unsol-key in the unsolicited messages. The unit is seconds. For example, if the value of detect-time is 10, the console checks the unsolicited messages for 10 seconds. The default is 30 seconds.</p>

Table 320. Request content for the issue command request (continued)

Field name	Required or Optional	Description
<b>async</b>	Optional	<p>Indicates the method of issuing the command:</p> <p><b>Y</b> Asynchronously</p> <p><b>N</b> Synchronously.</p> <p>If you omit this property, N is used by default.</p>
<b>system</b>	Optional	<p>Name of the system in the same sysplex that the command is routed to. The default is the local system.</p>
<b>unsol-detect-sync</b>	Optional	<p>Indicates how to detect the keyword that is specified with the unsol-key field from unsolicited messages:</p> <p><b>Y</b> Synchronously detect the keyword from unsolicited messages. The request is not returned until the unsol-detect-timeout value elapses or the detection result is complete.</p> <p><b>N</b> Asynchronously detect the keyword from unsolicited messages. The request is returned immediately with the detection-url. The client application must invoke the value of detection-url to poll the result of the detection asynchronously.</p> <p>If you omit this property, N is used by default.</p>
<b>unsol-detect-timeout</b>	Optional	<p>Indicates how long, in seconds, the request is blocked when the value for unsol-detect-sync is Y and the detection result is not complete. The default value, 20 seconds, is used when this field is not specified and the value for unsol-detect-sync is Y.</p>
<b>solKeyReg</b>	Optional	<p>If the sol-key property is specified, this property indicates whether sol-key represents a regular expression.</p> <p><b>Y</b> sol-key is a regular expression.</p> <p><b>N</b> sol-key is a normal string.</p> <p>For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key:</p> <pre>a[\s]+regular</pre> <p>If you omit this property, N is used by default.</p>

Table 320. Request content for the issue command request (continued)

Field name	Required or Optional	Description
<b>unsolKeyReg</b>	Optional	<p>If the unsol-key property is specified, this property indicates whether unsol-key represents a regular expression.</p> <p><b>Y</b> unsol-key is a regular expression.</p> <p><b>N</b> unsol-key is a normal string.</p> <p>For example, suppose that you want to use a regular expression to find the phrase "a regular" in the message "This is a regular expression". If you are not sure how many spaces exist between "a" and "regular" in the message, you can use following key:</p> <pre>a[\s]+regular</pre> <p>If you omit this property, N is used by default.</p>
<b>auth</b>	Optional	<p>Command authority for the console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p><b>MASTER</b> Allows this console to act as a master console, which can issue all MVS operator commands.</p> <p><b>ALL</b> Allows this console to issue system control commands, input/output commands, console control commands, and informational commands.</p> <p><b>INFO</b> Allows this console to issue informational commands.</p> <p><b>CONS</b> Allows this console to issue console control and informational commands.</p> <p><b>IO</b> Allows this console to issue input/output and informational commands.</p> <p><b>SYS</b> Allows this console to issue system control commands and informational commands.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>

Table 320. Request content for the issue command request (continued)

Field name	Required or Optional	Description
<b>routcde</b>	Optional	<p>Routing codes for the console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p><b>ALL</b> All routing codes.</p> <p><b>NONE</b> No routing codes.</p> <p><b>(routing-codes)</b> One or more routing codes or sequences of routing codes. The routing codes can be list of n and n1:n2, where n, n1, and n2 are integers 1 - 128, and n2 is greater than n1.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>
<b>mscope</b>	Optional	<p>The systems from which this console can receive messages that are not directed to a specific console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p><b>(system-name)</b> List of one or more system names, where system-name can be any combination of A - Z, 0 - 9, # (X'7B'), \$ (X'5B'), or @ (X'7C').</p> <p><b>LOCAL</b> System on which the console is active.</p> <p><b>ALL</b> All systems.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>
<b>storage</b>	Optional	<p>Amount of storage in kilobytes in the TSO/E user's address space, which can be used for message queuing to this console. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p> <p>If the expected console message size is large, it is recommended that you increase the STORAGE parameter to a larger value. For example, if the expected response size is 10 megabytes, consider setting the STORAGE parameter to 15M, based on the following formula: 10M+5M=15M.</p>

Table 320. Request content for the issue command request (continued)

Field name	Required or Optional	Description
<b>auto</b>	Optional	<p>Specifies whether the console receives messages that are eligible for automation. The first time the user issues a command from the console, the value of this field is returned to the user.</p> <p>The values are:</p> <p><b>YES</b> The console receives messages that are eligible for automation.</p> <p><b>NO</b> The console does not receive messages that are eligible for automation.</p> <p>If you omit this property, the value that is defined in your external security manager, such as RACF, is used by default.</p>

## Authorization requirements

See [“Required authorizations”](#) on page 527.

## HTTP status codes

For a successful request, HTTP status code 200 is returned and the response body is provided, as described in [“Request content”](#) on page 529.

For unsuccessful requests, the service returns the status codes that are described in [Table 321](#) on page 533.

Table 321. HTTP error response codes for an issue command request

HTTP Status	Return Code	Reason Code	Reason	Description
<b>400</b>	1	3	No match for method PUT and pathInfo=' %s '.	<p>The path information, %s, in the original request contains a URL that is not acceptable for the z/OS Console API. Ensure that the request contains the correct URL.</p> <p>A console name must be 2 - 8 alphanumeric characters, the first of which must be alphabetic or one of the special characters #, \$ or @.</p>
<b>400</b>	1	5	Console API cannot recognize the JSON field: %s	The JSON field, %s, in the request body, is not a supported field.
<b>400</b>	1	6	The Content-Type ' %s ' cannot be handled, 'application/json' is expected.	The Content-Type, %s, in the original request contains an incorrect value for the HTTP Content-Type header. The z/OS Console API accepts only application/json for the Content-Type. Update the value of the HTTP Content-Type header and make sure that the request body is in JSON format.

Table 321. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	11	Format of parameter 'rsize' is wrong, it cannot be changed to a number.	The rsize parameter requires a numeric value, but the supplied value is not a number. Change the value to a number.
400	1	12	The body of the request is not in JSON format.	The request body must be in JSON format, but the supplied request body is not in JSON format. Correct the request body to be in JSON format.
400	1	13	Cannot find 'cmd' in request body, or value of 'cmd' is empty. No command is issued.	No cmd field was found in the request body, or the cmd field is empty. The cmd field specifies the command to be issued. No command is issued. Ensure that the request body includes a cmd field with a value.
400	1	14	Incorrect console name. The length of console name must be greater than 1 and less than 9.	The console name that is specified in the URL is not valid. Supply a valid console name.
400	1	17	Command length must be less than 127.	The value of the cmd field exceeds the maximum length of a command, which is 126 characters. Provide a valid command.
400	1	21	The TSO/E address space cannot be created because an error occurred with the logon procedure or the user settings.	<p>Refer to message IZUG1121E for a detailed explanation.</p> <p>To resolve the issue, try one or more of the following actions:</p> <ul style="list-style-type: none"> <li>• Verify that the logon procedure exists and is valid.</li> <li>• Specify a different region size, or use the installation-defined default.</li> <li>• If profile sharing is turned on, turn it off. Ensure that you are not simultaneously running a 3270 z/OS ISPF session.</li> <li>• If you want to use profile sharing, do the following: <ul style="list-style-type: none"> <li>– Ensure that each data set that is defined on the ISPPROF DD statement in the logon procedure is allocated with DISP=SHR.</li> <li>– Turn on profile sharing in the user settings for the z/OSMF ISPF task.</li> <li>– For a 3270 z/OS ISPF session, start the z/OS ISPF application with the SHRPROF option.</li> </ul> </li> <li>• Use the same logon procedure to start a 3270 z/OS ISPF session, and correct any errors that are identified.</li> </ul>

Table 321. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	2	7	Internal https connection timeout.	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	8	I/O error when connecting TSO service	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	2	21	Timeout when creating TSO address space for console %s	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	22	Timeout when activating console %s	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	3	1	REST TSO service returned non-200 status code when creating TSO address space.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a TSO address space. Contact your z/OSMF administrator.
500	3	2	REST TSO service returned an error message when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned a success (200) HTTP response with an unexpected message. Contact your z/OSMF administrator.
500	3	3	REST TSO service returned non-200 status code when setting up solicited and unsolicited message display.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	4	Cannot retrieve TSO AS key from data that is returned by REST TSO service.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	6	Unknown error occurred when creating or getting the TSO AS.	An unknown error occurred during an attempt to create a TSO address space. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	7	REST TSO service returned a non-200 status code.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when issuing a command. Contact the z/OSMF administrator.
500	3	8	Server end program cannot be found.	The server end program of the REST Console API cannot be found. Contact the z/OSMF administrator.
500	3	9	JSON serialization failed when calling a REXX program.	An internal error occurred during the process of converting the response from a TSO service. Contact the z/OSMF administrator.

Table 321. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	10	Unexpected messages were found when calling a REST TSO service.	TSO error messages were found when calling the REST TSO service to issue a command. Contact the z/OSMF administrator.
500	3	11	The maximum number of TSO/E address spaces that are allowed for the current user has been reached.	Refer to message IZUG1127E for a detailed explanation.  Display the active TSO/E address spaces, and remove or cancel any address spaces that the user no longer needs. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console.  To cancel a TSO/E address space, issue the <code>C u=user-ID,a=ASID</code> command from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.
500	3	14	An exception occurred when connecting to the TSO/E address space. Error description: %s.	The request failed because an error occurred. The context of the error is provided in the message text: <code>error description, %s</code> . For details about the error, check the z/OSMF logs. Correct any errors that you find. If the problem persists, contact IBM Support and provide the error details.
500	3	18	The maximum number of TSO/E address spaces has been reached.	Refer to IZUG1105E for a detailed explanation.  Display the active TSO/E address spaces, and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console.  To cancel a TSO/E address space, enter the command <code>C u=user-ID,a=ASID</code> from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.

Table 321. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	19	The TSO/E user account number for the console is not defined for use.	Refer to message IKJ56486I for a detailed explanation. One of the following occurred: <ul style="list-style-type: none"> <li>The specified account number is not defined to the RACF database.</li> </ul> The RACF administrator must first define the account number as a RACF resource and then give the user access using the PERMIT command. However, if the procedure is not in the procedure library, the logon attempt will continue to fail. <ul style="list-style-type: none"> <li>The RACF class ACCTNUM is not active.</li> </ul> The RACF administrator must activate the RACF class ACCTNUM using the SETROPTS command.
500	3	20	TSO/E user account number for the console has not been authorized for the user.	Refer to message IKJ56487I for a detailed explanation. The specified account number is defined to the RACF database. However, this particular user ID is not allowed to use it.
500	3	21	TSO/E user account number for the console is not valid.	Refer to message IKJ56702I for a detailed explanation. The specified account number is incorrect.
500	3	30	An error occurred in the TSO/E address space. Error description: %s	The request failed because an error occurred. The context of the error is provided in the message text: <i>error description, %s</i> . To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	5	1	REST TSO service returned a non-200 status code when creating a console.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a console. Contact the z/OSMF administrator.
500	5	2	Incorrect parameters were passed in when creating a console object.	An internal error occurred during an attempt to create a console. Contact the z/OSMF administrator.
500	5	3	No application data returned when initialize time zone for console service.	An internal error occurred during an attempt to prepare a console. Check the JES spool or other system resources for a resource shortage in the system. Retry the request. If the problem persists, contact the z/OSMF administrator.

Table 321. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	4	Unexpected IEE136I message: %s	The returned IEE136I message, %s, is not in the correct format. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	5	Failed to retrieve the time zone from message IEE136I. Local Time: %s, UTC Time: %s	z/OSMF console service failed to retrieve necessary information from the returned IEE136I message. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	6	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	7	Create console failed due to a TSO console command error.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	8	The number of consoles has reached the limit.	The maximum number of consoles that are supported by the z/OS Console API was reached. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	10	The requested EMCS console exists in another TSO/E address space.	The console is already created by another user. Delete the other console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	12	User is not authorized to MVS.MCSOPER.console_name.	The user requires at least READ access for resource profile MVS.MCSOPER.console_name.
500	5	14	CONSPROF is not defined as a TSO/E authorized command.	Refer to message IKJ55354I for a detailed explanation.  Ensure that the CONSPROF command resides in an authorized library and that the CONSPROF command name is placed in the authorized command name table. For more information, see <a href="#">Customizing the CONSOLE and CONSPROF commands in z/OS TSO/E Customization</a> .

Table 321. HTTP error response codes for an issue command request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	17	The value for auth is not valid. The following values are valid: MASTER, ALL, INFO, CONS, IO, SYS.	The command authority for the console is not valid. Valid values are: <ul style="list-style-type: none"> <li>• MASTER</li> <li>• ALL</li> <li>• INFO</li> <li>• CONS</li> <li>• IO</li> <li>• SYS</li> </ul>
500	5	18	The value for routcode is not valid.	The routing codes for the console are not valid. Valid values are: <ul style="list-style-type: none"> <li>• ALL</li> <li>• NONE</li> <li>• List of one or more routing codes in the range 1-128, enclosed in parentheses. Multiple entries are separated by commas.</li> </ul>
500	5	19	The value for mscope is not valid.	The message scope for the console is not valid. Valid values are: <ul style="list-style-type: none"> <li>• ALL</li> <li>• LOCAL</li> <li>• List of one or more system names, which are enclosed in parentheses. Multiple entries are separated by commas.</li> </ul>
500	5	20	The value for storage is not valid.	The value for storage must be a number in the range 1 - 2000.
500	5	21	The value for auto is not valid.	The message automation status for the console is not valid. Valid values are: <ul style="list-style-type: none"> <li>• YES</li> <li>• NO</li> </ul>
500	5	22	The EMCS console cannot be activated because the user has insufficient access to resource profile CONOPER.	User is not authorized to CONOPER. The user requires at least READ access for resource profile CONOPER CLASS(TSOAUTH).
500	5	23	OPERPARM value is not valid.	Check the console's OPERPARM values for syntax errors. Correct the errors and try the request again.

<i>Table 321. HTTP error response codes for an issue command request (continued)</i>				
<b>HTTP Status</b>	<b>Return Code</b>	<b>Reason Code</b>	<b>Reason</b>	<b>Description</b>
<b>500</b>	8	13	Recovery of persistence data is not complete, try later.	The z/OS Console API recovery process was not complete when you issued the request. Wait a few seconds, then try the request again.
<b>500</b>	8	14	Cannot get the command response.	The z/OS Console API failed to get the command response. Try the request again. If the problem persists, contact the z/OSMF administrator.

## Response content

On successful completion, the service returns a response body, which contains a JSON object. The JSON object varies depending on whether the request was synchronous or asynchronous. For a description of fields in the JSON object, see either of the following tables:

- [Table 322 on page 540](#)
- [Table 323 on page 541.](#)

Table 322 on [page 540](#) describes the response content for a successful synchronous issue command response.

<i>Table 322. Response content for a successful synchronous issue command request</i>	
<b>Field name</b>	<b>Description</b>
<b>cmd-response</b>	Command response.
<b>cmd-response-url</b>	URL that can be used to retrieve the command response later when the value for cmd-response is empty.
<b>cmd-response-uri</b>	URI that can be used to retrieve the command response later when the value for cmd-response is empty. The URI starts with /zosmf.
<b>cmd-response-key</b>	Key that can be used to retrieve the command response later when the value for cmd-response is empty.
<b>consoleAuth</b>	Command authority for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
<b>consoleRoutcde</b>	Routing codes for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
<b>consoleMscope</b>	The systems from which this console can receive messages that are not directed to a specific console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.

Table 322. Response content for a successful synchronous issue command request (continued)

Field name	Description
<b>consoleStorage</b>	<p>Amount of storage in kilobytes in the TSO/E user's address space, which can be used for queuing messages to this console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.</p> <p><b>Note:</b> If the operand OPERPARM(STORAGE) is not specified, the console uses STORAGE(1) when the session is established.</p> <p>If the expected console message size is very large, it is recommended that you increase the STORAGE parameter to a larger value. For example, if the expected response size is 10 megabytes, consider setting the STORAGE parameter to 15M, based on the following formula: 10M+5M=15M.</p>
<b>consoleAuto</b>	<p>Specifies whether the console receives messages that are eligible for automation. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.</p>
<b>sol-key-detected</b>	<p>Returned when sol-key was specified, and unsol-detect-sync was specified as N or not specified. If the keyword was detected in the command response, the value is true. Otherwise, the value is false.</p>
<b>ipcmmsgqbytes</b>	<p>Maximum number of bytes in a single message queue. The returned value is a decimal value in the range 0 - 2147483647. The default value is 2147483647 (or 2G).</p> <p>This field is included only for the first command to be issued with this console. On subsequent uses, this field is omitted.</p> <p><b>Note:</b> If this value is less than the maximum (2147483647), the z/OSMF Operator Consoles task might lose some solicited or unsolicited messages. If so, your system programmer can use the SETOMVS or SET OMVS command to increase the value of IPCMSGQBYTES.</p>

Table 323 on page 541 describes the response content for a successful asynchronous issue command response.

Table 323. Response content for a successful asynchronous issue command request

Field name	Description
<b>cmd-response-url</b>	URL that can be used to retrieve the command response.
<b>cmd-response-uri</b>	URI that can be used to retrieve the command response. The URI starts with / zosmf.
<b>cmd-response-key</b>	Key that can be used to retrieve the command response.
<b>consoleAuth</b>	Command authority for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
<b>consoleRoutcde</b>	Routing codes for the console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
<b>consoleMscope</b>	The systems from which this console can receive messages that are not directed to a specific console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.

Table 323. Response content for a successful asynchronous issue command request (continued)

Field name	Description
<b>consoleStorage</b>	<p>Amount of storage in kilobytes in the TSO/E user's address space, which can be used for queuing messages to this console. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.</p> <p><b>Note:</b> If the operand OPERPARM (STORAGE) is not specified, the console uses STORAGE (1) when the session is established.</p> <p>If the expected console message size is very large, it is recommended that you increase the STORAGE parameter to a larger value. For example, if the expected response size is 10 megabytes, consider setting the STORAGE parameter to 15M, based on the following formula: 10M+5M=15M.</p>
<b>consoleAuto</b>	Specifies whether the console receives messages that are eligible for automation. The first time the user issues a command from the console, the value of this field is returned to the user. Otherwise, this field is omitted.
<b>detection-url</b>	The URL that can be used later to retrieve the detection result for detecting a keyword from unsolicited messages. Returned when unsol-key was specified to detect a keyword in unsolicited messages, and unsol-detect-sync was specified as N or not specified.
<b>detection-uri</b>	The URI that can be used later to retrieve the detection result for detecting a keyword from unsolicited messages. Returned when unsol-key was specified to detect a keyword in unsolicited messages, and unsol-detect-sync was specified as N or not specified.
<b>detection-key</b>	Returned when unsol-key was specified to detect a keyword in unsolicited messages. You can use this value to retrieve the result.
<b>status</b>	<p>Status of the unsolicited detection request. Returned when sol-key was specified, and unsol-detect-sync is specified as Y. The values are:</p> <p><b>expired</b> The detection request is expired. No matching record in the unsolicited messages was found in the time that is specified by detect-time.</p> <p><b>detected</b> Matching records in the unsolicited messages were found in the time that is specified by detect-time. msg contains the message that contains the keyword.</p> <p><b>timeout</b> The unsol-detect-timeout elapsed before the detection result completed.</p> <p><b>detection-url</b> The URL that can be used to retrieve the detection result for detecting a keyword from unsolicited messages.</p> <p><b>detection-uri</b> The URI that can be used to retrieve the detection result for detecting a keyword from unsolicited messages.</p> <p><b>detection-key</b> The key that can be used to retrieve the unsolicited keyword detection result.</p>

Table 323. Response content for a successful asynchronous issue command request (continued)

Field name	Description
<b>ipcmsgqbytes</b>	<p>Maximum number of bytes in a single message queue. The returned value is a decimal value in the range 0 - 2147483647. The default value is 2147483647 (or 2G).</p> <p>This field is included only for the first command to be issued with this console. On subsequent uses, this field is omitted.</p> <p><b>Note:</b> If this value is less than the maximum (2147483647), the z/OSMF Operator Consoles task might lose some solicited or unsolicited messages. If so, your system programmer can use the SETOMVS or SET OMVS command to increase the value of IPCMSGQBYTES.</p>

The client application can use any one of detection-url, detection-uri, or detection-key to retrieve the detection result.

If a failure occurs, the response body contains a JSON object that describes the error.

Table 324. Response content for an unsuccessful issue command request

Field name	Description
<b>return-code</b>	Category of the error.
<b>reason-code</b>	Specific error.
<b>reason</b>	Text that describes the cause of the error.

## Example HTTP interactions

1. The example in [Figure 282 on page 543](#) shows a request to issue the system command **d a, pegasus** synchronously.

```
PUT https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/ibmusecn
{"cmd":"d a,PEGASUS", "routcode" : "ALL"}
```

Figure 282. Sample request to issue a command synchronously

The following is the response body for the request. In the response, \r is the return character.

```
{
  "cmd-response-key":"C6557643",
  "cmd-response-uri":"https://pev076.pok.ibm.com:443/zosmf/restconsoles/consoles/ibmusecn/solmsgs/C6557643",
  "IPCMMSGQBYTES":"2147483647",
  "consoleRoutcode":"ALL",
  "consoleMscope":"ALL",
  "consoleAuto":"NO",
  "cmd-response-uri":"/zosmf/restconsoles/consoles/ibmusecn/solmsgs/C6557643",
  "cmd-response":"CNZ4106I 04.22.11 DISPLAY ACTIVITY 532\r
JOBS   M\ S   TS USERS  SYSAS  INITS  ACTIVE\MAX VTAM   OAS\r 00002   00015   00002   00032   00005   00001\
00020 00011\r
PEGASUS PEGASUS *OMVSEX IN  SO  A=0038  PER=NO  SMC=000\r
PGN=N\A  DMN=N\A  AFF=NONE\r
CT=006.589S  ET=05.49.06\r
WUID=STC00061 USERID=ZOSMFAD\r
WKL=SYSTEM  SCL=SYSSTC  P=1\r
RGP=N\A     SRVR=NO  QSC=NO\r
ADDR SPACE ASTE=01ED0E00",
  "consoleStorage":"1024",
  "consoleAuth":"INFO"}
```

Figure 283. Sample response body

2. The example in [Figure 284 on page 544](#) shows a request to issue the system command `d a,PEGASUS` synchronously, and attempt to detect PEGASUS in the command response.

```
PUT https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/ibmusecn
{"cmd":"d a,PEGASUS","sol-key":"PEGASUS"}
```

*Figure 284. Sample request to issue a command and detect a keyword*

The following is the response body for the request.

```
{"cmd-response": "CNZ4106I 07.30.59 2016.011 DISPLAY ACTIVITY 070\r JOBS M/S TS USERS SYSAS INITS ACTIVE/MAX
VTAM OAS\r 00003 00013 00002 00032 00011 00001\V00020 00015\r PEGASUS NOT FOUND",
"sol-key-detected":true,"cmd-response-uri":"\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"cmd-response-url":"https://pev076.pok.ibm.com:443\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"cmd-response-key":"C005291"}
```

*Figure 285. Sample response body*

3. The example in [Figure 286 on page 544](#) shows a request to issue the system command `s PEGASUS` asynchronously and attempt to detect PEGASUS in the unsolicited messages.

```
PUT https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/defcn
{"cmd":"s PEGASUS","unsol-key":"PEGASUS","async":"Y"}
```

*Figure 286. Sample request to issue a system command asynchronously*

The following is the response body for the request.

```
{"cmd-response-uri":"\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"detection-url":"https://pev076.pok.ibm.com:443\zosmf\restconsoles\consoles\ibmusecn\detections\dec6800",
"detection-uri":"\zosmf\restconsoles\consoles\ibmusecn\detections\dec6800",
"detection-key":"dec6800",
"cmd-response-url":"https://pev076.pok.ibm.com:443\zosmf\restconsoles\consoles\ibmusecn\solmsgs\C005291",
"cmd-response-key":"C005291"}
```

*Figure 287. Sample response body*

4. The example in [Figure 288 on page 544](#) shows a request to issue an `s PEGASUS` command synchronously, by using the default console, and detect keyword PEGASUS in the unsolicited messages synchronously. The keyword is found in unsolicited messages before the timeout is reached.

```
PUT https://PEV076.pok.ibm.com/zosmf/restconsoles/consoles/defcn
{"cmd":"s PEGASUS","unsol-key":"PEGASUS","unsol-detect-sync":"Y"}
```

*Figure 288. Sample request to issue an s PEGASUS command synchronously and detect the keyword PEGASUS*

The following is the response body for the request.

```
{"status":"detected","cmd-response":"BPXM023I (ZOSMFAD) CFZ02300I: Configuration property
httpAuthType is not supported. Setting ignored.","msg":"$HASP100 PEGASUS ON STCINRDR"}
```

*Figure 289. Sample response body*

5. The example in [Figure 288 on page 544](#) shows a request to issue an `s PEGASUS` command asynchronously, by using the default console, and detect keyword XIAOX in the unsolicited messages

synchronously. The detection result is not complete before the timeout (the default of 20 seconds) was reached.

```
PUT https://PEV076.pok.ibm.com/zosmf/restconsoles/consoles/defcn
{"cmd":{"s PEGASUS","async":"Y","unsol-key":"XIAOX","unsol-detect-sync":"Y"}}
```

Figure 290. Sample request to issue an s PEGASUS command asynchronously and detect the keyword XIAOX

The following is the response body for the request.

```
{"cmd-response-uri":"/zosmf/restconsoles/consoles/defcn/solmsgs/C2790426","detection
-url":"https://PEV076.pok.ibm.com:443
/zosmf/restconsoles/consoles/defcn/detections/D5303564","detection-uri":"
/zosmf/restconsoles/consoles/defcn/detections
/D5303564","detection-key":"D5303564","status":"timeout","cmd-response-url":"https:
//PEV076.pok.ibm.com:443/zosmf/restconsoles/consoles/defcn/solmsgs/C2790426",
"cmd-response-key":"C2790426"}
```

Figure 291. Sample response body

## Get a command response

Use this operation to get the response to a command that was issued asynchronously with the Issue Command service.

### HTTP method and URI path

```
GET /zosmf/restconsoles/consoles/console-name/solmsgs/Ckey-number
GET /zosmf/restconsoles/consoles/defcn/solmsgs/Ckey-number
```

where:

**consolename**

is the name of the EMCS console that was used in the Issue Command request.

**defcn**

indicates that name of the console that was used to issue the command was generated by the REST Console API.

**Ckey-number**

is the command response key from the Issue Command request.

The URL is returned by the Issue Command request in the cmd-response-url field.

### Query parameters

None.

### Description

This operation gets the messages that were issued in response to a command that was issued asynchronously with the Issue Command service. For the properties that you can specify, see [“Request content” on page 546](#).

On successful completion, HTTP status code 200 is returned. The response content is described in [“Response content” on page 550](#).

The Issue Command service returns the URL of the command response in the cmd-response-url field. For more information about the response content of the Issue Command service, see [“Response content” on page 540](#).

## Request content

None.

## Authorization requirements

See [“Required authorizations” on page 527](#).

## HTTP status codes

On successful completion, HTTP status code 200 is returned and the response body is provided, as described in [“Response content” on page 550](#).

Otherwise, the HTTP status codes in [Table 325 on page 546](#) are returned for the indicated errors.

<i>Table 325. HTTP error response codes for a get command response request</i>				
HTTP Status	Return Code	Reason Code	Reason	Description
400	1	3	No match for method GET and pathInfo=' %s '.	The path information, %s, in the original request contains a URL that is not acceptable for the z/OS Console API. Ensure that the request contains the correct URL.  A console name must be 2 - 8 alphanumeric characters, the first of which must be alphabetic or one of the special characters #, \$ or @.
400	1	5	Console API cannot recognize the JSON field: %s	The JSON field, %s, in the request body, is not a supported field.
400	1	14	Invalid console name. The length of console name must be greater than 1 and less than 9.	The console name that is specified in the URL is not valid. Supply a valid console name.

Table 325. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	21	The TSO/E address space cannot be created because an error occurred with the logon procedure or the user settings.	<p>Refer to message IZUG1121E for a detailed explanation.</p> <p>To resolve the issue, try one or more of the following actions:</p> <ul style="list-style-type: none"> <li>• Verify that the logon procedure exists and is valid.</li> <li>• Specify a different region size, or use the installation-defined default.</li> <li>• If profile sharing is turned on, turn it off. Ensure that you are not simultaneously running a 3270 z/OS ISPF session.</li> <li>• If you want to use profile sharing, do the following: <ul style="list-style-type: none"> <li>– Ensure that each data set that is defined on the ISPPROF DD statement in the logon procedure is allocated with DISP=SHR.</li> <li>– Turn on profile sharing in the user settings for the z/OSMF ISPF task.</li> <li>– For a 3270 z/OS ISPF session, start the z/OS ISPF application with the SHRPROF option.</li> </ul> </li> <li>• Use the same logon procedure to start a 3270 z/OS ISPF session, and correct any errors that are identified.</li> </ul>
500	2	21	Timeout when creating TSO address space for console %s	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	22	Timeout when activating console %s	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	3	1	REST TSO service returned a non-200 status code when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a TSO address space. Contact your z/OSMF administrator.
500	3	2	REST TSO service returned an error message when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned a success (200) HTTP response with an unexpected message. Contact your z/OSMF administrator.

Table 325. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	3	REST TSO service returned non-200 status code when setting up solicited and unsolicited message display.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	4	Cannot retrieve TSO AS key from data returned by REST TSO service.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	7	REST TSO service returned a non-200 status code.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when issuing a command. Contact the z/OSMF administrator.
500	3	8	Server end program cannot be found.	The server end program of the REST Console API cannot be found. Contact the z/OSMF administrator.
500	3	9	JSON serialization failed when calling a REXX program.	An internal error occurred during the process of converting the response from a TSO service. Contact the z/OSMF administrator.
500	3	10	Unexpected messages were found when calling a REST TSO service.	TSO error messages were found when calling the REST TSO service to issue a command. Contact the z/OSMF administrator.
500	3	11	The maximum number of TSO/E address spaces allowed for the current user has been reached.	Refer to message IZUG1127E for a detailed explanation.  Display the active TSO/E address spaces, and remove or cancel any address spaces that the user no longer needs. To display the active TSO/E address spaces, enter the command D TS,ALL from the operator console.  To cancel a TSO/E address space, issue the C u=user-ID,a=ASID command from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.

Table 325. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	18	The maximum number of TSO/E address spaces has been reached.	Refer to IZUG1105E for a detailed explanation.  Display the active TSO/E address spaces, and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command <code>D TS, ALL</code> from the operator console.  To cancel a TSO/E address space, enter the command <code>C u=user-ID, a=ASID</code> from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.
500	3	30	An error occurred in the TSO/E address space. Error description: %s	The request failed because an error occurred. The context of the error is provided in the message text: <i>error description, %s</i> . To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	5	1	REST TSO service returned a non-200 status code when creating a console.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a console. Contact the z/OSMF administrator.
500	5	2	Invalid parameters were passed in when creating a console object.	An internal error occurred during an attempt to create a console. Contact the z/OSMF administrator.
500	5	3	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Check the JES spool or other system resources for a resource shortage in the system. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	4	Cannot retrieve local time zone.	The returned IEE136I message, %s, is not in the correct format. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	5	Cannot retrieve local time zone.	z/OSMF console service failed to retrieve necessary information from the returned IEE136I message. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	6	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.

Table 325. HTTP error response codes for a get command response request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	7	Create a console failed due to a TSO console command error.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	8	The numbers of consoles has reached the limit.	The maximum number of consoles that are supported by the z/OS Console API was reached. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	10	The requested EMCS console already exists in another TSO/E address space.	The console is already created by another user. Delete the other console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	12	User is not authorized to MVS.MCSOPER.console_name.	The user requires at least READ access for resource profile MVS.MCSOPER.console_name.
500	5	14	CONSPROF is not defined as a TSO/E authorized command.	Refer to message IKJ55354I for a detailed explanation.  Ensure that the CONSPROF command resides in an authorized library and that the CONSPROF command name is placed in the authorized command name table. For more information, see <a href="#">Customizing the CONSOLE and CONSPROF commands in z/OS TSO/E Customization</a> .
500	8	13	Recovery of persistence data is not complete, try later.	The z/OS Console API recovery process was not complete when you issued the request. Wait a few seconds, then try the request again.
500	10	1	The message you requested cannot be retrieved due to earlier shutdown of z/OSMF server.	The z/OS Console API failed to get the command response. Try the request again. If the problem persists, contact the z/OSMF administrator.

## Response content

On successful completion, the service returns a response body, which contains a JSON object. [Table 326](#) on page 550 lists the fields in the JSON object.

Table 326. Response content for a successful get command response request

Field name	Description
<b>cmd-response</b>	Command response
<b>sol-key-detected</b>	Returned when sol-key was specified on the Issue Command service. If the keyword specified with sol-key was found in the command response, the value is true. Otherwise, the value is false.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 327. Response content for an unsuccessful get command response request

Field name	Description
return-code	Category of the error.
reason-code	Specific error.
reason	Text that describes the cause of the error.

## Example HTTP interaction

The example in [Figure 292 on page 551](#) shows a request to get the response to a system command that was issued asynchronously. The command was issued with a generated console name. The command response key returned by the issue command request is C003715.

```
GET https://pev061.pok.ibm.com:443/zosmf/restconsoles/consoles/ibmusecn/solmsgs/C508135
```

Figure 292. Sample request to get the response for a system command that was issued asynchronously

The following is the response body for the request.

```
{ "cmd-response": "IEE215I 07.36.34 2016.011 PARMLIB DISPLAY 513\R PARMLIB DATA SETS SPECIFIED\R AT IPL\R ENTRY FLAGS  
VOLUME DATA SET\R 1 S PEVTS3 CIMSSRE.R22ONLY.PARMLIB\ 2 S PEVTS3  
CIMSSRE.R14ONLY.PARMLIB\ 3 S PEVTS3 CIMSSRE.R13ONLY.PARMLIB\ 4 S PEVTS3  
CIMSSRE.R12ONLY.PARMLIB\ 5 S PEVTS3 CIMSSRE.PARMLIB\ 6 S PEVTS3 HDENNIS.ZOS17.PARMLIB\ 7  
S CTTAK XESCT.PARMLIB\ 8 S CTTAK SYS1.PARMLIB\ 9 S SDR22 SYS1.PARMLIB.POK\ 10  
S SDR22 SYS1.PARMLIB.INSTALL" }
```

Figure 293. Sample response body for a get command response request

## Get the detect result for unsolicited messages

Use this operation to get the result for detecting a keyword in unsolicited messages after an Issue Command request. The command must have been issued with the unsol-key field.

### HTTP method and URI path

```
GET /zosmf/restconsoles/consoles/consolename/detections/Dkey-number  
GET /zosmf/restconsoles/consoles/defcn/detections/Dkey-number
```

where:

#### ***consolename***

is the name of the EMCS console that was used in the Issue Command request.

#### ***defcn***

indicates that name of the console that was used to issue the command was generated by the REST Console API.

#### ***Dkey-number***

is the detection key from the Issue Command request.

The URL is be returned by the Issue Command request in the detection-url field.

### Query parameters

None.

## Description

This operation gets the results of attempting to detect a keyword in the unsolicited messages that were issued following an Issue Command request. The keyword being detected was specified with the unsol-key field on the Issue Command service.

On successful completion, HTTP status code 200 is returned. The response content is described in [“Response content” on page 556](#).

The Issue Command service returns the URL in the detection-url field. For more information about the response content of the Issue Command service, see [“Response content” on page 540](#).

## Request content

None.

## Authorization requirements

See [“Required authorizations” on page 527](#).

## HTTP status codes

On successful completion, HTTP status code 200 is returned and the response body is provided, as described in [“Response content” on page 556](#).

Otherwise, the HTTP status codes in [Table 328 on page 552](#) are returned for the indicated errors.

Table 328. HTTP error response codes for a detect result for unsolicited messages request				
HTTP Status	Return Code	Reason Code	Reason	Description
400	1	3	No match for method GET and pathInfo=' %s '.	The path information, %s, in the original request contains a URL that is not acceptable for the z/OS Console API. Ensure that the request contains the correct URL.  A console name must be 2 - 8 alphanumeric characters, the first of which must be alphabetic or one of the special characters #, \$ or @.
400	1	5	Console API cannot recognize the JSON field: %s	The JSON field, %s, in the request body, is not a supported field.
400	1	14	Invalid console name. The length of console name must be greater than 1 and less than 9.	The console name that is specified in the URL is not valid. Supply a valid console name.

Table 328. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
400	1	21	The TSO/E address space cannot be created because an error occurred with the logon procedure or the user settings.	<p>Refer to message IZUG1121E for a detailed explanation.</p> <p>To resolve the issue, try one or more of the following actions:</p> <ul style="list-style-type: none"> <li>• Verify that the logon procedure exists and is valid.</li> <li>• Specify a different region size, or use the installation-defined default.</li> <li>• If profile sharing is turned on, turn it off. Ensure that you are not simultaneously running a 3270 z/OS ISPF session.</li> <li>• If you want to use profile sharing, do the following: <ul style="list-style-type: none"> <li>– Ensure that each data set that is defined on the ISPPROF DD statement in the logon procedure is allocated with DISP=SHR.</li> <li>– Turn on profile sharing in the user settings for the z/OSMF ISPF task.</li> <li>– For a 3270 z/OS ISPF session, start the z/OS ISPF application with the SHRPROF option.</li> </ul> </li> <li>• Use the same logon procedure to start a 3270 z/OS ISPF session, and correct any errors that are identified.</li> </ul>
500	2	21	Timeout when creating TSO address space for console %s	The internal connection to the z/OSMF REST TSO service timed out. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	2	22	Timeout when activating console %s	An error occurred in the internal connection to the z/OSMF REST TSO service. Retry the command. If the problem persists, contact your z/OSMF administrator.
500	3	1	REST TSO service returned a non-200 status code when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a TSO address space. Contact your z/OSMF administrator.
500	3	2	REST TSO service returned an error message when creating a TSO address space.	The internal connection to the z/OSMF REST TSO service returned a success (200) HTTP response with an unexpected message. Contact your z/OSMF administrator.

Table 328. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	3	REST TSO service returned a non-200 status code when setting up solicited and unsolicited message display.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	4	Cannot retrieve TSO AS key from data returned by REST TSO service.	The attempt to prepare a TSO address space failed. Retry the request. If the problem persists, contact your z/OSMF administrator.
500	3	7	REST TSO service returned a non-200 status code.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when issuing a command. Contact the z/OSMF administrator.
500	3	8	Server end program cannot be found.	The server end program of the REST Console API cannot be found. Contact the z/OSMF administrator.
500	3	9	JSON serialization failed when calling a REXX program.	An internal error occurred during the process of converting the response from a TSO service. Contact the z/OSMF administrator.
500	3	10	Unexpected messages were found when calling a REST TSO service.	TSO error messages were found when calling the REST TSO service to issue a command. Contact the z/OSMF administrator.
500	3	11	The maximum number of TSO/E address spaces allowed for the current user has been reached.	<p>Refer to message IZUG1127E for a detailed explanation.</p> <p>Display the active TSO/E address spaces, and remove or cancel any address spaces that the user no longer needs. To display the active TSO/E address spaces, enter the command <code>D TS,ALL</code> from the operator console.</p> <p>To cancel a TSO/E address space, issue the <code>C u=user-ID,a=ASID</code> command from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.</p>

Table 328. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	3	18	The maximum number of TSO/E address spaces has been reached.	Refer to IZUG1105E for a detailed explanation.  Display the active TSO/E address spaces, and remove or cancel any address spaces that are no longer needed. To display the active TSO/E address spaces, enter the command <code>D TS, ALL</code> from the operator console.  To cancel a TSO/E address space, enter the command <code>C u=user-ID, a=ASID</code> from the operator console, where <i>user-ID</i> is the user's TSO/E ID and <i>ASID</i> is the address space identifier.
500	3	30	An error occurred in the TSO/E address space. Error description: %s	The request failed because an error occurred. The context of the error is provided in the message text: <i>error description, %s</i> . To obtain more details about the error, check the z/OSMF logs. Correct any errors. If the problem persists, contact the IBM Support Center and provide the error details.
500	5	1	REST TSO service returned a non-200 status code when creating a console.	The internal connection to the z/OSMF REST TSO service returned an error HTTP response when creating a console. Contact the z/OSMF administrator.
500	5	2	Invalid parameters were passed in when creating a console object.	An internal error occurred during an attempt to create a console. Contact the z/OSMF administrator.
500	5	3	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Check the JES spool or other system resources for a resource shortage in the system. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	4	Cannot retrieve local time zone.	The returned IEE136I message, %s, is not in the correct format. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	5	Cannot retrieve local time zone.	z/OSMF console service failed to retrieve necessary information from the returned IEE136I message. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	6	Cannot retrieve local time zone.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.

Table 328. HTTP error response codes for a detect result for unsolicited messages request (continued)

HTTP Status	Return Code	Reason Code	Reason	Description
500	5	7	Create console failed due to TSO console command error.	An internal error occurred during an attempt to prepare a console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	8	The number of consoles has reached the limit.	The maximum number of consoles that are supported by the z/OS Console API was reached. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	9	Cannot find the result for specified detection ID.	Cannot find the result for the specified detection ID. Ensure that the detection ID is correct.
500	5	10	The requested EMCS console already exists in another TSO/E address space.	The console is already created by another user. Delete the other console. Retry the request. If the problem persists, contact the z/OSMF administrator.
500	5	12	User is not authorized to MVS.MCSOPER.console_name.	The user requires at least READ access for resource profile MVS.MCSOPER.console_name.
500	5	14	CONSPROF is not defined as a TSO/E authorized command.	Refer to message IKJ55354I for a detailed explanation.  Ensure that the CONSPROF command resides in an authorized library and that the CONSPROF command name is placed in the authorized command name table. For more information, see <a href="#">Customizing the CONSOLE and CONSPROF commands in z/OS TSO/E Customization</a> .
500	8	13	Recovery of persistence data is not complete, try later.	The z/OS Console API recovery process was not complete when you issued the request. Wait a few seconds, then try the request again.
500	10	2	The detection result you requested cannot be retrieved due to earlier shutdown of the z/OSMF server.	The detection result cannot be retrieved because of an earlier shutdown of z/OSMF server.

## Response content

On successful completion, the service returns a response body, which contains a JSON object. [Table 329 on page 557](#) lists the fields in the JSON object.

Table 329. Response content for a successful get detect result request

Field name	Description
<b>status</b>	Status of the detection request:  <b>waiting</b> The detection request is still valid, the keyword has not yet been detected in the unsolicited messages.  <b>expired</b> The detection request expired, and the keyword was not found in the unsolicited messages. The detection request expires when the value for detect-time on the issue command request is exceeded.  <b>detected</b> The keyword was found in the unsolicited messages.
<b>msg</b>	Returned when the value of status is detected. This is the message that contains the keyword that was detected.

If a failure occurs, the response body contains a JSON object with a description of the error.

Table 330. Response content for an unsuccessful get detect result request

Field name	Description
<b>return-code</b>	Category of the error.
<b>reason-code</b>	Specific error.
<b>reason</b>	Text that describes the cause of the error.

## Example HTTP interaction

1. The example in [Figure 294 on page 557](#) shows a request to get the results for a detect request. The command was issued with a generated console name. The detection key returned by the issue command request is D002185.

```
GET https://pev076.pok.ibm.com/zosmf/restconsoles/consoles/defcn/detections/D002185
```

Figure 294. Sample request to get the detect result

The following is the response body for the request. The request is still valid, but the keyword has not been found.

```
{"status":"waiting","msg":""}
```

Figure 295. Sample response body for a get detect result request

2. The example in [Figure 294 on page 557](#) shows a request to get the results for a detect request. The command was issued with a generated console name. The detection key that was returned by the issue command request is D122033.

```
GET https://pev076.pok.ibm.com/zosmf/restconsole/consoles/defcn/detections/D122033
```

Figure 296. Sample request to get the detect result

The following is the response body for the request. The keyword was found. In the response, \r is the return character.

```
{ "status": "detected", "msg": "BPXM023I (ZOSMFAD)\r CFZ12584W: CIM Runtime Environment Userid currently only has READ\r access to BPX.SERVER. It is recommended to have either UPDATE access\r to BPX.SERVER or has to be UID 0." }
```

Figure 297. Sample response body for a successful get detect result request

## Get messages from logs

Use this operation to retrieve messages from logs on the system.

### HTTP method and URI path

```
GET /zosmf/restconsoles/log
```

#### Notes:

1. The maximum return size of a log is 9990. If more than 9990 logs exist in a specified timeframe, only the first 9990 logs are returned with the following condition: The logs in the same time frame are either returned or not returned at all.
2. If the returned log is more than 1 megabyte (1 MB), the response is compressed.

### Query parameters

Table 331. Query parameters for a Get Messages request

Parameter	Required or Optional	Description
time	Optional	Specifies when z/OSMF starts to retrieve messages in the ISO 8601 JSON time format. For example, 2021-01-26T03:33:18.065Z.  The default value is the current UNIX timestamp on the server.  This value is used if the timestamp parameter is not specified. Either time or timestamp must be specified.
timestamp	Optional	Specifies the UNIX timestamp, which is the number of milliseconds since 1970-01-01 UTC.  This value is used if the time parameter is not specified. Either time or timestamp must be specified.

Table 331. Query parameters for a Get Messages request (continued)

Parameter	Required or Optional	Description
timeRange	Optional	<p>Specifies the time range for which the log is to be retrieved. Supported time units include s, m, and h for seconds, minutes, and hours. For example: 10s, 10m, 10h.</p> <p>The default is 10m.</p> <p><b>Note:</b> The maximum return size of the log is 10000. If more than 10000 logs exist in the timeframe, the system returns the first 10000 logs with the following condition: The logs in the same 10ms timeframe will either be returned or not returned at all.</p>
direction	Optional	<p>Specifies the direction (from a specified time) in which messages are retrieved. The options are “forward” or “backward.” These strings are case-insensitive.</p> <p>The default is “backward,” meaning that messages are retrieved backward from the specified time.</p>

## Description

This operation retrieves the messages that were issued in response to a command that was issued asynchronously with the Issue Command service. For more information about the Issue Command service, see [“Response content” on page 540](#).

On successful completion, HTTP status code 200 is returned. The response content is described in [Table 333 on page 560](#).

## Request content

None.

## Authorization requirements

See [“Required authorizations” on page 527](#).

## HTTP status codes

On successful completion, HTTP status code 200 is returned and the response body is provided.

Otherwise, the HTTP status codes in [Table 332 on page 560](#) are returned for the indicated errors.

Table 332. HTTP Error response codes for a Get Messages request

HTTP Status	Return Code	Reason Code	Reason	Descriptions
400	1	22	The parameter s%: s% is invalid. Here are valid examples: s%.	One or more of the following parameters is not valid: <b>time</b> Must be in the ISO-8601 format. Example: 2021-05-25T07:00Z. <b>timeRange</b> The format is nnnU, nnn, in the range 1 - 999, where "U" can be hours "h," minutes "m," or seconds "s". For example: 20h, 543m, 5s. <b>direction</b> Either "backward" or "forward". <b>timestamp</b> Must be a valid UNIX timestamp.
400	1	23	The time or timestamp specified is a future time. Only a point at the past time is valid.	The specified time or timestamp is in the future. You must specify a time in the past.
500	12	1	ZlogsException occurred: macro, name=SYSPLEX.OPERLOG, returnCode=0xn, reasonCode=0xnnn	The macro is IXGCONN or IXGBRWSE. The values in returnCode and reasonCode explain the reason for the exception. For more information, see <a href="#">IXGCONN IXGCONN - Connect/disconnect to log stream in z/OS MVS Programming: Assembler Services Reference IAR-XCT</a> or <a href="#">IXGBRWSE IXGBRWSE- Browse/read a log stream in z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG</a> .

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the messages and logs. [“Response content” on page 560](#) lists the fields in the JSON object.

Table 333. Response content for a successful Get Messages request

Field name	Description
<b>timezone</b>	Specify the timezone of the z/OS system. Valid values for the timezone range from -12 to 12. For example, "-3" means UTC-3 timezone.
<b>totalItems</b>	Total number of messages returned in the response.
<b>nextTimestamp</b>	The UNIX timestamp. This value might be used in a subsequent request to specify a starting timestamp. Logs in the “nextTimestamp” are not returned in the current response.
<b>items</b>	JSON array of messages. For more information, see <a href="#">Table 334 on page 561</a> .

Table 333. Response content for a successful Get Messages request (continued)	
Field name	Description
source	Indicates the source of the messages. The valid value is "OPERLOG," which indicates the operations log.

Table 334. Messages JSON object	
Field name	Description
cart	Eight character command and response token (CART).
color	The color of the message.
jobName	The name of the job that generates the message.
message	The content of the message.
messageId	The message ID.
replyId	Reply ID, in decimal.
system	Original eight character system name.
type	HARDCOPY.
subtype	Indicate whether the message is a DOM, WTOR, or HOLD message.
time	For example, "Thu Feb 03 03:00 GMT 2021".
timestamp	UNIX timestamp. For example, 1621920830109.

If a failure occurs, the response body contains a JSON object with a description of the error. [Table 335 on page 561](#) shows the format of the JSON object.

Table 335. Response content for an unsuccessful Get Messages request	
Field name	Description
returnCode	Identifies the category of error.
reasonCode	Identifies the specific error.
reason	Text that describes the cause of the error.

## Example HTTP interaction

The example in [Figure 298 on page 561](#) shows a request to retrieve messages that occurred during a 1-hour duration.

```
GET https://pev076.pok.ibm.com/zosmf/restconsoles/v1/log?time=2021-05-25T07:00Z&timeRange=1h
```

*Figure 298. Sample request to get messages from a 1-hour duration*

The example in [Figure 299 on page 562](#) shows the response body for the preceding request.

```

{
  "nextTimestamp": 1621922666069,
  "source": "OPERLOGS",
  "totalItems": 2,
  "items": [
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163467248",
      "subType": "NULL",
      "time": "Tue May 25 06:04:26 GMT 2021",
      "message": " $HASP395 BPXAS  ENDED - RC=0000",
      "type": "HARDCOPY",
      "card": "0",
      "timestamp": 1621922666070
    },
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163467760",
      "subType": "NULL",
      "time": "Tue May 25 06:04:26 GMT 2021",
      "message": " $HASP395 BPXAS  ENDED - RC=0000",
      "type": "HARDCOPY",
      "card": "0",
      "timestamp": 1621922666070
    }
  ]
}

```

*Figure 299. Sample response body for a Get Messages request*

The example in [Figure 300 on page 562](#) shows a request to retrieve messages that start at a specific timestamp.

```
GET https://pev076.pok.ibm.com/zosmf/restconsoles/v1/log?timestamp=1621920870789&timeRange=15s&direction=forward
```

The example in [Figure 301 on page 562](#) shows the response body for the preceding request.

*Figure 300. Sample request for a Get Messages request that starts at a specific timestamp*

```

{
  "nextTimestamp": 1621920856259,
  "source": "OPERLOGS",
  "totalItems": 2,
  "items": [
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163454704",
      "subType": "NULL",
      "time": "Tue May 25 05:34:17 GMT 2021",
      "message": " $HASP373 BPXAS  STARTED",
      "type": "HARDCOPY",
      "card": "0",
      "timestamp": 1621920857500
    },
    {
      "jobName": "BPXAS  ",
      "system": "SY1    ",
      "color": "green",
      "replyId": "0",
      "messageId": "1163454960",
      "subType": "NULL",
      "time": "Tue May 25 05:34:18 GMT 2021",
      "message": " BPXP024I BPXAS INITIATOR STARTED ON BEHALF OF JOB IZUSVR13 RUNNING IN ASID 0028",
      "type": "HARDCOPY",
      "card": "0",
      "timestamp": 1621920858120
    }
  ]
}

```

*Figure 301. Sample response body for a Get Messages request*

## z/OS data set and file REST interface

The z/OS data set and file REST interface is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with data sets and UNIX files on a z/OS system.

The z/OS data set and file REST interface services provide a programming interface for working with z/OS data sets and UNIX files. This function is similar to using GET and PUT requests through file transfer protocol (FTP), but secured through traditional z/OS security controls for user authentication and resource authorizations. For setup details, see [“Required authorizations”](#) on page 567.

Table 336 on page 563 lists the operations that the z/OS data set and file REST interface services provide.

Table 336. Operations provided through the z/OS data set and file REST interface services.	
Operation	HTTP method and URI path
<a href="#">“List the z/OS data sets on a system”</a> on page 569	GET /zosmf/restfiles/ds? dslevel=<dataset_name_pattern>[&volser=<volser>&start=<dsname>]
<a href="#">“List the members of a z/OS data set”</a> on page 572	GET /zosmf/restfiles/ds/<dataset_name>/member? start=<member>&pattern=<mem-pat>
<a href="#">“Retrieve the contents of a z/OS data set or member”</a> on page 575	GET /zosmf/restfiles/ds/[-(<volser>)]<data-set-name>[(<member-name>)]
<a href="#">“Write data to a z/OS data set or member”</a> on page 580	PUT /zosmf/restfiles/ds/[-(<volser>)]<data-set-name>[(<member-name>)]
<a href="#">“Create a sequential or partitioned data set”</a> on page 585	POST /zosmf/restfiles/ds/<data-set-name>
<a href="#">“Delete a sequential and partitioned data set”</a> on page 588	DELETE /zosmf/restfiles/ds/<data-set-name> DELETE /zosmf/restfiles/ds/-(<volume>)/<data-set-name>
<a href="#">“Delete a partitioned data set member”</a> on page 590	DELETE /zosmf/restfiles/ds/<dataset-name>(<member-name>)
<a href="#">“z/OS data set and member utilities”</a> on page 592	PUT /zosmf/restfiles/ds/<to-data-set-name>
<a href="#">“Access Method Services Interface”</a> on page 596	PUT /zosmf/restfiles/ams

Table 336. Operations provided through the z/OS data set and file REST interface services. (continued)

Operation	HTTP method and URI path
<a href="#">“List the files and directories of a UNIX file path” on page 598</a>	GET /zosmf/restfiles/fs?path=<file-path-name>
<a href="#">“Retrieve the contents of a z/OS UNIX file” on page 603</a>	GET /zosmf/restfiles/fs/<file-path-name>
<a href="#">“Write data to a z/OS UNIX file” on page 607</a>	PUT /zosmf/restfiles/fs/<filepath-name>
<a href="#">“Create a UNIX file or directory” on page 610</a>	POST /zosmf/restfiles/fs/<file-path-name>
<a href="#">“Delete a UNIX file or directory” on page 613</a>	DELETE /zosmf/restfiles/fs/<file-pathname>
<a href="#">“z/OS UNIX file utilities” on page 615</a>	PUT /zosmf/restfiles/fs/<file-path-name>
<a href="#">“List z/OS UNIX Filesystems” on page 621</a>	GET /zosmf/restfiles/mfs/
<a href="#">“Create a z/OS UNIX zFS filesystem” on page 623</a>	POST /zosmf/restfiles/mfs/zfs/<file-system-name>
<a href="#">“Delete z/OS UNIX zFS Filesystem” on page 625</a>	DELETE /zosmf/restfiles/mfs/zfs/<file-system-name>
<a href="#">“Mount a z/OS UNIX file system” on page 626</a>	PUT /zosmf/restfiles/mfs/<file-system-name>
<a href="#">“Unmount a UNIX file system” on page 629</a>	PUT /zosmf/restfiles/mfs/<file-system-name>

## Using the Swagger interface

You can use the Swagger interface to display information about the z/OS data sets and files REST APIs. The Swagger interface includes four sections: AMS APIs, Dataset APIs, File APIs, and Filesystem APIs. For more information, see [“Using the Swagger interface” on page 1](#).

## Processing overview

The z/OS data set and file REST interface services can be invoked by any client application, running on the local z/OS system or a remote system. Your program (the client) initiates a request to the server through a

standard HTTP request method, such as GET or PUT. If the server determines that the request is valid, it performs the requested service and returns an HTTP response to your program.

For a successful request, this response takes the form of an HTTP 2nn status code and, if applicable, a result set that is passed back to your program. Depending on which service is requested, the result set might be returned in a format that requires parsing by your program, such as a JSON object. In other cases, the results might be returned in another format, such as plain text or binary data.

For an unsuccessful request, the server response consists of a non-OK HTTP response code and details of the error, which are provided in the form of a JSON object.

The contents of the JSON objects are described in [“JSON document specifications for z/OS data set and file REST interface requests”](#) on page 631.

**Note:**

If the URL contains a reserved character such as # \$ @, it must be URL-encoded so that it can be escaped. For example:

GET /zosmf/restfiles/ds/SYS1.PROCLIB(#ABC) HTTP/1.1

Should be changed to:

GET /zosmf/restfiles/ds/SYS1.PROCLIB(%23ABC) HTTP/1.1

## Common HTTP Request Headers

### X-IBM-Async-Threshold = <nnn>

This header can be added to a request to enable support for asynchronous responses with the HTTP status code 202 Accepted. This specifies the number of seconds that the client wants to wait for a response before receiving a 202 Accepted response. This response includes a Location response header with the URL (excluding protocol, host, and port) that can be used with a subsequent GET method request to obtain the results from the original request. Each subsequent request includes its own X-IBM-Async-Threshold header if additional async responses are required. The value of this header must be an integer 0 - 60 seconds. A value of 0 indicates that an async response is returned if the actual response is not immediately available. If X-IBM-Async-Threshold is specified, then X-IBM-Response-Timeout does not apply and is ignored. A DELETE method request can also be sent to the URL returned in an asynchronous response to abandon the original request and terminate the associated CEA TSO address space.

### Example of an Asynchronous request

Request:

```
GET /zosmf/restfiles/ds?dslevel=D10 HTTP/1.1
X-IBM-Async-Threshold: 3
```

Response:

```
202 Accepted
X-Powered-By: Servlet/3.0
Location: /zosmf/restfiles/queue/FS11fae7
X-IBM-Txid: tx000000000000D159
Content-Language: en-US
Content-Length: 0
Date: Fri, 18 Nov 2016 07:17:21 GMT
```

Request:

```
GET GET /zosmf/restfiles/queue/FS11fae7 HTTP/1.1 HTTP/1.1
X-IBM-Async-Threshold: 3
```

Response:

```
200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 39199
X-IBM-Txid: tx000000000000D160
Date: Fri, 18 Nov 2016 07:18:32 GMT
{"items":[
  {"dsname":"D10"},

```

```
{ "dsname": "D10.$DATA.SETS" },
{ "dsname": "D10.AAAAA" },
{ "dsname": "D10.AACTIVE.JCL" },
{ "dsname": "D10.AA11797.R1K.D050701A" },
{ "dsname": "D10.AA12484.HDZ11K0" },
{ "dsname": "D10.AA12484.HDZ11K0.TRSD" },
{ "dsname": "D10.ABACKUP.SM02631.FPGA.D14163.T131433" },
{ "dsname": "D10.ABARS1.C.C01V0001" },
```

### **X-IBM-Response-Timeout = <nnn>**

This header specifies the number of seconds that a TsoServlet request runs before a timeout occurs and an exception is returned to the client. This time does not include the time that can be required to start a new CEATSO address space. The default is 30 seconds, and the allowed range for this value is 5 - 600 seconds. An invalid value that is supplied for this header is converted into the closest valid value and the request proceeds.

**Note:** This timeout does not affect any timeouts that might occur in the z/OSMF WebSphere container or the REST service client.

### **X-IBM-Session-Limit-Wait = <nnn>**

This header specifies the number of seconds that a request waits if the limit for CEA TSO addresses spaces, as defined by CEA TSO services, is reached. Without X-IBM-Session-Limit-Wait, when a request exceeds the limit on CEA TSO address spaces, it returns an exception. With X-IBM-Session-Limit-Wait, you can direct the request to wait for up to 120 extra seconds for an available CEA TSO address space.

When X-IBM-Session-Limit-Wait is enabled, the request is retried at 1-second intervals until either an available CEA TSO address space is found or the new time limit is exceeded. If the new time limit is exceeded, the request returns the same exception as when X-IBM-Session-Limit-Wait is not enabled.

The allowed range for this value is 0 - 120 seconds. The default is 0 seconds, which means that the request does not wait and instead immediately return an exception. An invalid value that is supplied for this header, for example 121 or A, is ignored and the default value of 0 is used.

**Note:** Enabling X-IBM-Session-Limit-Wait generates a CEA0403I message in your console log for every second that it tries to find an available CEA TSO address space. If you do not want this clutter in your console log, consider suppressing the CEA0403I message.

### **X-IBM-Target-System**

This header indicates the target system for the request, where the target system name (nick name) is defined in the Systems table of the local system. The target host must support single-sign-on using an LTPA token. If the target system is the local system, this header is ignored.

## **Common HTTP Response Headers**

### **X-IBM-Txid = <string>**

This header returns the transaction ID that was assigned by z/OSMF to the request. It can be useful for diagnostic purposes to identify z/OSMF log records relating to a failed transaction. The transaction ID is also logged in the TSO address space. The transaction ID should not be used for other purposes; its format can change; and it might not be present in future releases.

### **Content-Encoding**

This header is used to compress the response data. If present, its value indicates which encoding method can be used to decompress the media-type that was specified in the Accept-Encoding header. The supported encoding method is gzip, which is specified as follows:

```
Content-Encoding: gzip
```

The response content is compressed by this method, if the content exceeds 4096 bytes.

## **Specifying an entity tag with your read and write requests**

During request processing, your program's access to the resource (a data set or file) is serialized by z/OS. No other users can read the resource or write to it, thus preventing concurrent updates of the resource from overwriting each other. Serialization, however, ends with the completion of the request. If your program must perform multiple read and write requests on a resource, you require a method of ensuring that the resource is not modified between your program's requests. Otherwise, you might overlay another user's changes.

To help you to ensure the integrity of a resource between requests, the z/OS data set and file REST interface supports the use of an *entity tag* (or *ETag*) on your requests. Obtained on the initial read (GET) request, the ETag is an identifier that is assigned by the web server to a specific version of the resource. If the resource content changes, a new ETag is associated with the resource.

To determine whether a resource was changed between requests, your program supplies its ETag as a header value on each request. If the ETag matches the current ETag for the resource, the system considers the resource to be unchanged and performs the request. Otherwise, the system fails the request; your program must obtain the ETag again before it can perform the request.

Generally, the process of updating the contents of a z/OS data set or UNIX file is as follows:

1. Retrieve the current content of the resource by using a GET request.
2. The server returns the contents of the resource in the response body as plain text, along with information about the resource, in the response headers:
  - Content-Length header specifies the length of the data that was returned
  - ETag header specifies the ETag that identifies the current version of the resource.
3. Replace the contents of the resource by using a PUT request. The request includes the following headers:
  - A request header to supply the ETag that was returned from the previous GET request on that resource. If the token still matches, the resource was not changed since the previous GET request. If the supplied token does not match a currently valid token, the PUT request fails with an HTTP 412. This response indicates that the host system file was modified in the time since the read operation was performed.
  - Optionally, a request header to specify whether data conversion is required.

For a PUT request, the request body contains the new contents of the file.

After the data is written, the 204 No Content response is returned with a new ETag, for use with any subsequent read or write requests.

Suppose you only want to replace a resource with a new copy, without first reading the contents of the current resource. To overlay the resource, have your program issue the initial GET request to obtain the ETag. Here, you would specify a maximum read amount of 0. Then, have your program issue a PUT request with the ETag and the new data to be written to the resource.

## Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. For the detailed format of each JSON object, see ["JSON document specifications for z/OS data set and file REST interface requests" on page 631](#).

## Required installation

To enable the z/OS data set and file REST interface services, IBM supplies a default procedure in your z/OSMF order, which you must install before you configure z/OSMF. For information, see [Configure the z/OS data set and file REST services in IBM z/OS Management Facility Configuration Guide](#).

## Required authorizations

Generally, your user ID requires the same authorizations for using the z/OS data set and file REST interface services as when you perform these operations through a TSO/E session on the system. For

example, listing the members of a z/OS data set through the z/OS data set and file REST interface requires authorization to start TSO on the system and access to the specified data set.

In addition, your user ID requires authorization to the z/OSMF SAF profile prefix on the target z/OS system, as follows:

- READ access to <IZU\_SAF\_PROFILE\_PREFIX> in the APPL class.
- READ access to the <IZU\_SAF\_PROFILE\_PREFIX>.\*.izuUsers profile in the EJBROLE class.

By default, the z/OSMF SAF profile prefix is IZUDFLT.

Where applicable, further authorization requirements are noted in the descriptions of the individual z/OS data set and file REST interface services.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document” on page 640](#).

The following HTTP status codes are valid:

### HTTP 200 OK

Request was processed successfully.

### HTTP 204 No content

Request was processed successfully, however, no content was returned. This status is normal for some types of requests, such as when no data sets or files match the filter criteria, or the specified partitioned data set has no members.

### HTTP 206 Partial content

Request was processed successfully, however, only a portion of the available content was received. The request contained the X-IBM-Max-Items header, which limited the amount of content that was returned.

### HTTP 304 Not Modified

An ETag token was included in the request. z/OSMF determined that the requested resource did not change since the ETag token was created.

### HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

### HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

### HTTP 404 Not found

Requested resource does not exist.

### HTTP 405 Method not allowed

Requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

### HTTP 412 Precondition failed

The supplied ETag token indicated that the resource was modified since the token was created. Therefore, the request failed the If-Match precondition that was specified in the header.

### HTTP 413 Request entity too large

The supplied data is too large to process. Or, the requested resource is too large to return.

### HTTP 429 Too many requests

The client submitted too many unsuccessful login attempts.

### HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

### HTTP 503 Service unavailable

Server is not available.

## Error logging

Errors from the z/OS data set and file REST interface services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## List the z/OS data sets on a system

You can use this operation to list the data sets on a z/OS system. You can filter the returned list of data set names through the specification of high-level qualifiers and wildcards.

### HTTP method and URI path

---

```
GET /zosmf/restfiles/ds/?dslevel=<filter-criteria>[&start=dsname]
```

---

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **?dslevel=<dataset-name-pattern>[&volser=<volser>&start=<dsname>]** represents the query parameters used to qualify the request, such as a data set name and, optionally, a volume serial (VOLSER).

### Standard headers

None.

### Custom headers

Include the following custom HTTP header with this request:

#### X-IBM-Max-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

#### X-IBM-Attributes

This header specifies whether the results are to include the data set base or volume attributes.

#### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request

- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

You can specify the following query parameter on this request:

### **dslevel**

The search parameter that identifies the cataloged data sets to be listed. Either the dslevel or volser parameter must be specified and can be a fully qualified data set name or a partial data set name with a filter to display a list of matches. A partial data set name can include:

- One or more high-level qualifiers or name segments
- One or more wildcard symbols: percent sign (%), asterisk (\*), or double asterisk (\*\*)
- A percent sign is a single character wildcard.
- An asterisk is any number of characters within a qualifier.
- A double asterisk is any number of characters within any number of lower-level qualifiers.

The parameter values must be URL-encoded, otherwise you may receive an error message. If you use the percent sign (%) as a wildcard to filter the list of data sets returned, you must enter %25 to avoid receiving this error message: URLDecoder: Incomplete trailing escape (%) pattern. For example:

```
GET /zosmf/restfiles/ds?dslevel=sys%25d.*lib HTTP/1.1
```

### **Notes:**

1. The length of the data set name that you specify on the request cannot exceed 44 characters. The length limit includes wildcards, which are treated as one character each. The wildcard %25 is treated as one character.
2. The system appends the following to any filter criteria that you specify: .\*\*
3. Lowercase characters are automatically folded to uppercase.

### **volser**

A parameter that identifies the volume serials to be searched for data sets with names that match the specified **dslevel** parameter. The volume serial is one to six characters. You cannot use wildcard characters for this parameter. If you omit this parameter, the cataloged data set name is returned by default. If this parameter is specified, the data sets on the volume that match the **dslevel** pattern are returned.

### **start**

An optional search parameter that specifies the first data set name to return in the response document. The length of the data set name that you specify cannot exceed 44 characters, and cannot contain wildcards. If the data set name is not found for the given search, then the next data set matching the search will be returned.

## **X-IBM-Attributes**

### **dsname**

Requests that only data set names be returned. If you omit this header, it is set to "dsname".

### **base**

Setting the X-IBM-Attribute to base returns all of the basic attributes for the data set being queried. These attributes are commonly found in the **ISPF List Data set panel**. The base key is mutually exclusive with volser, and dsname.

## vol

Setting the X-IBM-Attribute to vol returns the volume where the data set resides. If the dataset is a multi-volume dataset, use the base option to list all the volumes.

## ,total

The suffix ,total, can be added to request that the "totalRows" property is returned if more data sets than the maximum requested are available.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Example request

In the following example, the GET method is used to list all of the cataloged data sets that match the partial name IBMUSER.CONFIG.\*.

```
GET /zosmf/restfiles/ds?dslevel=IBMUSER.CONFIG.* HTTP/1.1
```

## Example response

A sample response is shown in [Figure 302 on page 571](#).

Response:

```
200 OK
x-powered-by: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 201
Content-Language: en-US
Date: Mon, 23 Nov 2015 09:10:11 GMT
```

Response Body:

```
{
  "items": [
    {
      "dsname": "IBMUSER.CONFIG.FS",
      "dsname": "IBMUSER.CONFIG.FS.DATA",
      "dsname": "IBMUSER.CONFIG.ORIG.FS",
      "dsname": "IBMUSER.CONFIG.ORIG.FS.DATA"
    }
  ],
  "returnedRows": 4,
  "JSONversion": 1
}
```

*Figure 302. Example: list all of the data sets.*

## Example request

The GET method is used to list all of the cataloged data sets with specified base attributes.

```
GET /zosmf/restfiles/ds?dslevel=***&volser=PEVTS2 HTTP/1.1
```

Request Headers:

```
X-IBM-Attributes: 'base'
```

## Example response

A sample response is shown in [Figure 303 on page 572](#).

Response:

```
200 OK
x-powered-by: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 714
Content-Language: en-US
Date: Mon, 23 Nov 2015 09:11:46 GMT
```

Response Body:

```
{
  "items": [
    {
      "dsname": "IBMUSER.CONFIG.FS", "catnm": "CATPAK.MASTER.CATALOG", "dsorg": "VS",
      "migr": "NO", "mvol": "N", "vol": "*VSAM*",
      "dsname": "IBMUSER.CONFIG.FS.DATA", "blksz": "?", "catnm": "CATPAK.MASTER.CATALOG",
      "cdate": "2011/08/14", "dev": "3390", "dsorg": "VS", "edate": "***None***", "extx": "1",
      "lrecl": "?", "migr": "NO", "mvol": "N", "ovf": "NO", "rdate": "2015/07/28", "recfm": "?",
      "size": "14250", "spacu": "CYLINDERS", "used": "?", "vol": "CIMSSR",
      "dsname": "IBMUSER.CONFIG.ORIG.FS", "catnm": "CATPAK.MASTER.CATALOG", "dsorg": "VS",
      "migr": "NO", "mvol": "N", "vol": "*VSAM*",
      "dsname": "IBMUSER.CONFIG.ORIG.FS.DATA", "catnm": "CATPAK.MASTER.CATALOG",
      "dev": "3390", "migr": "NO", "mvol": "N", "vol": "PEVTS2",
      "dsname": "IBMUSER.MULTIVOL.DATA", "catnm": "CATPAK.MASTER.CATALOG",
      "dev": "3390", "migr": "NO", "mvol": "Y", "vol": "PEVTS2", "vols": "PEVTS2,PEVTS3,PEVTS4"
    }
  ],
  "returnedRows": 5, "JSONversion": 1
},
  "returnedRows": 4, "JSONversion": 1
}
```

Figure 303. Example: List all of the cataloged data sets with specified base attributes.

## List the members of a z/OS data set

You can use this operation to list the members of a z/OS partitioned data set.

### HTTP method and URI path

```
GET /zosmf/restfiles/ds/<dataset-name>/member?start=<member>&pattern=<mem-pat>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **/<dataset-name>** identifies the data set for which members are to be listed. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters. You cannot use wildcard characters for this parameter.
- **/member** indicates that member names are to be returned.

### Standard headers

None.

### Custom headers

Include the following custom HTTP headers with this request:

#### X-IBM-Max-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

#### X-IBM-Attributes

This header is optional.

##### member

A request that only member names be returned. If you omit this header, it is set to "member".

**base**

Setting the X-IBM-Attribute to base returns all of the basic attributes for the data set member being queried. These attributes are commonly found in the **ISPF List Data set panel**. The base key is mutually exclusive with member.

**,total**

The suffix ,total, can be added to request that the "totalRows" property is returned if more data set members than the maximum requested are available.

**X-IBM-Migrated-Recall**

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values may be specified too:

**wait**

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

**nowait**

If the data set is migrated, request it to be recalled, but do not wait.

**error**

If the data set is migrated, do not attempt to recall the data set.

**X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

**X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

**Query parameters****start**

An optional search parameter that specifies the first member name to return in the response document. The length of the data set name that you specify cannot exceed 8 characters, and cannot contain wildcards. If the member name is not found for the given search, then the next member matching the search is returned.

**pattern**

An optional search parameter restricts the returned member names to only the names that match the given pattern. The syntax of this argument is the same as "pattern" parameter of the ISPF LMMLIST service.

**Required authorizations**

See [“Required authorizations” on page 567](#).

**Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For a successful request, the HTTP response includes an array of data set members, each as one of the following types of JSON list document:

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example request

In the following example, the GET method is used to list all of the members of a data set.

```
GET /zosmf/restfiles/ds/SYS1.PROCLIB/member HTTP/1.1
```

## Example response

A sample response is shown in [List all of the members of a data set](#).

### Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 235
Content-Language: en-US
Date: Tue, 24 Nov 2015 05:31:51 GMT
```

### Response Body

```
{
  "items": [
    { "member": "CREATECD" },
    { "member": "SPROCLA1" },
    { "member": "TESTJCL" },
    { "member": "WASACR" },
    { "member": "WLMCD" },
    { "member": "XRACFH" },
    { "member": "XRACFHT" },
    { "member": "XRACFH2" }
  ],
  "returnedRows": 8,
  "JSONversion": 1
}
```

*Figure 304. Example: List all of the members of a data set*

## Example request

In the following example, the GET method is used to list all of the members of a data set with specified base attributes.

```
GET /zosmf/restfiles/ds/SYS1.PROCLIB/member HTTP/1.1
```

### Request Headers:

```
X-IBM-Attributes: 'base'
```

## Example response

A sample response is shown in [List all of the members of a data set with specified base attributes](#).

## Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 1287
Content-Language: en-US
Date: Tue, 24 Nov 2015 05:33:57 GMT
```

## Response Body

```
{
  "items": [
    {
      "member": "CREATECD", "vers": 1, "mod": 0, "c4date": "2015/08/12", "m4date": "2015/08/12", "cnoirc": 22,
      "inoirc": 22, "mnoirc": 0, "mtime": "05:48", "msec": "43", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "SPROCLA1", "vers": 1, "mod": 12, "c4date": "2009/10/16", "m4date": "2014/09/18", "cnoirc": 132,
      "inoirc": 122, "mnoirc": 0, "mtime": "07:55", "msec": "23", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "TESTJCL", "vers": 1, "mod": 0, "c4date": "2015/07/29", "m4date": "2015/07/29", "cnoirc": 22,
      "inoirc": 22, "mnoirc": 0, "mtime": "01:49", "msec": "36", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "WASACR", "vers": 1, "mod": 0, "c4date": "2015/08/14", "m4date": "2015/08/14", "cnoirc": 22,
      "inoirc": 22, "mnoirc": 0, "mtime": "04:44", "msec": "19", "user": "IBMUSER", "sclm": "N"},
    {
      "member": "XRACFH", "vers": 1, "mod": 1, "c4date": "2005/09/26", "m4date": "2005/11/03", "cnoirc": 514,
      "inoirc": 506, "mnoirc": 8, "mtime": "11:10", "msec": "45", "user": "HDENNIS", "sclm": "N"},
    {
      "member": "XRACFHT", "vers": 1, "mod": 0, "c4date": "2005/11/04", "m4date": "2005/11/04", "cnoirc": 130,
      "inoirc": 130, "mnoirc": 0, "mtime": "11:28", "msec": "12", "user": "HDENNIS", "sclm": "N"},
    {
      "member": "XRACFH2", "vers": 1, "mod": 0, "c4date": "2005/11/04", "m4date": "2005/11/04", "cnoirc": 130,
      "inoirc": 130, "mnoirc": 0, "mtime": "11:27", "msec": "43", "user": "HDENNIS", "sclm": "N"}
  ],
  "returnedRows": 8, "JSONversion": 1
}
```

Figure 305. Example: List all of the members of a data set with specified base attributes.

## Retrieve the contents of a z/OS data set or member

You can use this operation to retrieve the contents of a sequential data set, or a member of a partitioned data set (PDS or PDSE). To retrieve the contents of an uncataloged data set, include the volume serial on the request.

### HTTP method and URI path

```
GET /zosmf/restfiles/ds/[-(<volser>)]<dataset-name>[(<member-name>)]
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **-(<volser>)** represents a volume serial. For an uncataloged data set, include this parameter to identify the volume to be searched for data sets or members that match the specified **<data-set-name>** or **<member-name>**. The length of the volume serial cannot exceed six characters. You cannot use wildcard characters for this parameter. Indirect volume serials are not supported.
- **<dataset-name>** identifies the data set to be read. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters.
- **<member-name>** identifies the name of the PDS or PDSE member to be read. Include this parameter for a member read request.

Based on the object to be read, you can specify one of the following parameter combinations:

- **/<data-set-name>**: To retrieve data from a sequential data set.
- **/<data-set-name>(<member-name>)**: To retrieve data from a member of a PDS or PDSE.
- **/-(<volser>)/<data-set-name>**: To retrieve data from an uncataloged sequential data set.
- **/-(<volser>)/<data-set-name>(<member-name>)**: To retrieve data from a member of an uncataloged PDS or PDSE.

## Optional Query Parameters

### **search=<string>**

The data set is searched for the first record that contains the string, without respect to case (by default).

Optionally, insensitive=false may be specified for case sensitive matching.

This parameter may not be specified with the research= parameter.

### **research=<regular-expression>**

The data set is searched for the first record that matches the given extended regular expression.

This parameter may not be specified with the search= parameter.

Implementation note: the regcomp() C Library function with the REG\_EXTENDED flag is used.

### **insensitive=true|false**

The default is 'true'. When 'true', searches (search and research) are case insensitive. For case sensitive searches, specify 'false'.

### **maxreturnsize=<integer>**

This parameter may be specified only with search= or research=.

The value given is the maximum number of records to return.

The default, if not specified, is 100.

For the search and research queries, records are returned starting with the first matching record. The X-IBM-Record-Range request header may be used to specify the range of records to be searched, but it will not restrict the number of records returned (see maxreturnsize).

If no X-IBM-Record-Range request header is present, the search will begin with the first record. In all cases, an X-IBM-Record-Range=p,q response header will be returned where p is the first matching record and q is the number of records returned.

If no matching records are found, the response header X-IBM-Record-Range=0,0 will be returned.

The parameter may not be used if a request header X-IBM-Data-Type specifies any option except 'text'.

## Standard headers

You can include the following standard HTTP header with this request:

### **If-None-Match**

This header is optional; use it to specify the ETag token to be used to correlate this request with a previous request. If the data on the z/OS host has not changed since the ETag token was generated, z/OSMF returns a status of HTTP 304 Not Modified.

For the initial request to the resource, you can omit this header.

**Note:** If this header is used with very large data sets then performance may be impacted since the data set may have to be read twice by the system. This header is ignored if X-IBM-Record-Range is specified (see below). The ETag response header may be returned containing a hash string. See [“X-IBM-Return-ETag” on page 577](#) for details on whether this header will be present.

## Custom headers

You can include the following custom HTTP header with this request:

### **X-IBM-Data-Type**

This header is optional; use it to indicate whether data conversion is to be performed on the returned data, as follows:

- When set to text, data conversion is performed. The data transfer process converts each record from EBCDIC to the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. A newline (NL) character from the response charset is inserted between logical records. For data sets with fixed-length records, trailing blanks are removed.

A value "text;fileEncoding=<codepage>" can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

**Note:** An alternate file encoding cannot be specified with the "research" query parameter.

- When set to `binary`, no data conversion is performed. The data transfer process returns each record as-is, without translation. No delimiters are added between records. The response Content-Type is "application/octet-stream".
- When set to `record`, no data conversion is performed. Each logical record is preceded by the 4-byte big endian record length of the record that follows. This length does not include the prefix length. For example: a zero-length record is 4 bytes of zeros with nothing following.

If you omit this header, the default is `text`; the response is converted.

### **X-IBM-Return-Etag**

This header is optional; set it to 'true' to force the response to include an "Etag" header, regardless of the size of the response data. If this header is not present or set to something other than 'true', then the default is to only send an Etag in the response for data sets smaller than a system determined length, which is at least 8MB. If X-IBM-Record-Range is present, then this header may not be specified with the value "true" and an Etag will never be returned.

If this header is enabled for very large data sets, then performance is impacted since the data set must be read twice by the system.

### **X-IBM-Migrated-Recall**

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values may be specified too:

#### **wait**

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

#### **nowait**

If the data set is migrated, request it to be recalled, but do not wait.

#### **error**

If the data set is migrated, do not attempt to recall the data set.

### **X-IBM-Record-Range**

Use this header to retrieve a range of records from a data set. You can specify this range using either of the following formats:

#### **SSS-EEE**

Where SSS identifies the start record and EEE identifies the end record to be retrieved. Both values are relative offsets (0-based).

When EEE is set to 0, records through the end of the data set are retrieved.

When SSS is omitted (i.e. -EEE), the final EEE records of the data set are retrieved.

#### **SSS,NNN**

Where SSS identifies the start record and NNN identifies the number of records to be retrieved.

NNN must be greater than zero.

#### **Usage notes:**

If X-IBM-Record-Range is specified, then an ETag header will not be returned and the If-None-Match request header is ignored.

If X-IBM-Record-Range header is present on the request, then header X-IBM-Return-Etag=true may not be specified.

If no records are found in the range specified, an exception is returned.

### **X-IBM-Obtain-ENQ**

This header is optional; set it to one of the following values to request that a system ENQ be obtained and held after the completion of this request. If not specified, then no ENQs will be held after the completion of this request.

**EXCLU**

a SYSDSN/Exclusive ENQ will be held on the data set

**SHRW**

a SYSDSN/SHR ENQ will be held on the data set, and a SPFEDIT/EXCLU ENQ will be held on the data set, including the member name if this is a request for a PDS member.

A successful response will include an X-IBM-Session-Ref response header that can be added as a request header to subsequent requests to specify affinity to the TSO address space holding this ENQ.

**X-IBM-Session-Ref**

This header is optional; include it with the value returned from a previous X-IBM-Session-Ref response header to indicate that your request should be executed in the TSO address space that was previously reserved with a X-IBM-Obtain-ENQ request header. This address space will not be used for other requests and if not used at least once every 10 minutes it will be terminated.

The following URL request may be used to "ping" the reserved address space to keep it alive:

GET https://zosmf1.yourco.com/zosmf/restfiles/ping HTTP/1.1

X-IBM-Session-Ref: xxxxxx

The X-IBM-Obtain-ENQ and X-IBM-Session-Ref headers are mutually exclusive.

**X-IBM-Release-ENQ**

This header is optional; it may be specified with a value "true" to request that the ENQ held by the associated TSO address space be released.

This header must be specified along with a valid X-IBM-Session-Ref header.

**X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

**X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

**Required authorizations**

See [“Required authorizations” on page 567](#).

**Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

**Expected response**

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. Status code 304 indicates unchanged content when a conditional get is performed (such as when using the **If-None-Match** header with an ETag from a previous response). A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See “[Error report document](#)” on page 640.

### Example request

In the following example, the GET method is used to retrieve the contents of the member SMFPRM00 in data set SYS1.PARMLIB.

```
GET /zosmf/restfiles/ds/SYS1.PARMLIB(SMFPRM00) HTTP/1.1
```

### Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 200)
- ETag that you can use on subsequent requests to test for changes to the resource
- Content-Length response header that specifies the amount of data that was returned (in bytes)
- A response body that contains the resource in plain text.

A sample response header is shown in [Figure 306](#) on page 579.

```
200 OK
Etag: B5C6454F783590AA8EC15BD88E29EA63
Content-Type: text/plain; charset=UTF-8
Content-Language: en-US
Content-Length: 1944
Date: Fri, 07 Nov 2014 02:13:07 GMT
Connection: close
```

*Figure 306. Example: Returned contents of the SMFPRM00 member of sys1.parmlib*

A sample response body is shown in [Figure 307](#) on page 579.

```
ACTIVE                                /*ACTIVE SMF RECORDING*/          00010000
DSNAME(SYS1.&SMFDSN1,SYS1.&SMFDSN2,   /*SMF ON 3390 */              00020000
SYS1.&SMFDSN3)                        /*FT: SYSAQ3, TS: SYSAQ4 */      00030000
NOPROMPT                             /*PROMPT THE OPERATOR FOR OPTIONS*/ 00040000
REC(PERM)                             /*TYPE 17 PERM RECORDS ONLY*/      00050000
MAXDORM(3000)                         /* WRITE AN IDLE BUFFER AFTER 30 MIN*/ 00060000
MEMLIMIT(256M)                       /* 256M FOR 64 BIT APPS */        00061005
STATUS(003000)                       /* WRITE SMF STATS AFTER HALF HOUR*/ 00070000
JWT(0700)                             /* INVOKE EXIT IEFUTL AFTER 7HR 00M*/ 00080002
SID(&SYSNAME),                       /* SYSTEM ID FOR 3084 - SINGLE IMAGE*/ 00090000
LISTDSN                              /* LIST DATA SET STATUS AT IPL*/    00100000
INTVAL(30)                           /* INTVAL OPTION SP430 */         00110000
SYNCVAL(00)                          /* SYNCVAL OPTION SP430 */         00120000
SYS(NOTYPE(19,40,92),                00130001
  EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI,
    IEFUSI,IEFUTL,IEFU29),INTERVAL(010000),DETAIL) 00140000
/* WRITE ALL RECORDS AS THE SYSTEM DEFAULT, TAKE ALL KNOWN 00160000
EXITS, NOTE: JES EXITS CONTROLLED BY JES , THERE IS NO    00170000
DEFAULT INTERVAL RECORDS WRITTEN AND ONLY SUMMARY T32    00180000
RECORDS AS A DEFAULT FOR TSO */                          00190000
SUBSYS(STC,NOTYPE(19,40,92),          00200000
  EXITS(IEFU29,IEFU83,IEFU84,IEFUTL), 00210000
  INTERVAL(SMF,SYNC),DETAIL)          00220001
/*SP430*/                                                  00230000
/* WRITE ALL RECORDS AS BY SYSTEM DEFAULT, TAKE ONLY THREE 00240000
EXITS, NOTE: IEFU29 EXECUTES IN THE MASTER ASID WHICH IS A 00250000
STC ADDRESS SPACE SO IEFU29 MUST BE ON FOR STC. USE ALL OTHER 00260000
SYS PARAMETERS AS A DEFAULT */          00270000
                                          00280000
                                          00290000
```

*Figure 307. Example: Returned contents of the SMFPRM00 member of sys1.parmlib*

## Example request

In the following example, the GET method is used to retrieve the contents of a sequential data set.

```
GET /zosmf/restfiles/ds/JIAHJ.REST.SRVMP HTTP/1.1
```

## Example response

A sample response body is shown in Figure 308 on page 580.

### Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: text/plain; charset=UTF-8
Content-Length: 2131
Etag: 47029CDDCD91E2887E1FAAD6FCD75ECB
Content-Language: en-US
Date: Wed, 25 Nov 2015 02:27:15 GMT
```

### Response Body

```
//JH2FPROC EXEC PGM=IKJEFT01,DYNAMNBR=200
//*****
//* TSO LOGON PROC FOR Z/OS DATA SET AND FILE REST INTERFACE */
//*
//* PROPRIETARY STATEMENT:
//*
//* LICENSED MATERIALS - PROPERTY OF IBM
//* 5610-A01
//* COPYRIGHT IBM CORP. 2014
//* STATUS = HSMA210
//*****
//CEEOPDS DD *
DYNDDUMP(*USERID.PRIVATE)
//SYSEXEC DDDISP=SHR,DSN=ISP.SISPEXEC
// DD DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSPROC DD DISP=SHR,DSN=ISP.SISPCLIB
// DD DISP=SHR,DSN=SYS1.SBPXEXEC
//ISPLLIB DD DISP=SHR,DSN=JIAHJ.REST.LMOD
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPTLIB DD RECFM=FB,LRECL=80,SPACE=(TRK,(1,0,1))
// DD DISP=SHR,DSN=ISP.SISPTENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPROF DD DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(15,15,5)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//IZUSRVMP DD DISP=SHR,DSN=JIAHJ.REST.PARMLIB(IZUSRVMP)
//SYSOUT DD SYSOUT=H
//CEEDUMP DD SYSOUT=H
//SYSUDUMP DD SYSOUT=H
//
```

Figure 308. Example: Retrieve the contents of a sequential data set

## Write data to a z/OS data set or member

You can use this operation to write data to an existing sequential data set, or a member of a partitioned data set (PDS or PDSE). To write to an uncataloged data set, include a volume serial on the request.

### HTTP method and URI path

```
PUT /zosmf/restfiles/ds/[-(<volser>)]<dataset-name>[(<member-name>)]
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **-(<volser>)** represents a volume serial. For an uncataloged data set, include this parameter to identify the volume to be searched for data sets or members that match the specified *<data-set-name>* or

*<member-name>*. The length of the volume serial cannot exceed six characters. You cannot use wildcard characters for this parameter. Indirect volume serials are not supported.

- ***<dataset-name>*** identifies the data set to which to write. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters.
- ***<member-name>*** identifies the name of the PDS or PDSE member to which to write. Include this parameter for a PDS or PDSE member write request.

If the member does not exist, it is created. If the data set name identifies a base name of a Generation Data Group (GDG), then member may refer to relative data sets, for example: (0), (+1), (-1)

Based on the object to which you want to write, you can specify one of the following parameter combinations:

- ***/<data-set-name>***: To write to a sequential data set.
- ***/<data-set-name>(<member-name>)***: To write to a member of a PDS or PDSE.
- ***/-(<volser>)/<data-set-name>***: To write to an uncataloged sequential data set.
- ***/-(<volser>)/<data-set-name>(<member-name>)***: To write to a member of an uncataloged PDS or PDSE.

## Request body

The data to write to the target data set. The data is interpreted according to the content-type as one of binary, text, record or 'diff -e' format according a combination of the "Content-Type" and the value of the X-IBM-Data-Type custom header, if present.

## Standard headers

You can include the following standard HTTP header with this request:

### If-Match

This header is optional; use it to specify the ETag to be used for correlating this request with a previous request on the same resource. If the resource has not changed since the ETag token was generated, the data is written to the target data set or member. Otherwise, if the resource has been modified, the request is failed with status code HTTP 412.

If you omit this header, the data is always written, regardless of whether the resource is changed.

## Custom headers

You can use the following custom HTTP header with this request:

### X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the request body.

#### text

When set to `text`, data conversion is performed. The data transfer process converts each record from the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. Each line of data, delimited by a Line Feed in the request charset, is converted to EBCDIC and written as a record to the data set or member. (The line feed character is removed from the data, and the data is padded with the space character to the end of the record if it is a fixed record size data set. For variable record size data sets, the record is written without padding.) If the record size of the data set is smaller than any line of text, an HTTP 400 is returned with a JSON error document indicating that not all data was written.

A value `"text;fileEncoding=<codepage>"` can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

A value `text;CrLf=true` can be used to control whether each input text line is terminated with a carriage return line feed (CRLF), rather than a line feed (LF), which is the default.

**Note:** When set to 'text' and "Content-Type" is "application/x-ibm-diff-e", the input consists of commands in the same format as produced by the z/OS UNIX 'diff -e' command. These commands are used to add, replace and delete lines in the target data set. The following commands are supported:

```
a
c
d
s/.//
opt : g|<n>, g means global
n means search and replace <n> times
```

Each command may be optionally preceded by a line or line range, as allowed by the z/OS UNIX 'ed' command. If an error is detected while processing a command, status code 500 is returned with an exception.

#### **binary**

When set to binary, no data conversion is performed. The data is written to the data set without respect to record boundaries. All records will be written at their maximum record length and for fixed length record data sets, the last record will be padded with nulls if required.

#### **record**

When set to record, no data conversion is performed. Each logical record is preceded by the 4-byte big endian record length of the record that follows. This length does not include the prefix length. For example: a zero-length record would be 4 bytes of zeros with nothing following.

If you omit this header, the default is text; the request body is converted.

### **X-IBM-Migrated-Recall**

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values may be specified too:

#### **wait**

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

#### **nowait**

If the data set is migrated, request it to be recalled, but do not wait.

#### **error**

If the data set is migrated, do not attempt to recall the data set.

### **X-IBM-Obtain-ENQ**

This header is optional; set it to one of the following values to request that a system ENQ be obtained and held after the completion of this request. If not specified, then no ENQs will be held after the completion of this request.

#### **EXCLU**

a SYSDSN/Exclusive ENQ will be held on the data set

#### **SHRW**

a SYSDSN/SHR ENQ will be held on the data set, and a SPFEDIT/EXCLU ENQ will be held on the data set, including the member name if this is a request for a PDS member.

A successful response will include an X-IBM-Session-Ref response header that can be added as a request header to subsequent requests to specify affinity to the TSO address space holding this ENQ.

### **X-IBM-Session-Ref**

This header is optional; include it with the value returned from a previous X-IBM-Session-Ref response header to indicate that your request should be executed in the TSO address space that was previously reserved with a X-IBM-Obtain-ENQ request header. This address space will not be used for other requests and if not used at least once every 10 minutes it will be terminated.

The following URL request may be used to "ping" the reserved address space to keep it alive:

GET <https://zosmf1.yourco.com/zosmf/restfiles/ping> HTTP/1.1

X-IBM-Session-Ref: xxxxxx

The X-IBM-Obtain-ENQ and X-IBM-Session-Ref headers are mutually exclusive.

### **X-IBM-Release-ENQ**

This header is optional; it may be specified with a value "true" to request that the ENQ held by the associated TSO address space be released.

This header must be specified along with a valid X-IBM-Session-Ref header.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Required authorizations**

See [“Required authorizations” on page 567](#).

## **Request content**

Your request must supply the data set content. For an example, see [“Example request” on page 583](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## **Expected response**

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. Status code 201 indicates success if a new PDS member was created. Status code 412 indicates that the document does not match the supplied ETag token on the If-Match header as described above. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## **Example request**

Suppose that you want to update the contents of the SMFPRM00 member of SYS1.PARMLIB using a PUT request. Figure 309 on page 583 shows an example of the request header that you might use.

```
PUT /zosmf/restfiles/ds/SYS1.PARMLIB(SMFPRM00)
If-Match: B5C6454F783590AA8EC15BD88E29EA63
Content-Type: text/plain; charset=UTF-8
```

*Figure 309. Example: Request header for a write request to the SMFPRM00 member of sys1.parmlib*

In [Figure 309 on page 583](#), notice that the optional header If-Match is included. This header is specified with an ETag that was obtained from a previous read request on the parmlib member. Using an ETag in this manner allows for conditional processing; the new member contents are written only when the member has not been modified on the host system since the ETag was generated. If the member was modified, for example, by another user or process, this request is failed with HTTP status code 412.

A sample request body is shown in [Figure 310 on page 584](#). The body contains the new contents of the member.

```

/*****
/* THIS PARMLIB MEMBER CONTAINS CONFIGURATION FOR SMF */
*****/
ACTIVE                               /*ACTIVE SMF RECORDING*/          00010000
DSNAME(SYS1.&SMFDSN1,SYS1.&SMFDSN2,    /*SMF ON 3390 */              00020000
SYS1.&SMFDSN3)                        /*FT: SYSAQ3, TS: SYSAQ4 */      00030000
NOPROMPT                             /*PROMPT THE OPERATOR FOR OPTIONS*/ 00040000
REC(PERM)                            /*TYPE 17 PERM RECORDS ONLY*/    00050000
MAXDORM(3000)                        /* WRITE AN IDLE BUFFER AFTER 30 MIN*/ 00060000
MEMLIMIT(256M)                       /* 256M FOR 64 BIT APPS */        00061005
STATUS(003000)                       /* WRITE SMF STATS AFTER HALF HOUR*/ 00070000
JMT(0700)                            /* INVOKE EXIT IEFUTL AFTER 7HR 00M*/ 00080002
SID(&SYSNAME),                       /* SYSTEM ID FOR 3084 - SINGLE IMAGE*/ 00090000
LISTDSN                              /* LIST DATA SET STATUS AT IPL*/    00100000
INTVAL(30)                           /* INTVAL OPTION SP430 */        00110000
SYNCVAL(00)                          /* SYNCVAL OPTION SP430 */        00120000
SYS(NOTYPE(19,40,92),                00130001
EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI, 00140000
IEFUSI,IEFUTL,IEFU29),INTERVAL(010000),DETAIL) 00150000
/* WRITE ALL RECORDS AS THE SYSTEM DEFAULT, TAKE ALL KNOWN 00160000
EXITS, NOTE: JES EXITS CONTROLLED BY JES , THERE IS NO 00170000
DEFAULT INTERVAL RECORDS WRITTEN AND ONLY SUMMARY T32 00180000
RECORDS AS A DEFAULT FOR TSO */      00190000
SUBSYS(STC,NOTYPE(19,40,92),          00200000
EXITS(IEFU29,IEFU83,IEFU84,IEFUTL),    00210000
INTERVAL(SMF,SYNC),DETAIL) /*SP430*/ 00220001
/* WRITE ALL RECORDS AS BY SYSTEM DEFAULT, TAKE ONLY THREE 00230000
EXITS, NOTE: IEFU29 EXECUTES IN THE MASTER ASID WHICH IS A 00240000
STC ADDRESS SPACE SO IEFU29 MUST BE ON FOR STC. USE ALL OTHER 00250000
SYS PARAMETERS AS A DEFAULT */      00260000
                                     00270000
                                     00280000
                                     00290000

```

*Figure 310. Example: Request body for a write request to the SMFPRM00 member of sys1.parmlib*

### Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 204)
- ETag that you can use on subsequent requests to test for changes to the resource

```

204 No Content
Etag: DE2BE8B8485EB8F1E28D3716DFFE0680
Content-Type: application/json; charset=UTF-8
Content-Language: en-US
Date: Fri, 07 Nov 2014 02:31:39 GMT

```

### Example request

The PUT method is used to write the contents of a sequential data set.

```

PUT /zosmf/restfiles/ds/JIAHJ.REST.SRVMP HTTP/1.1
If-Match: B5C6454F783590AA8EC15BD88E29EA42
Content-Type: text/plain; charset=UTF-8

```

### Example response

A sample response is shown in [Contents of a sequential data set](#).

Response:

```
204 No Content
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Etag: 39E89731CE27214AE2FE0BB09200DC26
Content-Language: en-US
Date: Wed, 25 Nov 2015 03:10:12 GMT
```

*Figure 311. Example: Contents of a sequential data set*

### Example request

The PUT method is used to write the contents to a member of data set with regular expression.

```
PUT /zosmf/restfiles/ds/SYS1.PROCLIB(JH2FPROC) HTTP/1.1
s/a*b/c*c/g
```

### Example response

A sample response is shown in [Figure 312 on page 585](#).

Response:

```
204 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Etag: 8EF84322919807BB003EBF2DE067AD38
Content-Language: en-US
Date: Wed, 11 Oct 2017 05:20:10 GMT
```

*Figure 312. Example: Contents of a member of data set with regular expression.*

## Create a sequential or partitioned data set

You can use this operation to create sequential and partitioned data sets on a z/OS system.

### HTTP method and URI path

```
POST /zosmf/restfiles/ds/<dataset-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **<dataset-name>** is the name of a z/OS data set that you are going to create.

### Request Body

The request body to create a sequential or partitioned data set is shown in [Request body to create a sequential and partitioned data set](#).

Table 337. Request body to create a sequential or partitioned data set

Field	Type	Description
volser	String	Volume serial.
unit	String	Device type.
dsorg	String	Data set organization.
alcunit	String	Unit of space allocation.
primary	Integer	Primary space allocation.
secondary	Integer	Secondary space allocation.
dirblk	Integer	Number of directory blocks.
avgbk	Integer	Average block size.
recfm	String	Record format.
blksize	Integer	Block size.
lrecl	Integer	Record length.
storclass	String	Storage class.
mgntclass	String	Management class.
dataclass	String	Data class.
dsntype	String	Data set type.
like	String	Model data set name.

## Standard headers

None.

## Custom headers

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

None.

## Content type

The content type is application/json.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For a successful creating request, 201 Created with no content is returned.

## Example request

In the following example, the POST method is used to create a sequential data set.

```
POST /zosmf/restfiles/ds/JIAHJ.REST.TEST.NEWS HTTP/1.1
```

Request body:

```
{ "volser": "zmf046", "unit": "3390", "dsorg": "PS", "alcunit": "TRK", "primary": 10,
  "secondary": 5, "avgblk": 500, "recfm": "FB", "blksize": 400, "lrecl": 80 }
```

## Example response

A sample response is shown in [Example: Create a data set](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 10:54:21 GMT
```

*Figure 313. Example: Create a data set*

## Example request

In the following example, the POST method is used to create a partitioned data set.

```
POST /zosmf/restfiles/ds/JIAHJ.REST.TEST.NEWS02 HTTP/1.1
```

Request Body

```
{ "volser": "zmf046", "unit": "3390", "dsorg": "PO", "alcunit": "TRK", "primary": 10,
  "secondary": 5, "dirblk": 10, "avgblk": 500, "recfm": "FB", "blksize": 400, "lrecl": 80 }
```

## Example response

A sample response is shown in [Example: Create data set](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 11:14:13 GMT
```

*Figure 314. Example: Create data set.*

## Example request

In the following example, the POST method is used to create a PDSE data set.

```
POST /zosmf/restfiles/ds/JIAHJ.REST.TEST.NEWS02 HTTP/1.1
```

### Request Body

```
{ "volser": "zmf046", "unit": "3390", "dsorg": "PO", "alcunit": "TRK", "primary": 10,
  "secondary": 5, "dirblk": 10, "avgblk": 500, "recfm": "FB", "blksize": 400, "lrecl": 80, "dsntype": "LIBRARY"
}
```

## Example response

A sample response is shown in [Figure 315 on page 588](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 11 Oct 2017 11:14:13 GMT
```

*Figure 315. Example: Create PDSE data set.*

## Delete a sequential and partitioned data set

You can use this operation to delete sequential and partitioned data sets on a z/OS system.

### HTTP method and URI path

```
DELETE /zosmf/restfiles/ds/<dataset-name>
```

```
DELETE /zosmf/restfiles/ds/-(<volume>)/<dataset-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **<dataset-name>** is the name of a z/OS data set, that you are going to delete.
- **<volume>** is where the data set is resided, when the data set is uncataloged.

### Request Body

None.

## Standard headers

None.

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

None.

## Content type

The content type is application/json.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

If the request is successfully executed, status code 204 indicates success and no content is returned.

## Example request

In the following example, the DELETE method is used to delete a data set.

```
DELETE /zosmf/restfiles/ds/JIAHJ.REST.TEST.DATASET HTTP/1.1
```

### Example response

A sample response is shown in [Figure 316 on page 590](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:08:38 GMT
```

*Figure 316. Example: Delete a data set*

### Example request

The DELETE method is used to delete an uncataloged data set.

```
DELETE /zosmf/restfiles/ds/-(ZMF046)/JIAHJ.REST.TEST.DATASET2 HTTP/1.1
```

### Example response

A sample response is shown in [Delete uncataloged data set](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:10:22 GMT
```

*Figure 317. Example: Delete uncataloged data set.*

## Delete a partitioned data set member

You can use this operation to delete a member of a PDS or PDSE.

### HTTP method and URI path

```
DELETE /zosmf/restfiles/ds/<dataset-name>(<member-name>)
```

```
DELETE /zosmf/restfiles/ds/-(volume)/<dataset-name>(<member-name>)
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ds** indicates a data set request
- **<dataset-name>** is the name of a z/OS data set that contains a member you are going to delete.
- **<member-name>** is the name of the partitioned data set member, that you are going to delete.
- **<volume>** is where the data set resides, when the data set is uncataloged.

### Request Body

None.

### Standard headers

None.

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

None.

## Content type

The content type is application/json.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

If the request is successfully executed, status code 204 indicates success and no content is returned.

## Example request

In the following example, the DELETE method is used to delete a member of a cataloged partitioned data set.

---

```
DELETE zosmf/restfiles/ds/JIAHJ.REST.TEST.PDS(MEMBER01) HTTP/1.1
```

---

## Example response

A sample response is shown in [Delete a member of a cataloged partitioned data set](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Tue, 15 Sep 2015 10:36:14 GMT
```

*Figure 318. Example: Delete a member of a cataloged partitioned data set*

## Example request

The DELETE method is used to delete a member of an uncataloged partitioned data set.

```
DELETE zosmf/restfiles/ds/-(ZMF046)/JIAHJ.REST.TEST.PDS.UNCAT(MEMBER01) HTTP/1.1
```

## Example response

A sample response is shown in [Delete a member of an uncataloged partitioned data set](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Tue, 15 Sep 2015 11:37:12 GMT
```

*Figure 319. Example: Delete a member of an uncataloged partitioned data set*

## z/OS data set and member utilities

You can use the z/OS data set and member utilities to work with data sets and members. The available operations include: rename data set, rename member, copy data set, copy member, migrate data set, recall a migrated data set, and delete a backup version of a data set.

## HTTP method and URI path

```
PUT /zosmf/restfiles/ds/<to-data-set-name>
PUT /zosmf/restfiles/ds/<to-data-set-name>(<member-name>)
```

*Figure 320. 'rename' request*

```
PUT /zosmf/restfiles/ds/<to-data-set-name>
PUT /zosmf/restfiles/ds/<to-data-set-name>(<member-name>)
PUT /zosmf/restfiles/ds/-(<to-volser>)/<to-data-set-name>
PUT /zosmf/restfiles/ds/-(<to-volser>)/<to-data-set-name>(<member-name>)
```

*Figure 321. 'copy' request*

```
PUT /zosmf/restfiles/ds/<to-data-set-name>
```

*Figure 322. 'hmigrate', 'hrecall', or 'hdelete' request*

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface.
- **/ds** indicates a data set request.
- **-(<volser>)** represents a volume serial. For an uncataloged data set, include this parameter to identify the volume to be searched for data sets or members that match the specified *<data-set-name>* or *<member-name>*. The length of the volume serial cannot exceed 6 characters. You cannot use wildcard characters for this parameter. Indirect volume serials are not supported.
- **<to-data-set-name>** identifies the target data set name. This parameter is required and must consist of a fully qualified data set name. The length of the data set name that you specify on the request cannot exceed 44 characters.
- **<member-name>** identifies the target PDS or PDSE member name.

## Custom headers

The header Content-Type: application/json; charset={charset-name} must be specified, too.

### X-IBM-BPXK-AUTOCVT

This header is optional. Use it to indicate how file auto conversion is handled when using the copy operation to copy text mode data sets to POSIX files. If you omit this header, the system default is taken.

#### 'on' or 'all'

The target file is a candidate for automatic conversion if its TXTFLAG is tagged TEXT and the source data set is type TEXT.

#### 'off'

The target file is not a candidate for automatic conversion.

### X-IBM-Migrated-Recall

This header is optional; use it to specify how a migrated data set is handled. By default, a migrated data set is recalled synchronously. The following values can be specified, too:

#### wait

This is the default value. If the data set is migrated, wait for it to be recalled before processing the request.

#### nowait

If the data set is migrated, request it to be recalled, but do not wait.

#### error

If the data set is migrated, do not attempt to recall the data set.

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

### Request body

A JSON request document (content-type=application/json, character-encoding=UTF-8) must be supplied in one of the following forms:

Table 338. Request			
Function	Property	Description	Required
hmigrate	request	Indicates the function name.	Yes
	wait:true false	If true, the function waits for completion of the request. If false (default) the request is queued.	No
hrecall	request	Indicates the function name.	Yes
	wait:true false	If true, the function waits for completion of the request. If false (default) the request is queued.	No
hdelete	request	Indicates the function name.	Yes
	wait:true false	If true, the function waits for completion of the request. If false (default) the request is queued.	No
	purge:true false	If true, the function uses the PURGE=YES on ARCHDEL request. If false (default) the function uses the PURGE=NO on ARCHDEL request.	No
rename	request	Indicates the function name.	Yes
	from-dataset	The data set to rename.	Yes
		<b>dsn</b> The source data set name. This is required.	Yes
		<b>member</b> If renaming a member this is the old member name. This is not required.	No
	enq	enq for the "to" data set is only allowed for renaming members. Values can be SHRW or EXCLU.  SHRW is the default or PDS members, EXCLU otherwise.	No

Table 338. Request (continued)

Function	Property	Description	Required
copy	request	Indicates the function copy.	Yes
	from-file	The file to copy.	You must choose either from-file or from-dataset.
		<b>filename</b> The absolute source file name. This value is required.	
		<b>type</b> One of "binary   executable   text". Default is text. This is not required.	
	from-dataset	The data set to copy.	
		<b>dsn</b> The source data set. This is required.	
		<b>member</b> Used to specify a member; "*" means all members. This is not required.	
		<b>volser</b> Can be specified if dsn is not cataloged. This is not required.	
		<b>alias:true false</b> if true, aliases are copied along with main member;if false(default), alias relationships are not maintained. This is not required.	
	enq	Only applicable when from-dataset is specified. With from-file, an error is reported (see note below).This is the enqueue type for the "to" data set. Allowed values are: SHR, SHRW, EXCLU;SHRW is the default for PDS, EXCLU for sequential. The source data set is always enqueued via SHR.  <b>Note:</b> When from-file is specified, the target dsn is opened with DISP=OLD (EXCLU) with one exception: If the target is a PDS and the from-file/type is text, the target PDS is enqueued SHRW. This is not required.	No
	replace:true false	Applicable with from-dataset. When from-file specified,ignored unless from-file/type=text. If true, members in the target data set are replaced. If false(default), like named members are not copied and an error is returned.	No

**Note:** The "to" data set must be a PDS if from-dataset/member is '\*' or a <member-name> is specified on the URL. When from-dataset/member is a single member name and the member name is NOT specified on the URL, the 'to' data set is expected to be sequential.

### Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

Users of the IEBCOPY data set utility might be accustomed to freeing unused space during a data set copy operation for a partitioned data set (PDS). This function is similar to ISPF option 3.1 compress, in which the unused space occupied by deleted or updated members is removed when a PDS is copied. The z/OS data set and member utilities REST API does not offer an equivalent option for a data set copy operation.

For other usage considerations, see [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example

See [Figure 323 on page 596](#) for an example of renaming a data set.

```
Request:
PUT https://zosmf1.yourco.com/zosmf/restfiles/ds/MY.NEW.DSN HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"request": "rename", "from-dataset": {"dsn": "MY.OLD.DSN"}}
```

*Figure 323. Example: Rename MY.OLD.DSN to MY.NEW.DSN*

## Example

See [Figure 324 on page 596](#) for an example of copying a PDS member.

```
Request:
PUT https://zosmf1.yourco.com/zosmf/restfiles/ds/MY.NEW.DSN(MYMEM2) HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"request": "copy", "from-dataset": {"dsn": "MY.OLD.DSN", "member": "MYMEM1"}, "replace": true }
```

*Figure 324. Example: copy member MYMEM1 from MY.OLD.DSN to MY.NEW.DSN(MYMEM2)*

## Access Method Services Interface

You can use the Access Method Service (IDCAMS) to provide a REST/JSON interface to IDCAMS. You can use this operation to create a new zFS filesystem.

### HTTP method and URI path

```
PUT /zosmf/restfiles/ams
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/ams** indicates a request for Access Method Services (IDCAMS) services.

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Input Content

Input content in a json document:

Table 339. Input Content		
Property	Description	Required
input	one or more input lines <= 255 in length	Yes
JSONversion:1	JSON Version	No

**Note:** The size of all input lines plus the number of input lines must be <= 8K.

## Response Body

If the request is successfully executed, will return 200 status code (IDCAMS RC<=4). In all cases an application/json document will be returned:

Table 340. Response		
Property	Description	Required
rc	Return code from IDCAMS.	Yes
output	One or more input lines <= 255 in length.	Yes
JSONversion	JSON Version.	No

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling”](#) on page 568.

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document”](#) on page 640.

## Example

Refer to [Figure 325](#) on page 598 for an example of IDCAMS Access Methods Services.

```
request:
PUT https://zosmf1.yourco.com/zosmf/restfiles/ams HTTP/1.1
Content-Type: application/json
Content-Length: nn
{
  "input": [
    "DEFINE CLUSTER(NAME (EXAMPL1.KSDS) VOLUMES(VSER05)) - ",
    "DATA (KILOBYTES (50 5))"],
    "JSONversion":1
  ]
}
```

*Figure 325. IDCAMS Access Methods Services*

## List the files and directories of a UNIX file path

You can use this operation to list the files and directories in a UNIX file path on a z/OS system.

### HTTP method and URI path

```
GET /zosmf/restfiles/fs?path=<filepath-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface.
- **/fs** identifies a UNIX file path request.
- **?path=<filepath-name>** is a query parameter that specifies the directory that contains the files and directories to be listed.

### Standard headers

None.

### Custom headers

You can include the following custom HTTP headers with this request:

#### X-IBM-Max-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

## X-IBM-Lstat

If the value of this header is "true", a `lstat()` is performed on the path rather than `stat()` and a list containing one item is returned with the `lstat` results. For more information about `lstat()` and `stat()`, see [lstat\(\), lstat64\(\) — Get status of file or symbolic link](#) and [stat\(\), stat64\(\) — Get file information in z/OS XL C/C++ Runtime Library Reference](#).

## X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

## X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

## X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

Specify the following query parameter on this request:

### path

This parameter identifies the UNIX directory that contains the files and directories to be listed. This parameter is required and can consist of one or more directories in the hierarchical file system structure, or a fully qualified file name. A fully qualified file name, which consists of the name of each directory in the path to a file plus the file name itself, can be up to 1023 bytes long. You cannot use wildcard characters for this parameter.

The following list contains sample file path names:

```
/
/bin
/usr/lib/libSM.a
```

**Filter parameters:** To further qualify your request, you can include one or more *filter* parameters; see Table 341 on page 599. The filter parameters are used if the path parameter refers to a UNIX directory. The filter parameters cause z/OSMF to search the directory tree starting with 'path' and return the results that are filtered by the supplied parameters. If more than one of the filter parameters are specified, a logical AND is performed. That is, all of the supplied filter parameters must be true for an entry to be returned.

Table 341. Filter parameters for a "List UNIX Files and Directories" request	
Parameter	Function
group	Select entries that have a group owner of name. If name is an integer value, select entries that have a group owner of GID.

Table 341. Filter parameters for a "List UNIX Files and Directories" request (continued)

Parameter	Function
mtime	Select entries that were modified with a value of number days ago. If a number is given without a minus sign or plus sign, files that are modified exactly number of days ago are selected. If number is preceded with a plus sign, files modified more than number of days ago are selected. If number is preceded with a minus sign, files modified less than number of days ago are selected.
name	Select entries that match pattern according to the rules of fnmatch(). The supplied pattern is matched against the absolute path of the entry, with behavior similar to the find -name option.
size	Select files that are number long. If number does not include a suffix, the number of the file size in bytes. If number includes a suffix of K, the number size is in 1024-byte blocks; if M, the number size is 1048576-byte blocks, and if G the number size is 1073741824-byte blocks. If number is specified without a plus sign or minus sign, files of the size matching number exactly are selected. If number is preceded with a plus sign, files larger than number are selected. If number is preceded with a minus sign, files smaller than number are selected.  If this parameter is specified, only regular files are checked; see the type parameter.
perm	Select entries whose permissions match the value octal_mask. If octal_mask is prefixed by a minus sign, entries that have all of the bits set are present in octal_mask. Otherwise, only select entries whose permission bits match octal_mask exactly.  Only the rightmost 12 bits (07777) of octal_mask are used.
type	Select entries based on type, as follows:  <ul style="list-style-type: none"> <li>* c Character special file</li> <li>* d Directory</li> <li>* f File</li> <li>* l Symbolic link</li> <li>* p FIFO (named pipe)</li> <li>* s Socket</li> </ul> <p>If this parameter is specified with the size parameter, it must be set to 'f'. Sizes that are associated with all other types are unspecified.</p>
user	Select entries that have a user owner of name. If name is an integer value, select entries that have a user owner of UID.

**Tree traversal parameters:** If you include one or more of the above filter parameters in the request, you can add tree traversal parameters to further control the behavior of the directory tree search. See [Table 342 on page 601](#). If no filter parameters are supplied, any tree traversal parameters are ignored.

Table 342. Tree traversal parameters for a "List UNIX Files and Directories" request

Parameter	Function
depth	<p>The default value for this parameter is 0, which means that all subdirectories under path are listed, regardless of depth. When depth is greater than 1, subdirectories up to the specified depth are listed. When depth is 1, only the files in the path are listed.</p> <p>The name field in the returned JSON document contains the path of the entry, relative to the path query parameter.</p>
limit	<p>The limit parameter specifies the maximum number of items to return.</p> <p><b>Note:</b> This option is the same as X-IBM-Max-Items, but is added as query parameter for convenience. If both limit and X-IBM-Max-Items are supplied, limit is used.</p>
filesys	<p>The default value for this parameter is same. When set to same, only the sub directories on the same file system as the path parameter are listed. If the value is all, all sub-directories under path are listed.</p>
symlinks	<p>The default value for this parameter is follow. When using the follow value, symbolic links are followed. If the value is set to report, symbolic links are returned, but not followed.</p>

## Required authorizations

See ["Required authorizations" on page 567](#).

## Usage considerations

See ["Usage considerations for the z/OSMF REST services" on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see ["Error handling" on page 568](#).

For a successful request, the HTTP response includes an array of matching of UNIX files and directories, each as a JSON list document. For the contents, see ["File list document" on page 638](#).

For errors, the HTTP response includes error information as a JSON error report document. See ["Error report document" on page 640](#).

## Example request

In the following example, the GET method is used to list the files and directories in the UNIX path /usr:

```
GET /zosmf/restfiles/fs?path=/usr HTTP/1.1
```

## Example response

A sample response is shown in [Figure 326 on page 602](#).

#### Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 896
Content-Language: en-US
Date: Tue, 24 Nov 2015 06:12:16 GMT
```

#### Response Body

```
{
  "items": [
    {
      "name": ".", "mode": "drwxrwxrwx", "size": 8192, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2015-11-24T02:12:04"
    },
    {
      "name": "..", "mode": "drwxr-xr-x", "size": 8192, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2015-09-15T02:38:29"
    },
    {
      "name": ".profile", "mode": "-rwxrwxrwx", "size": 849, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2013-02-13T12:08:29"
    },
    {
      "name": ".sh_history", "mode": "-rw-----", "size": 4662, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2013-06-06T18:09:28"
    },
    {
      "name": "myFile.txt", "mode": "-rw-r--r-", "size": 20, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2015-11-24T02:12:04"
    },
    {
      "name": "profile.add", "mode": "-rwxrwxrwx", "size": 888, "uid": 0, "user": "WSADMIN", "gid": 1,
      "group": "OMVSGRP", "mtime": "2013-05-07T11:23:08"
    }
  ],
  "returnedRows": 6, "totalRows": 6, "JSONversion": 1
}
```

Figure 326. Example: Returned list of UNIX files and directories in path /usr

### Example request

In the following example, the GET method is used to list a UNIX file.

```
GET /zosmf/restfiles/fs?path=/u/ibmuser/myFile.txt HTTP/1.1
```

### Example response

A sample response is shown in Example: Returned list of UNIX files.

#### Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 210
Content-Language: en-US
Date: Tue, 24 Nov 2015 09:16:49 GMT
```

#### Response Body

```
{
  "items": [
    {
      "name": "/u/ibmuser/myFile.txt", "mode": "-rw-r--r-", "size": 20, "uid": 0, "user": "WSADMIN",
      "gid": 1, "group": "OMVSGRP", "mtime": "2015-11-24T02:12:04"
    }
  ],
  "totalRows": 1, "returnedRows": 1, "JSONversion": 1
}
```

Figure 327. Example: Returned list of UNIX files

### Example response

A sample response is shown in [Figure 328 on page 603](#).

## Response

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: application/json; charset=UTF-8
Content-Length: 210
Content-Language: en-US
Date: Tue, 10 Oct 2017 09:16:49 GMT
GET /zosmf/restfiles/fs?path=/usr/include&name=f*.h HTTP/1.1
```

## Response Body

```
{
  "items": [
    {
      "name": "sys/file.h",
      "mode": "-rw-r--r--",
      "size": 2054,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "metal/float.h",
      "mode": "-rw-r--r--",
      "size": 4954,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "ftpcapi.h",
      "mode": "-rw-r--r--",
      "size": 37,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fcntl.h",
      "mode": "-rw-r--r--",
      "size": 7928,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "features.h",
      "mode": "-rw-r--r--",
      "size": 54149,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fenv.h",
      "mode": "-rw-r--r--",
      "size": 5125,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "float.h",
      "mode": "-rw-r--r--",
      "size": 29480,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fntmsg.h",
      "mode": "-rw-r--r--",
      "size": 7668,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fnmatch.h",
      "mode": "-rw-r--r--",
      "size": 3438,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "fp.h",
      "mode": "-rw-r--r--",
      "size": 2293,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    },
    {
      "name": "ftw.h",
      "mode": "-rw-r--r--",
      "size": 7906,
      "uid": 0,
      "user": "BPXROOT",
      "gid": 1,
      "group": "SYS1",
      "mtime": "2017-04-03T01:48:03"
    }
  ],
  "returnedRows": 11,
  "totalRows": 11,
  "JSONversion": 1
}
```

Figure 328. Example: List UNIX files with more query parameters.

## Retrieve the contents of a z/OS UNIX file

You can use this operation to retrieve the contents of a z/OS UNIX System Services file.

### HTTP method and URI path

```
GET /zosmf/restfiles/fs/<filepath-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file request
- **<filepath-name>** identifies the UNIX file to be read. This parameter is required and must consist of a fully qualified path and file name.

### Optional Query Parameters

#### search=<string>

The file is searched for the first line that contains the string, without respect to case (by default).

Optionally, insensitive=false may be specified for case sensitive matching.

This parameter may not be specified with the research= parameter.

#### research=<regular-expression>

The file is searched for the first line that matches the given extended regular expression.

This parameter may not be specified with the search= parameter.

Implementation note: the regcomp() C Library function with the REG\_EXTENDED flag is used.

**insensitive=true|false**

The default is 'true'. When 'true', searches (search and research) are case insensitive. For case sensitive searches, specify 'false'.

**maxreturnsize=<integer>**

This parameter may be specified only with search= or research=.

The value given is the maximum number of lines to return.

The default, if not specified, is 100.

For the search and research queries, records are returned starting with the first matching line. The maximum line length supported for text searches is 64K. The X-IBM-Record-Range request header may be used to specify the range of lines to be searched, but it will not restrict the number of lines returned (see maxreturnsize).

If no X-IBM-Record-Range request header is present, the search will begin with the first line. In all cases, an X-IBM-Record-Range=p,q response header will be returned where p is the first matching line and q is the number of lines returned.

If no matching lines are found, the response header X-IBM-Record-Range=0,0 will be returned.

The parameter may not be used if a request header X-IBM-Data-Type specifies any option except 'text'.

**Standard headers**

You can include the following standard HTTP header with this request:

**If-None-Match**

This header is optional; use it to specify the ETag token to be used to correlate this request with a previous request. If the data on the z/OS host has not changed since the ETag token was generated, z/OSMF returns a status of HTTP 304 Not Modified.

For an initial request to the resource, you can omit this header.

**Range**

This header is optional; use this header to retrieve a range of bytes from a file. This header is supported only when X-IBM-Data-Type=binary. Specify this range using the following:  
bytes=first-byte-pos "-" last-byte-pos

The first-byte-pos value is the byte-offset of the first byte in a range. The last-byte-pos value is the byte-offset of the last byte in the range. The byte positions specified are inclusive. Byte offsets start at zero. When last-byte-pos is not specified or is zero, the range extends to the end of the file. When the first-byte-pos is not specified, a tail range is returned. Comma separated ranges are not supported. For an initial request to the resource, you can omit this header.

Usage notes: If the range cannot be satisfied, i.e. zero bytes are returned, then a status code of 416 is set.

Examples (assuming a file with 10000 bytes):

bytes=0-499 retrieves the first 500 bytes

bytes=500-999 retrieves the second 500 bytes

bytes=500- retrieves the final 9500 bytes

bytes=-500 retrieves the final 500 bytes

**X-IBM-Record-Range**

Use this header to retrieve a range of records (lines delimited by '\n') from a file. You can specify this range using either of the following formats:

**SSS-EEE**

Where SSS identifies the start record and EEE identifies the end record to be retrieved. Both values are relative offsets (0-based). When EEE is set to 0, records through the end of the file are retrieved. When SSS is omitted (i.e. -EEE), the final EEE records of the file are retrieved.

**SSS,NNN**

Where SSS identifies the start record and NNN identifies the number of records to be retrieved.

Usage notes: If X-IBM-Record-Range is specified with Range an error is reported. If zero bytes are returned due to the range specified, status code 500 is returned.

## Custom headers

You can include the following custom HTTP header with this request:

### X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the returned data, as follows:

- When set to `text`, data conversion is performed. The data transfer process converts each record from EBCDIC to the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. A newline (NL) character from the response charset is inserted between logical records. For data sets with fixed-length records, trailing blanks are removed.

A value `"text;fileEncoding=<codepage>"` can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

**Note:** An alternate file encoding cannot be specified with the "research" query parameter.

- When set to `binary`, no data conversion is performed. The data transfer process returns each line of data as-is, without translation.

If you omit this header, the default is `text`; the response is converted.

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

None.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. Status code 304 indicates an unchanged file when a conditional get is performed (such as when using the **If-None-Match** header with an ETag from a previous response). Status code 206 indicates that a part of the file has been returned as a result of a Range header on the request. Accompanying this status code will be a Content-Range header in the form sss-eee/nnnnnn where sss-eee is the byte range that was actually returned and nnnnnn is the length of the file. This status is returned only for the standard range header, not the custom X-IBM-Record-Range header. Status code 416 indicates that zero bytes have been returned due to the Range header on the request. This status is returned only for the standard range header, not the custom X-IBM-Record-Range header. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example request

In the following example, the GET method is used to retrieve the contents of the file inetd.conf in the /etc directory.

```
GET /zosmf/restfiles/fs/etc/inetd.conf HTTP/1.1
```

## Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 200)
- ETag that you can use on subsequent requests to test for changes to the resource
- Content-Length response header that specifies the amount of data that was returned (in bytes)
- A response body that contains the resource in plain text.

```
200 OK
X-Powered-By: Servlet/3.0
Content-Type: text/plain; charset=UTF-8
Content-Length: 2673
Etag: AEA05EC01C7922ADD5103EBD95FFCC58
Content-Language: en-US
Date: Wed, 25 Nov 2015 03:07:10 GMT
```

A sample response body is shown in [Figure 329 on page 607](#).

```

#####                                00000100
# Used to replace /etc/inetd.conf on 2nd level z/OS system          00000101
# so we can telnet/rlogin directly to OMVS using PuTTY consoles.    00000102
#####                                00000110
# Internet server configuration database                             00000200
#                                                                    00000300
# (C) COPYRIGHT International Business Machines Corp. 1985, 2001   00000400
# All Rights Reserved                                              00000500
# Licensed Materials - Property of IBM                             00000600
#                                                                    00000700
# US Government Users Restricted Rights - Use, duplication or      00000800
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp. 00000900
#                                                                    00001000
# /etc/inetd.conf                                                  00001100
#                                                                    00001200
#           Internet server configuration database                  00001300
#                                                                    00001400
# $01=PYQ0049, HOT7705, 010130, PDJP: Correct paths and remove    00001500
# unsupported services (FIN APAR OW45915)                           00001600
#                                                                    00001700
# Services can be added and deleted by deleting or inserting a     00001800
# comment character (ie. #) at the beginning of a line             00001900
#                                                                    00002000
#=====00002100
# service | socket | protocol | wait/ | user | server | server program 00002200
# name    | type   |         | nowait|      | program | arguments 00002300
#=====00002400
#                                                                    00002500
#                                                                    00002600
otelnets  stream tcp nowait OMVSKERN /usr/sbin/otelnetsd otelnetsd -l
#sh       stream tcp nowait OMVSKERN /usr/sbin/sshd sshd -i
shell     stream tcp nowait OMVSKERN /usr/sbin/orshd orshd -LV      00002700
login     stream tcp nowait OMVSKERN /usr/sbin/rlogind rlogind -m    00002800
exec      stream tcp nowait OMVSKERN /usr/sbin/orexecd orexecd -LV    00002900

```

Figure 329. Example: Response body for a GET request to the UNIX file `/etc/inetd.conf`

## Write data to a z/OS UNIX file

You can use this operation to write data to an existing z/OS UNIX System Services file.

### HTTP method and URI path

```
PUT /zosmf/restfiles/fs/<filepath-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file request
- **<filepath-name>** identifies the UNIX file to which to write. This parameter is required and must consist of a fully qualified path and file name. If the file already exists, it can be overwritten. If the file does not exist, it can be created.

### Request body

The data to write to the target UNIX file. The data is interpreted according to the content-type as one of binary, text, record or 'diff -e' format according a combination of the "Content-Type" and the value of the X-IBM-Data-Type custom header, if present.

### Standard headers

You can include the following standard HTTP header with this request:

#### If-Match

This header is optional; use it to specify the ETag to be used for correlating this request with a previous request on the same UNIX file. If the file has not changed since the ETag was generated, the request is processed. Otherwise, if the file has been modified, the request is failed with status code HTTP 412.

If you omit this header, the data is always written, regardless of whether the file is changed.

## Custom headers

You can include the following custom HTTP header with this request:

### X-IBM-Data-Type

This header is optional; use it to indicate whether data conversion is to be performed on the data to be written, as follows:

#### text

When set to `text`, data conversion is performed. The data transfer process converts each byte from the charset specified on the "Content-Type" header of the request. If no charset is specified, the default is ISO8859-1. Line Feed characters are left intact. This is the default value

A value `"text;fileEncoding=<codepage>"` can be used to select an alternate EBCDIC code page. The default code page is IBM-1047.

A value `text;CrLf=true` can be used to control whether each input text line is terminated with a carriage return line feed (CRLF), rather than a line feed (LF), which is the default.

**Note:** When set to 'text' and "Content-Type" is "application/x-ibm-diff-e", the input consists of commands in the same format as produced by the z/OS UNIX 'diff -e' command. These commands are used to add, replace and delete lines in the target data set. The following commands are supported:

a

c

d

s/.//

opt : g|<n>, g means global

n means search and replace <n> times

Each command may be optionally preceded by a line or line range, as allowed by the z/OS UNIX 'ed' command. If an error is detected while processing a command, status code 500 is returned with an exception.

#### binary

When set to `binary`, no conversion is performed.

If you omit this header, the default is `text`; the data is converted.

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

### **Query parameters**

None.

### **Required authorizations**

See [“Required authorizations” on page 567](#).

### **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### **Expected response**

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success if an existing file was updated. Status code 201 indicates a success if a new file was created. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

### **Example request**

In the following example, the PUT method is used to replace the UNIX file `/etc/inetd.conf`.

```
PUT /zosmf/restfiles/fs/etc/inetd.conf HTTP/1.1
If-Match: F4A5A479E78AFD4CFF7DF13937AB82AE
Content-Type: text/plain; charset=UTF-8
```

A sample request body is shown in [Figure 330 on page 610](#).

```

####
# Internet server configuration database
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 2001
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# /etc/inetd.conf
#
#           Internet server configuration database
#
# $01=PYQ0049, HOT7705, 010130, PDJP: Correct paths and remove
# unsupported services (FIN APAR OW45915
#
# Services can be added and deleted by deleting or inserting a
# comment character (ie. #) at the beginning of a line
#
#=====
# service | socket | protocol | wait/ | user | server | server program
# name    | type   |          | nowait|      | program | arguments
#=====
#
# Following line uncommented by USSSETUP job: 2013/04/24 15:04:00
otelnet  stream tcp nowait IBMUSER  /usr/sbin/otelnetd otelnetd -l
# Following line uncommented by USSSETUP job: 2013/04/24 15:04:00
shell    stream tcp nowait IBMUSER  /usr/sbin/orshd orshd -LV
# Following line updated by USSSETUP job: 2013/04/24 15:04:00
login    stream tcp nowait IBMUSER  /usr/sbin/rlogind rlogind -m
# Following line added by USSSETUP job: 2013/04/24 15:04:00
ssh      stream tcp nowait IBMUSER  /usr/sbin/sshd sshd -i
#exec    stream tcp nowait OMVSKERN /usr/sbin/orexecd orexecd -LV
# All users should use this configuration file

```

Figure 330. Example: Request body for a PUT request to the UNIX file /etc/inetd.conf

### Example response

For a successful request, the HTTP response contains the following:

- Status code indicating that the request completed (status code 204)
- ETag that you can use on subsequent requests to test for changes to the UNIX file

```

204 No Content
Etag: S8WNSD09SNSNE09B
Content-Type: application/json; charset=UTF-8
Content-Language: en-US
Date: Fri, 07 Nov 2014 02:31:39 GMT

```

## Create a UNIX file or directory

You can use this operation to create a UNIX file or directory.

### HTTP method and URI path

```
POST /zosmf/restfiles/fs/<file-path>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file or directory.

- **<file-path>** is the name of the file or directory you are going to create.

## Request Body

The request body to create a UNIX file or directory is shown in [Request body to create a UNIX file or directory](#).

Table 343. Request body to create a UNIX file or directory		
Field	Type	Description
type	String	The request type. This field supports the values: directory or dir to create a directory. The value: file is supported to create a file.
mode	String	Specifies the file or directory permission bits to be used in creating the file or directory. The characters used to describe permissions are:  r: Permission to read the file w: Permission to write on the file x: Permission to execute the file -: No permission  An example would be: rwxrwxrwx  The nine characters are in three groups of three; they describe the permissions on the file or directory. The first group of 3 describes owner permissions; the second describes group permissions; the third describes other (or world) permissions.

## Standard headers

None.

## Custom headers

You can include the following custom HTTP headers with this request:

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

None.

## **Content type**

The content type is application/json.

## **Required authorizations**

See [“Required authorizations” on page 567](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## **Expected response**

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For a successful create request, 201 Created is returned.

## **Example request**

In the following example, the POST method is used to create a UNIX file.

```
POST /zosmf/restfiles/fs/u/jiahj/text.txt HTTP/1.1
```

Request body:

```
{"type": "file", "mode": "RWXRW-RW-"}
```

## **Example response**

A sample response is shown in [Create a UNIX file](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 30 Sep 2015 11:46:21 GMT
```

*Figure 331. Example: Create a UNIX file*

## Example request

The POST method is used to create a UNIX directory.

```
POST /zosmf/restfiles/fs/u/jiahj/testDir HTTP/1.1
```

Request body

```
{"type":"directory","mode":"rwxr-xrwx"}
```

## Example response

A sample response is shown in [Create a UNIX directory](#).

```
201 Created
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 30 Sep 2015 10:50:21 GMT
```

*Figure 332. Example: Create a UNIX directory*

## Delete a UNIX file or directory

You can use this operation to delete a UNIX file or directory.

### HTTP method and URI path

```
DELETE /zosmf/restfiles/fs/<file-pathname>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX file or directory.
- **<file-pathname>** is the name of the file or directory you are going to delete.

### Request Body

None.

### Standard headers

None.

### Custom headers

**X-IBM-Option:** An optional parameter for deleting a directory. If it is not specified, only the empty directory can be deleted. If it is specified as recursive, it means all the files and sub-directories will be deleted.

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

None.

## **Content type**

The content type is application/json.

## **Required authorizations**

See [“Required authorizations” on page 567](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## **Expected response**

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

If the request is successfully executed, status code 204 indicates success and no content returned is returned.

## **Example request**

In the following example, the DELETE method is used to delete a UNIX file.

```
DELETE /zosmf/restfiles/fs/u/jiahj/text.txt HTTP/1.1
```

## **Example response**

A sample response is shown in [Figure 333 on page 614](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:10:22 GMT
```

*Figure 333. Example: Delete a UNIX file*

## Example request

The DELETE method is used to delete a UNIX directory.

```
DELETE /zosmf/restfiles/fs/u/jiahj/testDir HTTP/1.1
```

## Example response

A sample response is shown in [Delete a UNIX directory](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Wed, 16 Sep 2015 12:15:22 GMT
```

*Figure 334. Example: Delete a UNIX directory*

## z/OS UNIX file utilities

You can use the z/OS UNIX file utilities to operate on a UNIX System Services file or directory. Operations include: chmod, chown, chtag, copy, extattr, getfacl, move, and setfacl.

## HTTP method and URI path

```
PUT /zosmf/restfiles/fs/<file-path-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/fs** indicates a UNIX System Services file system request
- **<file-path-name>** identifies the UNIX file or directory to be the target of the operation. This is required and must consist of a fully qualified path and file or directory name.

## Custom headers

### X-IBM-BPXK-AUTOCVT

This header is optional. Use it to indicate how file auto conversion is handled when using the copy operation to copy text mode data sets to POSIX files, if you omit this header, the system default is taken.

#### 'on' or 'all'

The target file is a candidate for automatic conversion if its TXTFLAG is tagged TEXT and the source data set is type TEXT.

#### 'off'

The target file is not a candidate for automatic conversion

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-

Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance”](#) on page 398 was issued in a previous request.

### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

The header, Content-Type: application/json; charset={charset-name}, must be specified as well.

## Request body

A JSON request document (content-type=application/json, character-encoding=UTF-8) must be supplied in one of the following forms:

Table 344. JSON request document			
Function	Property	Description	Required
chmod	request	Indicates the function chmod.	Yes
	mode	The mode value, which is specified as the POSIX symbolic form or octal value (as a JSON string).	Yes
	links:"follow suppress"	The default is 'follow' encountered links. This applies a mode change to the file or directory pointed to by any encountered links. 'suppress' is a mode change for the file or directory pointed to by any encountered symbolic links.	No
	recursive:true false	The default is false. When 'true', the file mode bits of the directory and all files in the file hierarchy below it are changed (chmod -R).	No
chown	request	Indicates the function chown.	Yes
	owner	The user ID or UID (as a JSON string).	Yes
	group	The group ID or GID (as a JSON string).	No
	links:"follow change"	The default is 'follow'. 'change' does not follow the link, but instead changes the link itself (chown -h).	No
	recursive:true false	The default is false. When 'true', changes all the files and subdirectories in that directory to belong to the specified owner (and group, if :group is specified).chown -R)	No

Table 344. JSON request document (continued)

Function	Property	Description	Required
chtag	request	Indicates function chtag.	Yes
	action:"set remove list"	The file tag action. If "set", the file is tagged as specified in the "type" keyword. If "remove", any existing file tag is removed. If "list", the existing tag information will be returned in a JSON response document. See <a href="#">"list action."</a>	Yes
	type:"binary mixed text"	The default is "mixed" This option can be specified only when the action is "set".	No
	codeset	Specifies the coded character set in which text data is encoded, such as ASCII or EBCDIC. For example, the code set for ASCII is ISO8859-1; the code set for EBCDIC is IBM-1047.	No
	links:"change suppress"	The default is 'change' encountered links, applying a tag action to the file or directory pointed to by any encountered links. 'suppress' a tag action for the file or directory pointed to by any encountered	No
	recursive:true false	The default is false. When 'true', tags all the files and subdirectories in that directory (chtag -R).	No

**Note:** If the 'list' action is specified, a response JSON document is returned listing the current tag information, For example:

```
{ "stdout": [ "m ISO8859-1  T=off /tmp/file" ] }
```

The -q and -v options are not supported.

Table 344. JSON request document (continued)

Function	Property	Description	Required
copy	request	Indicates the function copy.	Yes
	from:file-or-directory	The file or directory to copy.	You can use either from:file-or-directory or from-dataset.
	from-dataset	<b>dsn</b> The fully qualified data set name. This is required.	
		<b>member</b> The data set member to copy. This is not required.	
		<b>type</b> One of "binary   executable   text". If not specified, the format of the data set is checked to try to determine the type. This is not required.	
	overwrite:true false	The default is true. May not be specified with 'from-dataset'.	No
	recursive:true false	The default is false. When 'true', copies all the files and subdirectories that are specified by source into a directory (cp -R). May not be specified with 'from-dataset'.	No
	links:"none src all"	The default is none. When 'src', follows symbolic links that are specified as source file or directory (cp -H). When 'all', follows symbolic links specified as source file/directory and those encountered in the tree traverse (cp -L). Cannot be specified with 'from-dataset'.	No
	preserve: "none modtime all"	The default is none, sets the modification time of the destination file to the present. When 'modtime', sets the modification and access time of each destination file to that of the corresponding source file. (cp -m). When 'all', preserves the modification and access times as well as the file mode, file format, owner, and group owner (cp -p). May not be specified with 'from-dataset'.	No
<b>Note:</b> When from-dataset/type == text, and the header X-IBM-BPXK-AUTOCVT == ON   ALL, the cp "-O u" switch is supplied to allow automatic conversion. If the from-dataset/type attribute is not specified, no "-O u" switch is applied and automatic conversion will not be available.			
extattr	request	Indicate the function extattr.	Yes
	set:"attrs"	One or more of the following: alps.	No
	reset:"attrs"	One or more of the following: alps.	No
<b>Note:</b> If neither set or reset are provided, a response JSON document is returned listing the attributes, For example:			
<pre> {   "stdout":   ["/etc/inetd.conf",   "APF authorized = NO",   "Program controlled = NO",   "Shared address space = YES",   "Shared library=NO"] } </pre>			
The -F option is not supported.			

Table 344. JSON request document (continued)

Function	Property	Description	Required
getfacl	request	Indicates the function getfacl.	Yes
	type:"access dir file"	The default is 'access', displays the access ACL entries for a file or directory (getfacl -a). 'dir' displays the directory default ACL entries (getfacl -d). If the target is not a directory, a warning is issued.	No
	user	The user ID or UID (as a JSON string), displays only the ACL entries for the specified types of access control lists (getfacl -a, -d, -f) which affects the specified user's access (getfacl -e user).	No
	use-commas:true false	The default is 'false'. When true, displays each ACL entry, using commas to separate the ACL entries instead of newlines.	No
	suppress-header:true false	The default is 'false'. When true, the comment header (the first three lines of each file's output) is not to be displayed (getfacl -m).	No
	suppress-baseacl:true false	The default is 'false'. When true, displays only the extended ACL entries. Does not display the base ACL entries (getfacl -o).	No
<b>Note:</b> On completion of this request, a response JSON document is returned, For example: <pre>{ "stdout": [ "#file: /etc/inetd.conf", "#owner: CFZSRV", "#group: SYS1", "user::rwx", "group::rwx", "other::rwx" ] }</pre>			
move	request	Indicates the function move.	Yes
	from	The file or directory to be moved.	Yes
	overwrite:true false	The default is true. May not be specified with 'from-dataset'.	No

Table 344. JSON request document (continued)

Function	Property	Description	Required
setfacl	request	Indicates the function setfacl.	Yes
	abort:true false	The default is false. When true, aborts processing if an error or warning occurs. See the setfacl command documentation for complete documentation on the errors and warnings (setfacl -a).	No
	links:"follow suppress"	The default is 'follow'. 'suppress' does not follow symbolic links. Because ACLs are not associated with symbolic links, nothing happens if a symbolic link is encountered (setfacl -h).  <b>Note:</b> At least one of the following four keywords must be specified. 'modify' and 'delete' may both be specified, but not with 'delete-type' and 'set'.	No
	delete-type	Delete all extended ACL entries by type (setfacl -D type):  <b>access</b> Access ACL  <b>dir</b> Directory default ACL  <b>file</b> File default ACL  <b>every</b> Every extended ACL for all ACL types that are applicable for the current path name.  <b>Note:</b> The 'delete-type' keyword cannot be specified with 'set', 'modify' or 'delete'.	No
	set	sets (replaces) all ACLs with 'entries'. 'entries' represents a string of ACL entries. Refer to the setfacl command reference for the string format (setfacl -s entries).  <b>Note:</b> The 'set' keyword cannot be specified with 'delete-type', 'modify' or 'delete'.	No
	modify	Modifies the ACL entries. 'entries' represents a string of ACL entries. Refer to the setfacl command reference for the string format. If an ACL entry does not exist for a user or group that is specified in 'entries', it is created. If an ACL entry exists for a user or group that was specified in 'entries', it is replaced.  <b>Note:</b> The 'modify' keyword cannot be specified with 'delete-type' or 'set'.	No
	delete	Deletes the extended ACL entries that are specified by 'entries'. 'entries' is a string of ACL entries. Refer to the setfacl command reference for the string format. If an ACL entry does not exist for the user or group specified, no error is issued.  <b>Note:</b> The 'delete' keyword cannot be specified with 'delete-type' or 'set'.	No

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example

Refer to [Figure 335 on page 621](#) for an example of renaming a UNIX file.

```
Example request for renaming /etc/inetd.conf to /etc/inetd.conf.bak:
PUT https://zosmf1.yourco.com/zosmf/restfiles/fs/etc/inetd.conf.bak HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"request": "move", "from": "/etc/inetd.conf"}
```

*Figure 335. Example: Rename a UNIX file*

## List z/OS UNIX Filesystems

You can use the list z/OS UNIX filesystems operation to list all mounted filesystems, or the specific filesystem mounted at a given path, or the filesystem with a given Filesystem name.

### HTTP method and URI path

```
GET /zosmf/restfiles/mfs/
GET /zosmf/restfiles/mfs/?path=file-path-name
GET /zosmf/restfiles/mfs/?fsname=file-system-name
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/mfs** indicates a Unix System Services filesystem(s) request.

A trailing "/" may optionally be specified.

## Query parameters

### path

This parameter identifies the UNIX directory that contains the files and directories to be listed. This parameter may not be specified if the 'fsname' parameter is specified. It can consist a directory or fully qualified path name in the UNIX file system structure. A fully qualified file name can be up to 1023 bytes long. You cannot use wildcard characters for this parameter.

### fsname

This parameter identifies the fully qualified filesystem name to be listed. For zFS filesystems, this is the data set name of the aggregate. This parameter may not be specified if the 'path' parameter is specified.

## Custom headers

### X-IBM-MAX-Items

This header value specifies the maximum number of items to return. To request that all items be returned, set this header to 0. If you omit this header, or specify an incorrect value, up to 1000 items are returned by default.

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. Status code 404 indicates that the specified filesystem was not found. A JSON response document with no filesystem items will be returned. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example

Refer to [Figure 336 on page 623](#) for an example of renaming a data set and PDS member.

```
JSON response document:
Request:
GET https://zosmf1.yourco.com/zosmf/restfiles/mfs/?path=/usr/local HTTP/1.1
GET https://zosmf1.yourco.com/zosmf/restfiles/mfs/?fsname=OMVS.USR.LOCAL.ZFS HTTP/1.1
```

JSON response document:

```
{
  "items": [
    {
      "name": "OMVS.USR.LOCAL.ZFS",
      "mountpoint": "/usr/local",
      "fsname": "ZFS",
      "status": "active",
      "mode": ["acl", "synchonly"],
      "dev": 52,
      "fstype": 1,
      "bsize": 1024,
      "bavail": 5615,
      "blocks": 9600,
      "sysname": "SYSNAME1",
      "readibc": 2,
      "writeibc": 0,
      "diribc": 0
    }
  ],
  "JSONversion": 1
}
```

For a non-specific request to list all filesystems, the following top-level attributes are also returned:

```
returnedRows - the number of filesystem items returned
totalRows - the total number of filesystems
```

*Figure 336. List UNIX Filesystems*

If more items than specified by X-IBM-Max-Items (or the default of 1000) match the request, then the following top-level attribute will be added to the top-level document:

```
"moreRows": true
```

#### JSON response attributes:

These attributes are the mnt3\_\* values returned from the w\_getmntent() C-Library API.

## Create a z/OS UNIX zFS filesystem

You can use this operation to create a z/OS UNIX zFS Filesystem.

### HTTP method and URI path

```
POST /zosmf/restfiles/mfs/zfs/<file-system-name>
```

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/mfs/zfs** a UNIX System Services filesystem request for a zFS aggregate. *<file-system-name>* is the filesystem (for zFS, the aggregate name) of the file system to be created. This is also the VSAM linear data set name.

### Optional Query Parameters

#### timeout={secs}

The number of seconds to wait for the "zfsadm format" command to complete before timing out with Category/RC/REAS = 1/8/9 "Command timed out". The default if not specified is 20 seconds. If a greater value is used, the "X-IBM-Async-Threshold" header should be used to allow the application process long running transactions.

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Input Content

Input content is a JSON document:

Table 345. Input Content	
Property	Description
owner	Defaults to 755. This property is not required.
group	Defaults to 755. This property is not required.
perms	Defaults to 755. This property is not required.
cylsPri	Defaults to 0. This property is required.
cylsSec	Defaults to 0. This property is not required.
storageClass	This property is not required.
managementClass	This property is not required.
dataClass	This property is not required.
volumes	This property is not required.
JSONversion:1	This property is not required.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 201 OK indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## Example

Refer to [Figure 337 on page 625](#) for an example of creating UNIX file systems.

```
request:
POST https://zosmf1.yourco.com/zosmf/restfiles/mfs/zfs/HLQ.MYNEW.ZFS HTTP/1.1
Content-Type: application/json
Content-Length: 86
{
  "cylsPri":100,
  "cylsSec": 10,
  "volumes": [ "ZFS001", "ZFS002"],
  "JSONversion":1
}

response:
201 Created
```

*Figure 337. Create UNIX Filesystems*

## Delete z/OS UNIX zFS Filesystem

You can use the delete z/OS UNIX zFS Filesystem operation to delete an existing zFS filesystem. Access Method Services are used to delete the filesystem linear data set. The file system must not be allocated (attached or mounted) for this operation to succeed.

### HTTP method and URI path

```
DELETE /zosmf/restfiles/mfs/zfs/<file-system-name>
```

Where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **/mfs/zfs** a UNIX System Services filesystem request for a zFS aggregate. *<file-system-name>* is the filesystem (for zFS, the aggregate name) of the file system to be deleted. This is also the VSAM linear data set name.

### Custom headers

#### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

#### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Required authorizations**

See [“Required authorizations” on page 567](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## **Expected response**

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 OK indicates success. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

For errors, the HTTP response includes error information as a JSON error report document. See [“Error report document” on page 640](#).

## **Example**

Refer to [Figure 338 on page 626](#) for an example of deleting UNIX Filesystems.

```
request:
DELETE https://zosmf1.yourco.com/zosmf/restfiles/mfs/zfs/HLQ.MYNEW.ZFS HTTP/1.1

response:
204 No Content
```

*Figure 338. Delete UNIX Filesystems*

## **Mount a z/OS UNIX file system**

You can use this operation to mount a z/OS UNIX file system on a specified directory.

### **HTTP method and URI path**

```
PUT /zosmf/restfiles/mfs/<file-system-name>
```

Where:

- /zosmf/restfiles specifies the z/OS data sets and files REST interface.
- /mfs is used for managing file systems.
- <file-system-name> is the file system that you want to mount.

## Content type

The content type is application/json.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Query parameters

None.

## Standard headers

None.

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together. Otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request.
- The service that is described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Request Body

The request body to mount a UNIX file system is shown in [Table 346 on page 627](#).

Table 346. Request body to mount a UNIX file system		
Field	Type	Description
action	String	Specifies the action mount to mount an UNIX file system.
mount-point	String	Specifies the mount point to be used for mounting the UNIX file system.

Table 346. Request body to mount a UNIX file system (continued)

Field	Type	Description
fs-type	String	Specifies the type of file system to be mounted. This value must match the TYPE operand on a FILESYSTYPE statement in the BPXPRMxx parmlib member for your system.
mode	String	Specifies the mode in which the file system is mounted, as follows: <ul style="list-style-type: none"> <li>• Specify <code>rdonly</code> for read only.</li> <li>• Specify <code>rdwr</code> for read/write.</li> </ul> The values are case-insensitive. If not specified, this value defaults to <code>rdonly</code> .

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. Status code 204 indicates success. A status code of `4nn` or `5nn` indicates that an error occurred. For details, see [“Error handling” on page 568](#).

If the request is successfully run, status code 204 indicates success and no content is returned.

## Example request

In the following example, the PUT method is used to mount a UNIX file system.

```
PUT /zosmf/restfiles/mfs/JIAHJ.ZOSMF.DRIVER.HFS HTTP/1.1
Content-Type: application/json; charset=UTF-8
```

### Request body

```
{"action": "mount", "mount-point": "/u/jiahj", "fs-type": "HFS", "mode": "rdonly"}
```

## Example response

A sample response is shown in Figure 339 on page 628.

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Thu, 17 Sep 2015 08:05:41 GMT
```

Figure 339. Example: Mount a UNIX file system

# Unmount a UNIX file system

You can use this operation to unmount a UNIX file system on a specified directory.

## HTTP method and URI path

PUT /zosmf/restfiles/mfs/<file-system-name>

where:

- **/zosmf/restfiles** specifies the z/OS data set and file REST interface
- **<file-system-name>** is the file system you want to unmount.

## Request Body

```
{
  "action" : "unmount"
}
```

## Request body to unmount a UNIX file system

The request body to unmount a UNIX file system is shown in [Request body to unmount a UNIX file system](#).

Table 347. Request body to unmount a UNIX file system		
Field	Type	Description
action	String	Unmount.

## Standard headers

None.

## Custom headers

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

## X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

None.

## Content type

The content type is application/json.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 568](#).

If the request is successfully executed, status code 204 indicates success and no content returned is returned.

## Example request

In the following example, the PUT method is used to unmount a UNIX file system.

```
PUT /zosmf/restfiles/mfs/JIAHJ.ZOSMF.DRIVER.HFS HTTP/1.1
Content-Type: application/json; charset=UTF-8
```

### Request body

```
{"action": "unmount"}
```

## Example response

A sample response is shown in [Unmount a UNIX file system](#).

```
204 No Content
Content-Type: application/json; charset=UTF-8
Content-Length: 0
Date: Thu, 17 Sep 2015 08:09:21 GMT
```

*Figure 340. Example: Unmount a UNIX file system*

## JSON document specifications for z/OS data set and file REST interface requests

This section describes the contents of the JSON documents that are used with z/OS data set and file REST interface requests.

For more information about the properties described in these JSON documents, see *z/OS ISPF Services Guide*.

The following JSON documents are described:

- [“Data set list document” on page 631](#)
- [Data set list with attributes document](#)
- [PDS/PDSE member list with attributes document](#)
- [Create a sequential and partitioned data set document](#)
- [“File list document” on page 638](#)
- [Unix file and directory list with attributes document](#)
- [“Mount and unmount a file system document” on page 639](#)
- [Create a UNIX file document](#)
- [“Error report document” on page 640.](#)

### Data set list document

[Table 348 on page 631](#) shows the contents of the JSON data set list document.

For more information about these properties, see the dialog variables for the YES | NO | PRT parameter of **LMDLIST** (list data sets) in [LMDLIST-list data sets](#) in *z/OS ISPF Services Guide*.

Table 348. Contents of the JSON data set list document		
Property	Description	Required
<b>items</b>	An array where each element contains the following key:value pairs	Yes
<b>dsname</b>	Data set name.	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
<b>JSONversion</b>	Version number of this JSON document.	Yes
<b>vol</b>	Volume serial	No
<b>used</b>	Percentage of used tracks or pages (PDSE)	No
<b>extx</b>	Number of extents used, long format (5 bytes)	No
<b>cdate</b>	Creation date	No
<b>edate</b>	Expiration date	No
<b>rdate</b>	Date last referenced	No

Table 348. Contents of the JSON data set list document (continued)		
Property	Description	Required
<b>migr</b>	Whether the data set is migrated (YES or NO) based on the value of the VOLUME_OF_MIGRATED_DATA_SETS keyword in the ISPF configuration table. If the volume name of the data set matches the value of VOLUME_OF_MIGRATED_DATA_SETS, ZDLMIGR is set to YES, otherwise it is set to NO.	No
<b>dsntp</b>	Dsname type (PDS, LIBRARY, or ' ')	No
<b>spacu</b>	Space units	No
<b>mvol</b>	Whether the data set is multivolume (Y) or not (N)	No
<b>ovf</b>	Space overflow indicator (YES or NO)	No
<b>dsorg</b>	Data set organization	No
<b>recfm</b>	Record format	No
<b>lrecl</b>	Logical record length	No
<b>blksz</b>	Block size	No
<b>size</b>	Data set size in tracks, long format (12 bytes)	No
<b>catnm</b>	Name of the catalog where the data set is located	No
<b>dev</b>	Device type	NO

## Data set list with attributes document

“Data set list with attributes document” on page 632 shows the attributes of the JSON data set list document.

For more information about these properties, see the dialog variables for the YES | NO | PRT parameter of **LMDLIST** (list data sets) in [LMDLIST-list data sets](#) in *z/OS ISPF Services Guide*.

Table 349. X-IBM-Attributes=vol		
Property	Description	Required
<b>items</b>	An array where each element contains the following key:value pairs <b>dsname</b> Data set name. <b>vol</b> Volume in which the data set resides.	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
<b>JSONversion</b>	Version number of this JSON document.	Yes

Table 350. X-IBM-Attributes=dsname

Property	Description	Required
<b>items</b>	An array where each element contains the following key value pairs: <b>dsname</b> Data set name.	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	Optional property; set to true when more rows can be returned.	No
<b>totalRows</b>	Total number of rows that match the request.	No
<b>JSONversion</b>	Version number of this JSON document.	Yes

Table 351. X-IBM-Attributes=base

Property	Description	Required
<b>items</b>	An array where each element contains the following key value pairs: <a href="#">Table 352 on page 633</a>	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
<b>JSONversion</b>	Version number of this JSON document.	Yes

Table 352. Items key:value pairs

Property	Description	Required
<b>dsname</b>	Data set name.	Yes
<b>blksz</b>	Block size.	No
<b>catnm</b>	Name of the catalog in which the data set was located.	No
<b>cdate</b>	Creation date.	No
<b>dev</b>	Device type.	No
<b>dsntp</b>	Dsname type (PDS, LIBRARY, or ' ') Percentage of used tracks or pages(PDSE).	No
<b>dsorg</b>	Data set organization.	No
<b>edate</b>	Expiration date.	No
<b>extx</b>	Number of extents used, long format (5 bytes).	No
<b>lrecl</b>	Logical record length.	No

Table 352. Items key:value pairs (continued)

Property	Description	Required
<b>migr</b>	Whether the data set is migrated (YES or NO) based on the value of the VOLUME_OF_MIGRATED_DATA_SETS keyword in the ISPF configuration table. If the volume name of the data set matches the value of VOLUME_OF_MIGRATED_DATA_SETS, ZDLMIGR is set to YES, otherwise it is set to NO.	No
<b>mvol</b>	Whether the data set is multivolume (Y) or not (N).	No
<b>ovf</b>	Space overflow indicator (YES or NO).	No
<b>rdate</b>	Date last referenced.	No
<b>recfm</b>	Record format.	No
<b>size</b>	Data set size in tracks, long format (12 bytes).	No
<b>spacu</b>	Space units.	No
<b>used</b>	Percentage of used tracks or pages (PDSE).	No
<b>vol</b>	Volume serial.	No
<b>vols</b>	If mvol is Y , all the volumes are filled in this field,otherwise the value is equal to vol.	No

## PDS/PDSE member list with attributes document

Contents of the [JSON PDS/PDSE member list document](#) shows the contents of the JSON member list document.

For more information about these properties, see the dialog variables for the YES | NO parameter description of **LMMFIND** (find a library member) in [LMMFIND—find a library member in z/OS ISPF Services Guide](#).

Table 353. Contents of the JSON PDS/PDSE member list document

Property	Description	Required
<b>items</b>	An array where each element contains the following key:value pairs <b>member</b> Member name	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
<b>JSONversion</b>	Version number of this JSON document.	Yes

Table 354. X-IBM-Attributes=base and data set RECFM=F or V

Property	Description	Required
<b>items</b>	An array where each element contains the following key:value pairs. See Table 355 on page 635.	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
<b>JSONversion</b>	Version number of this JSON document.	Yes

Table 355. Items key:value pairs for data set RECFM=F or V

Property	Description	Required
<b>member</b>	member name.	Yes
<b>vers</b>	Version number; a number from 1 to 99.	No
<b>mod</b>	Modification level; a number from 0 to 99.	No
<b>c4date</b>	Creation date in 4-character year format	No
<b>m4date</b>	Last change date in 4-character year format	No
<b>cnorc</b>	Current number of records; a number from 0 to 65 535.	No
<b>inorc</b>	Beginning number of records; a number from 0 to 65 535.	No
<b>mnorc</b>	Number of changed records; a number from 0 to 65 535.	No
<b>mtime</b>	Last change time; a character value in the format hh:mm.	No
<b>msec</b>	Seconds value of the last change time. This is a two character field.	No
<b>user</b>	User ID of last user to change the given member; an alphanumeric field with a maximum length of 7 characters.	No
<b>sclm</b>	Indicates whether the member was last modified by SCLM or ISPF. A value of Y indicates the last update was made through SCLM. A value of N indicates that the last update was made.	No

Table 356. X-IBM-Attributes=base and data set RECFM=U

Property	Description	Required
<b>items</b>	An array where each element contains the following key:value pairs. See Table 357 on page 636.	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	No
<b>JSONversion</b>	Version number of this JSON document.	Yes

Table 357. Items key:value pairs for data set RECFM=U

Property	Description	Required
<b>member</b>	member name.	Yes
<b>ac</b>	A 2-character field containing the authorization code of the member.	No
<b>alias-of</b>	An 8-character field containing the name of the real member that this member is an alias of. If the member is not an alias this field is blank.	No
<b>amode</b>	A 3-character field containing the AMODE of the member.	No
<b>attr</b>	A 20-character field containing the load module attributes. The attributes are 2-character strings separated by blanks. These strings can appear in the attribute string: NX Not executable OL Only Loadable OV Overlay RF Refreshable RN Reentrant RU Reusable SC Scatter Load TS Test	No
<b>rmode</b>	A 3-character field containing the RMODE of the member.	No
<b>size</b>	An 8-character field containing the load module size in hex.	No

Table 357. Items key:value pairs for data set RECFM=U (continued)		
Property	Description	Required
<b>ttr</b>	A 6-character field containing the TTR of the member.	No
<b>ssi</b>	An 8-character field containing the SSI information for a load module.	No

## Create a sequential and partitioned data set document

Table 358 on page 637 shows the contents of the JSON sequential and partitioned data set document.

For more information about these properties, see the dialog variables for the YES | NO parameter description of **LMMFIND** (find a library member) in [LMMFIND—find a library member](#) in *z/OS ISPF Services Guide*.

Table 358. Contents of the create a sequential and partitioned data set document		
Property	Description	Required
<b>volser</b>	Volume serial number of the device a data set will reside on.	No
<b>unit</b>	Unit name of the device that the data set will reside on.	No
<b>dsorg</b>	Unit name of the device that the data set will reside on.	No
<b>alcunit</b>	Unit of space allocation.	No
<b>primary</b>	Primary space allocation for the data set.	No
<b>secondary</b>	Secondary space allocation for the data set.	No
<b>dirblk</b>	Number of directory blocks for a partitioned data set.	No
<b>avgblk</b>	Specifies the unit of space allocations to be blocks and sets the average block length.	No
<b>recfm</b>	Record format of a data set: The following formats are supported: <ul style="list-style-type: none"> <li>• F: fixed</li> <li>• V: variable</li> <li>• U: undefined</li> <li>• B: block</li> <li>• FB: fixed blocked</li> <li>• VB: variable blocked</li> </ul>	No
<b>blksize</b>	Block size of the data set.	No
<b>lrecl</b>	Record length of data set.	No
<b>storclass</b>	Specifies the storage class of system managed storage.	No
<b>mgntclass</b>	Specifies the management class of the data set.	No
<b>dataclass</b>	Specifies the data class of the data set.	No

Table 358. Contents of the create a sequential and partitioned data set document (continued)

Property	Description	Required
<b>dsntype</b>	Data set type. The following types are supported: <ul style="list-style-type: none"> <li>• BASIC</li> <li>• EXTPREF</li> <li>• EXTREQ</li> <li>• HFS</li> <li>• LARGE</li> <li>• PDS</li> <li>• LIBRARY</li> <li>• PIPE</li> </ul>	No

## File list document

Table 359 on page 638 shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service in z/OS ISPF Services Guide](#).

Table 359. Contents of the JSON file list document

Property	Description
<b>items</b>	JSON array of member entries containing the properties that follow in the remainder of this table.
<b>filename</b>	File name
<b>returnedRows</b>	Number of rows that were returned for this request.
<b>moreRows</b>	Optional property; set to true when more rows can be returned.
<b>totalRows</b>	Total number of rows that match the request.

## Unix file and directory list with attributes document

[Contents of the unix file and directory](#) shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service in z/OS ISPF Services Guide](#).

Table 360. Contents of the unix file and directory

Property	Description	Required
<b>items</b>	An array where each element contains the following key:value pairs. See <a href="#">Unix items key:value pairs</a> .	Yes
<b>returnedRows</b>	Number of rows that were returned for this request.	Yes
<b>moreRows</b>	True, if more rows are available to return.	No

Table 360. Contents of the unix file and directory (continued)

Property	Description	Required
<b>totalRows</b>	Total number of data sets found matching the dslevel and volser criteria. If you specify ",total" as suffix in X-IBM-Attributes header, like "dsname,total", or "base,total", or "vol,total".	Yes
<b>JSONversion</b>	Version number of this JSON document	Yes

Table 361. Unix items key:value pairs

Property	Description	Required
<b>name</b>	File or directory name.	Yes
<b>mode</b>	indicating the permissions.	No
<b>size</b>	For regular files, the file's size in bytes. For other kinds of files, the value of this field is unspecified.	No
<b>uid</b>	The numeric user ID (UID) of the file's owner.	No
<b>user</b>	The user name of the file's owner got by UID.	No
<b>gid</b>	The numeric group ID (GID) of the file's group.	No
<b>group</b>	The group name of the file's group got by GID.	No
<b>mtime</b>	The most recent time the contents of the file were changed.	No
<b>target</b>	If the file is symlink, this indicates the really file/directory	No

## Mount and unmount a file system document

Contents of the [Mount and unmount a file system document](#) shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the `column-list` parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service](#) in *z/OS ISPF Services Guide*.

Table 362. Contents of the Mount and unmount a file system document

Property	Description	Required
<b>action</b>	Current support value: mount: you are going to mount an UNIX file system on the specified directory. unmount: you are going to unmount a specified directory.	Yes

Table 362. Contents of the Mount and unmount a file system document (continued)

Property	Description	Required
<b>mount-point</b>	Specify the mount point you are going to mount/unmount, generally, it is directory.	No
<b>fs-type</b>	Specify the file system type you are going to mount, --the name must match the TYPE operand on a FILESYSTYPE statement in the BPXPRMxx parmlib member for the file system.  For the mount action, this is required; but it is not required for the unmount action.	No
<b>mode</b>	Specify the mode you intend to mount the file system:  Support Value: rdonly: read only rdwr: read write --all the values are case insensitive	No

## Create a UNIX file document

Contents of the [create a UNIX file document](#) shows the contents of the JSON file list document for UNIX files.

For more information about these properties, see the column-list parameter of the **DIRLIST** (directory list service) in [DIRLIST—directory list service in z/OS ISPF Services Guide](#).

Table 363. Contents of the create a UNIX file document

Property	Description	Required
<b>type</b>	The request type, support value: directory or dir: you are going to create a directory. file: you are going to create a file.	No
<b>mode</b>	The mode specifies the file/directory permission bits to be used in creating the file/directory.  <b>rwXrwxrwx</b>  The 9 characters are in three groups of 3; they describe the permissions on the file. The first group of 3 describes owner permissions; the second describes group permissions; the third describes other (or "world") permissions. Characters that might appear are:  r: Permission to read the file w: Permission to write on the file x: Permission to execute the file -: No such a permission	No

## Error report document

Table 364 on page 641 shows the contents of the JSON error report document for z/OS data set and file REST interface requests.

<i>Table 364. Contents of the JSON error report document</i>	
Property	Description
<b>category</b>	Error category. This field is integer data type.
<b>rc</b>	Return code. This field is integer data type.
<b>reason</b>	Reason code. This field is integer data type.
<b>message</b>	Message that describes the error.
<b>details</b>	(Optional) Array of strings containing additional message details.

For the meanings of the category, rc, and reason fields, see [“Error reporting categories”](#) on page 641.

## Error reporting categories

This section describes the error categories and associated error codes that can be returned in the JSON error report document for z/OS data set and file REST interface requests. This document is described in [“Error report document”](#) on page 640.

### Categories

Table 365 on page 641 shows the error categories that are defined for errors that are returned in z/OS data set and file REST interface operations.

<i>Table 365. Error categories for z/OS data set and file REST interface operations</i>			
Category	Ordinal Value	Description	Where the error details are described
<b>Service</b>	1	Errors that are produced or detected in the service layer.	<a href="#">“Category 1 — Service error”</a> on page 642
<b>Message queue</b>	2	Errors that are produced or detected by the message queue.	<a href="#">“Category 2 — Message queue error”</a> on page 643
<b>CEA</b>	3	Errors that are produced or detected by the common event adapter (CEA) interface.	<a href="#">“Category 3 — Common event adapter (CEA) error”</a> on page 644
<b>ISPF</b>	4	Errors that are produced or detected by the Interactive System Productivity Facility (ISPF) interface.	<a href="#">“Category 4 — ISPF error”</a> on page 645
<b>CSI</b>	5	Errors that are produced or detected by the catalog search interface (CSI).	<a href="#">“Category 5 — Catalog Search Interface (CSI) error”</a> on page 645
<b>Read or write function</b>	6	Errors that are returned from an attempted read or write request.	<a href="#">“Category 6 — Read or write error”</a> on page 645
<b>JSON</b>	7	Errors that are produced or detected when parsing JSON.	<a href="#">“Category 7 — JSON Parser Conditions”</a> on page 647
<b>XL C/C++</b>	8	Errors that are produced or detected by z/OS XL C/C++	<a href="#">“Category 8 — z/OS XL C/C++ Conditions”</a> on page 647
<b>Unexpected</b>	16	Unexpected errors detected.	<a href="#">“Category 16 — Unexpected error”</a> on page 647

## Category 1 – Service error

Table 366 on page 642 shows the possible conditions for this error category.

Table 366. Category 1 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
4	1	%s	The request specified a URL that is not valid. %s indicates which parts of the URL are invalid, such as path information, data set name and volume serial.
4	2	Invalid data set name length.	The request specified a data set name length that is not valid. Data set names cannot exceed 44 characters in length.
4	3	The specific data set name is invalid.	The request specified a data set name (dslevel) that is not valid.
4	4	Retrieving USS files should start with absolute path.	The requested file cannot be retrieved because the request is missing an absolute path name, which is required. When retrieving UNIX files, the file path must be an absolute (or fully qualified) path name, rather than a relative (or partially qualified) path name.
4	5	Length of USS path is invalid.	The request specified a UNIX file path length that is not valid.
4	6	The specific path is invalid.	The request specified a UNIX file path that is not valid.
4	7	The specific volser is invalid.	The request specified a volume serial (volser) that is not valid.
4	8	File not found.	The request specified a file that does not exist.
4	9	Incorrect content type.	The request specified an unsupported content type.
4	10	Unrecognized HTTP method.	The request is not recognized as a supported HTTP method.
4	11	POST method is not supported.	The POST method was specified, however POST is not a supported HTTP method.
4	12	Incorrect attribute was specified.	The request contained one or more attributes that are not valid.
4	13	Incorrect parameter was specified.	The request contained one or more parameters that are not valid.
4	14	The specific JSON data in the request is invalid.	JSON data in the request is invalid.
4	15	The size of the data to be written is invalid.	The size of the data to be written is invalid.

Table 366. Category 1 errors for z/OS data set and file REST interface operations (continued)

rc	reason	message	Description
4	16	Request content-length must be specified. Request content-length was too long. Request content was longer than specified content-length.	Only used in ValidateJsonServlet.C.
4	17	Request document contained invalid JSON.	Not currently used.
4	18	Member name is not valid.	Member name is not valid.
4	19	Unix file/directory exists.	UNIX file or directory already exists.
8	1	Unable to get JSON Response.	Not currently used.
8	2	Unable to get JSON Response Table.	Not currently used.
8	3	Unable to load ISPEXEC.	The ISPF service route ISPEXEC cannot be loaded.
8	4	Unable to load TSO servlet mappings.	The servlet dispatcher was not able to load the servlet mappings JSON file.
8	5	A servlet-mapping was not found for servlet-path.	A servlet-mapping matching the request cannot be found. Indicates a servlet-mapping configuration error.
8	6	ExceptionHandler threw an exception after commit. Servlet failed but could not send error response.	An exception was thrown by the servlet.
8	7	TsoServlet already committed.	An exception was thrown by the servlet after its output stream header was committed.
16	1	Error occurred when connecting to remote server.	An error occurred when the X-IBM-Target-System header was set. See details to take action.

## Category 2 – Message queue error

Table 367 on page 643 shows the possible conditions for this error category.

Table 367. Category 2 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
4	1	Timeout	Timeout occurred when receiving a message from the message queue. MSG_QUEUE_PROTOCOL_ERROR_TIMEOUT
4	2	Received unexpected msgType=nn	Unexpected message received. MSG_QUEUE_PROTOCOL_ERROR_UNEXPECTED

Table 367. Category 2 errors for z/OS data set and file REST interface operations (continued)

rc	reason	message	Description
4	3	ServletDispatcher failed.	Back-end servlet dispatcher failed with the reason <reason>. For example, TSO prompt was received when TsoServletResponse was expected. MSG_QUEUE_PROTOCOL_ERROR_ENDED
4	4	Queue full while sending.	Message queue is full while sending the specific <message type>. MSG_QUEUE_PROTOCOL_ERROR_FULL
4	5	Illegal state.	Message queue is in an invalid state. MSG_QUEUE_PROTOCOL_ERROR_ILLEGAL_STATE
5	1	Error parsing TsoServletResponse.	Error occurred when parsing TsoServletResponse. MSG_QUEUE_JSON_PARSE_ERROR
5	2	Error serializing TsoServletRequest.	Error occurred when serializing TsoServletRequest. MSG_QUEUE_JSON_SERIALIZE_ERROR
12	1	The message queue cannot be created.	Message queue cannot be created.
12	2	Reserved.	Not used currently.
12	3	Message queue size is less than minimum.	Message queue size is less than the required minimum size. MSG_QUEUE_ERROR_SIZE_ERROR
12	4	Message prefix bytes are too short.	Message prefix bytes are shorter than expected. MSG_QUEUE_ERROR_DECODING
16	n(errno)	Varies.	Error from msgsnd(), msgrcv(), or a related message queue service. The UNIX errno is the reason code.  For reason code 1141, the following causes are possible: <ul style="list-style-type: none"> <li>• The TSO/E logon PROC is not correct. Ensure that the information specified on the RESTAPI_FILE statement in the IZUPRMxx parmlib member is correct.</li> <li>• The user ID does not have a TSO segment in the RACF user profile.</li> <li>• TSO/E logon exit is active and is preventing the user ID from logging on.</li> </ul> MSG_QUEUE_SYS_ERROR

### Category 3 – Common event adapter (CEA) error

Table 368 on page 645 shows the possible conditions for this error category.

Table 368. Category 3 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
12	1	TSO launcher exception: Client is not authorized for instrumentation.	The requester lacks sufficient authority to access the requested common event adapter (CEA) service.
12	2	TSO launcher exception: Error occurred.	The requested CEA service encountered an error.
12	3	TSO launcher exception: CEA address space is not available.	The CEA address space is not active or is not available.
12	4	TSO launcher exception: TSO address space cannot be created.	The TSO/E address space cannot be created because a required system resource is not available.
12	5	CeaTsoEnd request failed.	An error occurred ending a TSO Address Space. CEA TSO Reason code = 4103. CEA_ERROR_END_TSO_FAILED
12	6		Not currently used. CEA_ERROR_CMD_NOT_FOUND
12	7	Unable to find the system you requested	CEA_ERROR_NO_TARGET_SYSTEM
12	n	Error occurs while using remote CEA TSO address space services	Refer to CEA TSO address space services reason code and diagnostic code for details.

## Category 4 – ISPF error

Table 369 on page 645 shows the possible conditions for this error category.

Table 369. Category 4 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
n	m	varies	The return and reason code values match the return and reason code values that are set by the ISPF service.

## Category 5 – Catalog Search Interface (CSI) error

Table 370 on page 645 shows the possible conditions for this error category.

Table 370. Category 5 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
n	m	varies	The return and reason code values match the return and reason code values that are set by the consolidated software inventory (CSI) service.

## Category 6 – Read or write error

Table 371 on page 646 shows the return and reason codes that can be set for a read or write request.

Table 371. Category 6 errors for z/OS data set and file REST interface operations

rc	reason (hex)	message	Description
8	201	<methodName> failed	Unable to open a data set or member, <methodName> like fopen() or freopen() failed. RW_ERROR_OPEN_FAILED
8	202	<methodName> failed	Unable to close a data set or member, <methodName> like fclose() failed. If this error occurs for a write or put operation, the data set contents are not predictable. failedRW_ERROR_CLOSE_FAILED
8	204	Client ETag does not match the current ETag for the data set.	The attempt to write to the data set failed because the supplied ETag does not match the current ETag of the requested data set. This mismatch indicates that the data set content was modified in the time since the caller obtained the ETag. RW_ERROR_DS_ETAG_NOT_MATCHED
8	208	.<methodName> error	An error, <methodName> like fread(), occurred during I/O to a data set or UNIX file. RW_ERROR_IO
8	20C	Member not found.	The member cannot be located in the partitioned data set. Perhaps the data set name or member name was incorrectly specified. RW_ERROR_MBR_NOT_FOUND
8	0000020A	Dynamic allocation failed.	This RC is combined with S99ERROR in the high halfword. RW_ERROR_DS_DYNALLOC_ERR
8	1708020A	ISPF LMINIT - data set not found.	The specified data set cannot be found. Perhaps the data set name or member name was incorrectly specified. RW_ERROR_DS_NOT_FOUND
8	varies	Varies.	For UNIX file I/O errors, the reason code consists of the errno in the high order 16 bits and the errno2 in the low order 16 bits.
8	5E30062	File not found.	The specified UNIX file cannot be found. Perhaps the data set name or member name was incorrectly specified. RW_ERROR_FS_NOT_FOUND
8	5B6F0002	Client is not authorized for file access.	The request for the UNIX file failed because the caller does not have sufficient authority to access the file. RW_ERROR_FS_AUTH

Table 371. Category 6 errors for z/OS data set and file REST interface operations (continued)			
rc	reason (hex)	message	Description
8	406	Client ETag does not match the current ETag for the file.	The attempt to write to the UNIX file failed because the supplied ETag does not match the current ETag of the requested file. This mismatch indicates that the file content was modified in the time since the caller obtained the ETag.  RW_ERROR_FS_ETAG_NOT_MATCHED

## Category 7 – JSON Parser Conditions

Category 7 JSON parser conditions shows the possible conditions for this error category.

Table 372. Category 7 JSON parser conditions			
rc	reason	message	Description
n	m	Varies.	The rc and reason are set from the low-level JSON Parser return code and reason code.

## Category 8 – z/OS XL C/C++ Conditions

Category 8 z/OS XL C/C++ Conditions shows the possible conditions for this error category.

Table 373. Category 8 z/OS XL C/C++ Conditions			
rc	reason	message	Description
n	m	Varies.	The rc and reason are set from the low-level z/OS XL C/C++ return code and reason code.  If the message indicates a dynamic allocation error, then the rc is a decimal value and corresponds to a DYNALLOC reason code. For more information about the DYNALLOC reason code, see Interpreting error reason codes from DYNALLOC in <i>z/OS MVS Programming: Authorized Assembler Services Guide</i> .

## Category 16 – Unexpected error

Table 374 on page 647 shows the possible conditions for this error category.

Table 374. Category 9 errors for z/OS data set and file REST interface operations			
rc	reason	message	Description
16	1	Server error occurred.	For details about the exception, check the z/OSMF logs.

## z/OS jobs REST interface

The z/OS jobs REST interface is an application programming interface (API) implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with batch jobs on a z/OS system, as described in this topic.

Table 375 on page 648 lists the operations that the z/OS jobs REST interface services provide.

Table 375. Operations provided through the z/OS jobs REST interface services.

Operation	HTTP method and URI path
<b>“Obtain the status of a job” on page 652</b>	GET /zosmf/restjobs/jobs/<jobname>/<jobid>?[step-data=Y N] GET /zosmf/restjobs/jobs/<correlator>?[step-data=Y N]
<b>“List the jobs for an owner, prefix, or job ID” on page 655</b>	GET /zosmf/restjobs/jobs[?<parms>]
<b>“List the spool files for a job” on page 659</b>	GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files GET /zosmf/restjobs/jobs/<correlator>/files
<b>“Retrieve the contents of a job spool file” on page 661</b>	GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/<nnn>/records GET /zosmf/restjobs/jobs/<correlator>/files/<nnn>/records GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/JCL/records GET /zosmf/restjobs/jobs/<correlator>/files/JCL/records
<b>“Submit a job” on page 665</b>	PUT /zosmf/restjobs/jobs[/-<JESB>]
<b>“Hold a job” on page 669</b>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<b>“Release a job” on page 672</b>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<b>“Change the job class” on page 675</b>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<b>“Cancel a job” on page 677</b>	PUT /zosmf/restjobs/jobs/<jobname>/<jobid> PUT /zosmf/restjobs/jobs/<correlator>
<b>“Cancel a job and purge its output” on page 680</b>	DELETE /zosmf/restjobs/jobs/<jobname>/<jobid> DELETE /zosmf/restjobs/jobs/<correlator>

## Using the Swagger interface

You can use the Swagger interface to display information about the z/OS jobs REST APIs. The Swagger interface includes one section: Jobs APIs. For more information, see [“Using the Swagger interface” on page 49](#).

## Processing overview

The z/OS jobs REST interface services can be invoked by any HTTP client application, running on the z/OS local system or a remote system.

Your program (the client) initiates an HTTP request to the z/OS jobs REST interface. If the interface determines that the request is valid, it performs the requested service. After performing the service, the z/OS jobs REST interface creates an HTTP response. If the request is successful, this response takes the form of an HTTP 2nn response and, if applicable, a result set that is passed back to your program.

Depending on which service was requested, the result set might be returned in a format that requires parsing by your program, for example, a JSON object. In other cases, results might be returned in another

format, such as plain text or binary data. If the request is not successful, the response consists of a non-OK HTTP response code with details of the error that is provided in the form of a JSON object. The contents of the JSON objects are described in [“JSON document specifications for z/OS jobs REST interface requests”](#) on page 682.

## Resource URLs

The URLs of the z/OS jobs REST interface have the format that is shown in [Figure 341](#) on page 649:

```
https://{host}:{port}/zosmf/restjobs/jobs/{-jesb}/{resource}?{parm}
```

*Figure 341. Format of resource URLs for z/OS jobs REST interface.*

Where:

- "https://{host}:{port}" specifies the target system address and port.
- "/zosmf/restjobs/jobs/" identifies the z/OS jobs REST interface.
- "JESB" optionally specifies a secondary JES subsystem, if one is to be used to process the request. If you omit this value, the request is processed by the primary JES subsystem.
- "{resource}?{parm}" represents the resource, such as a job name and job ID, and optionally one or more parameters, to qualify the request.

## HTTP methods

The z/OS jobs REST interface provides the following HTTP methods:

### GET

Retrieves information about jobs that are running on the z/OS system.

### PUT

Updates job information on the z/OS system, or sets attributes and performs actions on jobs.

### DELETE

Removes jobs from the z/OS system.

Some situations might require the use of the POST method; see [“Usage considerations for the z/OSMF REST services”](#) on page 3.

## Supported HTTP versions

z/OS jobs REST interface supports requests in either of the following protocols: HTTP/1.0 or HTTP/1.1

## Content types

The data that is sent or returned by the HTTP methods has one of the following content types:

- Application/octet-stream (Content-Type: application/octet-stream) is used for data that is sent or returned in an uninterpreted format, such as a job that is submitted, or binary data or records that are obtained from a z/OS job spool file.
- JSON (Content-Type: application/json) is used for sent data and returned data. For the detailed format of each JSON object, see the description for each operation.
- Plain text (Content-Type: plain/text).

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the problem or provide it to IBM Support, if required. For the contents of the error report document, see [“Error report document”](#) on page 688.

The following HTTP status codes are valid:

**HTTP 200 OK**

Success.

**HTTP 201 Created**

Request was successful, and, as a result, a resource was created.

**HTTP 202 Accepted**

Request was received and accepted for processing, but the processing has not yet completed.

**HTTP 400 Bad request**

Request contained incorrect parameters.

**HTTP 500 Internal server error**

Programming error.

## Synchronous support for the job modify operations

The z/OS jobs REST interface includes services that you can use to perform job modify operations, as shown in [Table 376](#) on page 650.

Table 376. Job modify operations provided through the z/OS jobs REST interface services.	
Operation	Where described
Hold a job.	<a href="#">“Hold a job”</a> on page 669
Release a job.	<a href="#">“Release a job”</a> on page 672
Change the job class.	<a href="#">“Change the job class”</a> on page 675
Cancel a job.	<a href="#">“Cancel a job”</a> on page 677
Delete a job (cancel a job and purge its output).	<a href="#">“Cancel a job and purge its output”</a> on page 680

These services can be run synchronously if coded to use the latest version of the service. To request synchronous processing, set the "version" property in your request to 2.0 or omit the "version" property. If so, the system attempts to process the request synchronously, if such processing is supported on the target JES subsystem. Synchronous processing is supported for JES2 subsystems only. When the target subsystem is JES3, a synchronous request is ignored and the service is performed asynchronously.

Generally, the differences in processing are as follows:

- For an asynchronous request, z/OSMF returns control to the caller immediately. However, to verify that the initial request was performed, the caller must then issue the service that is described in [“Obtain the status of a job”](#) on page 652.
- For a synchronous request, z/OSMF does not return control to the caller until the requested action is performed and results of the request are available to be returned to the caller. Here, the JSON job feedback document provides more information about the success or failure of the request; see [“Job feedback document”](#) on page 685.

If your program does not require feedback on the results of requested actions, you can use these services asynchronously.

Due to timing behavior, if you submit a job and immediately issue a synchronous request for the same job, you might receive the error message "No job found for reference" in the JSON error report document (category 6, return code 4, reason code 10). To avoid this occurrence, it is recommended that you allow a small amount of time to pass between a job submit request and a subsequent job modify request.

## Required system setup

The z/OS jobs REST interface services require that the System REXX (SYSREXX) component is set up and active on your z/OS system. For information, see [Ensuring that System REXX is set up and active in IBM z/OS Management Facility Configuration Guide](#).

## Required authorizations

Generally, your user ID requires the same authorizations for using the z/OS jobs REST interface services as when you perform these operations through a TSO/E session on the system. For example, submitting a job through the z/OS jobs REST interface requires that your user ID is authorized to run jobs on the system and can access any protected resources that the job might require.

In addition, your user ID requires authorization to the z/OSMF SAF profile prefix on the target z/OS system, as follows:

- READ access to <SAF\_PREFIX> in the APPL class
- READ access to the <SAF\_PREFIX>.\*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF\_PREFIX>.IzuManagementFacilityRestJobs resource name in the EJBROLE class.

By default, the z/OSMF SAF profile prefix is IZUDFLT.

If you are using z/OS jobs REST interface services to perform job modify operations, your user ID must be authorized to the appropriate resources in the JESJOBS class, as shown in [Table 377 on page 651](#).

Table 377. JESJOBS class authorizations needed for performing job modify operations		
Operation	JESJOBS resource	Access required
Hold a job	HOLD.nodename.userid.jobname	UPDATE
Release a job	RELEASE.nodename.userid.jobname	UPDATE
Change the job class	MODIFY.nodename.userid.jobname	UPDATE
Cancel a job	CANCEL.nodename.userid.jobname	ALTER
Delete a job (cancel a job and purge its output)	CANCEL.nodename.userid.jobname	ALTER

For more information about JESJOBS class, see [Controlling the use of job names in z/OS Security Server RACF Security Administrator's Guide](#).

If run asynchronously, these services also require that your user ID is authorized to the Common Information Model (CIM) server and permitted to the JES2-JES3Jobs CIM provider. CIM includes jobs (CFZSEC and CFZRCUST) to help you configure the CIM server, including security authorizations and file system customization. For information, see [Quick guide: CIM server setup and verification in z/OS Common Information Model User's Guide](#).

Where applicable, additional authorization requirements are noted in the descriptions of the individual z/OS jobs REST interface services. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Requesting asynchronous job notifications

You can use the asynchronous job notifications function of z/OSMF to allow your programs to be notified when submitted jobs complete. With this function, the program that submits the job through the z/OS jobs REST interface services PUT method specifies a URL when submitting the job. When the job ends, z/OSMF returns an HTTP message to the URL location, indicating the job completion status. The returned data is in the form of a JSON document.

The asynchronous job notifications function is available for JES2 systems only; it is not available for JES3 systems.

The key requirement is that you must create a subscription to the Common Information Model (CIM) jobs indication provider for your system. Also, if the job notifications require a secure network connection, you must enable an SSL connection between the client application and the server, including the sharing of digital certificates. For instructions on enabling the asynchronous job notifications function, see [Configuring your system for asynchronous job notifications](#) in *IBM z/OS Management Facility Configuration Guide*.

## Enabling browser login through a client certificate

It is possible to run the z/OS jobs REST interface services directly from a web browser. Here, you must first authenticate to z/OSMF through your browser. In z/OSMF authentication is typically done by entering your user ID and password at the **Welcome** page. However, it is also possible to log in with a client certificate, if your installation favors this approach. With a client certificate, you can access z/OSMF through your browser without having to supply a user ID and password.

In client certificate authentication, the certificate is stored in the browser itself. When you log in to z/OSMF, the server requests the certificate from your browser. If your browser stores more than one certificate, you might be prompted to select the correct one to use with z/OSMF. Otherwise, your browser sends the certificate to z/OSMF. After z/OSMF identifies you as the owner of the key that is associated with the certificate, a secure connection is established.

If z/OSMF does not accept your client certificate, z/OSMF displays the Welcome page for you to enter your user ID and password.

If your installation plans to enable client certificate login for the z/OS jobs REST interface services, understand that it is your responsibility to create the certificate and manage its distribution to users. It is recommended that you ensure that users have browsers that support importing a certificate.

For information about creating digital certificates, see [RACF and digital certificates](#) in *z/OS Security Server RACF Security Administrator's Guide*.

## Error logging

Errors from the z/OS jobs REST interface services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Enabling traces for IBM analysis

For diagnostic purposes, your installation might be asked by IBM Support to enable tracing for the z/OS jobs REST interface. For information, see [Appendix A, “Enabling tracing for the z/OS jobs REST interface,”](#) on page 977

## Obtain the status of a job

You can use this operation to obtain the status of a batch job on z/OS.

### HTTP method and URI path

---

```
GET /zosmf/restjobs/jobs/<jobname>/<jobid>?[step-data=Y|N]
GET /zosmf/restjobs/jobs/<correlator>?[step-data=Y|N]
```

---

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job for which status is requested.

- **<correlator>** identifies the job for which status is requested. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.
- **[step-data]** is an optional parameter that indicates whether the service returns information about each step in the job that completed, such as the step name, step number, and completion code. To return the step data for completed steps, set the value for this parameter to Y. Otherwise, omit the parameter or set its value to N (default). If a step is run multiple times, duplicate data for that step might be included in the response.

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

You can specify following optional query parameters on this request:

### **user-correlator**

The user portion of the job correlator. This value is 1-32 characters in length, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 31) can be any combination of uppercase alphabetic, numeric (0-9), or special. Blank characters are not supported.

This query parameter is mutually exclusive with jobid.

This value is processed by the JES2 subsystem only; the JES3 subsystem does not process the correlator and instead, indicates zero job matches. For a system with JES3 as the primary subsystem, and one or more JES2 secondary subsystems, the primary JES3 subsystem does not process the correlator. However, the JES2 secondary subsystems can process the correlator.

### **exec-data**

This optional parameter specifies whether to return execution data about the job, if execution data is available. This parameter is a string value and is case-insensitive. Valid values are:

- Y (or y)
- N (or n)

For example: exec-data=Y

The following execution data is returned for each job:

**exec-system**

System name of the z/OS system on which the job ran (up to 8 characters)

**exec-member**

Member name of the z/OS system on which the job ran (up to 8 characters)

**exec-submitted**

Time when the job was submitted to run (the input end time)

**exec-started**

Time when job execution started

**exec-ended**

Time when job execution ended.

Timestamps are presented in the JSON UTC format: "yyyy-mm-ddThh:mm:ss.mmmZ"

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark (?) to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).
- Wildcard characters are permitted in the owner and prefix query parameter values. Use an asterisk (\*) for multiple characters, and a question mark (?) for a single character.

## Required authorizations

See [“Required authorizations” on page 651](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

---

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

---

where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

## Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with a JSON job document. For the contents, see [“Job document” on page 682](#).

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 688](#).

## Example request

The following request obtains the status for the job BLSJPRMI, job ID STC00052:

---

```
GET /zosmf/restjobs/jobs/BLSJPRMI/STC00052?exec-data=Y HTTP/1.1
Host: zosmf1.yourco.com
```

---

## Example responses

A sample response is shown in [Figure 342 on page 655](#).

```

HTTP/1.1 200 OK
Date: Sat, 03 Nov 2018 09:06:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "STC00052",
  "jobname": "BLSJPRMI",
  "subsystem": "JES2",
  "owner": "IBMUSER",
  "status": "OUTPUT",
  "type": "STC",
  "class": "STC",
  "retcode": "CC 0000",
  "url": "https://\host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A",
  "files-url": "https://\host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A/files",
  "job-correlator": "S0000052SY1.....CE35BDE8.....:",
  "phase": 20,
  "phase-name": "Job is on the hard copy queue",
  "exec-system": "SY1",
  "exec-member": "SY1",
  "exec-submitted": "2018-11-03T09:05:15.000Z",
  "exec-started": "2018-11-03T09:05:18.010Z",
  "exec-ended": "2018-11-03T09:05:25.332Z"
}

```

Figure 342. Example: Returned job status

A sample response for an active step is shown in [Figure 343 on page 655](#).

```

HTTP/1.1 200 OK
Date: Sat, 03 Nov 2018 09:06:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "retcode": "null",
  "jobname": "BLSJPRMI",
  "status": "ACTIVE",
  "job-correlator": "S0000052SY1.....CE35BDE8.....:",
  "class": "STC",
  "type": "STC",
  "jobid": "STC00052",
  "url": "https://\host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A",
  "phase-name": "Job is on the hard copy queue",
  "step-data": [
    {
      "smfid": "SP21",
      "active": true,
      "step-number": 1,
      "proc-step-name": "STARTING",
      "step-name": "IEFPROC",
      "program-name": "BLSQPRMI"
    }
  ],
  "owner": "IBMUSER",
  "subsystem": "JES2",
  "files-url": "https://\host:port/zosmf/restjobs/jobs/S0000052SY1.....CE35BDE8.....%3A/files",
  "phase": 20,
  "exec-system": "SY1",
  "exec-member": "SY1",
  "exec-submitted": "2018-11-03T09:05:15.000Z",
  "exec-started": "2018-11-03T09:05:18.010Z",
  "exec-ended": "2018-11-03T09:05:25.332Z"
}

```

Figure 343. Example: Returned status for an active step

## List the jobs for an owner, prefix, or job ID

You can use this operation to list the jobs for an owner, prefix, or job ID.

### HTTP method and URI path

```
GET /zosmf/restjobs/jobs[?<parms>]
```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- **<parms>** are optional parameters that you can use to qualify the request. For a list of the supported parameters, see [“Query parameters” on page 656](#).

## Custom headers

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

You can specify one or more of the following optional query parameters on this request:

### **owner**

User ID of the job owner whose jobs are being queried; the default is the z/OS user ID. Folded to uppercase; cannot exceed 8 characters.

### **prefix**

Job name prefix; default is \*. Folded to uppercase; cannot exceed 8 characters.

### **jobid**

Job ID. Folded to uppercase; cannot exceed 8 characters. This query parameter is mutually exclusive with user-correlator.

### **max-jobs**

Maximum number of jobs returned. The value must be in the range 1 - 1000. If this parameter is not specified, or is specified incorrectly, the default value of 1000 is used.

### **user-correlator**

The user portion of the job correlator. This value is 1 – 32 characters in length, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 31) can be any combination of uppercase alphabetic, numeric (0-9), or special. Blank characters are not supported.

This query parameter is mutually exclusive with jobid.

This value is processed by the JES2 subsystem only; the JES3 subsystem does not process the correlator and instead indicates zero job matches. For a system with JES3 as the primary subsystem, and one or more JES2 secondary subsystems, the primary JES3 subsystem does not process the correlator, however, the JES2 secondary subsystems can process the correlator.

### **exec-data**

This optional parameter specifies whether to return execution data about the job, if execution data is available. This parameter is a string value and is case-insensitive. Valid values are:

- Y (or y)
- N (or n)

For example: `exec-data=Y`

The following execution data is returned for each job:

#### **exec-system**

System name of the z/OS system on which the job ran (up to 8 characters)

#### **exec-member**

Member name of the z/OS system on which the job ran (up to 8 characters)

#### **exec-submitted**

Time when the job was submitted to run (the input end time)

#### **exec-started**

Time when job execution started

#### **exec-ended**

Time when job execution ended.

Timestamps are presented in the JSON UTC format: `"yyyy-mm-ddThh:mm:ss.mmmZ"`

By default, execution data for up to 100 jobs is returned. If this parameter is specified, its value overrides the `max-jobs` value, which defaults to 1000. If you set `max-jobs` to less than 100, the `max-jobs` value is used. Otherwise, the default of 100 is used.

### **status**

You can use this optional parameter to limit the returned data to only jobs that are currently active. This parameter is case-insensitive.

The following values are valid:

- `status=ACTIVE` or `status=active`

If you omit this parameter, the response includes data for both active and inactive jobs.

If you set this parameter to an incorrect value, the parameter is ignored and the default is used.

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark (?) to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).
- Wildcard characters are permitted in the owner and prefix query parameter values. Use an asterisk (\*) for multiple characters, and a question mark (?) for a single character.

## **Required authorizations**

See [“Required authorizations” on page 651](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, observe the following considerations for this request:

- The ordering of the jobs that are returned is not predictable.
- If the maximum number of jobs is returned, no indication is provided for whether more jobs remain to be retrieved.
- This request can be directed to a secondary JES subsystem. To do so, use the following request format:

```
https://host:port/zosmf/restjobs/jobs/-JESB
https://host:port/zosmf/restjobs/jobs/-JESB?owner=owner
```

where *JESB* is the name of the secondary JES subsystem in these examples.

- To list the jobs for a job ID on a secondary JES subsystem, you must specify the job ID, rather than a job correlator. For example:

```
https://host:port/zosmf/restjobs/jobs/-JESB?jobid=jobid
```

## Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with an array of matching jobs, each as a JSON job document. For the contents, see [“Job document” on page 682](#).

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 688](#).

## Example request

In the following example, the GET method is used to list the jobs that are owned by IBMUSER and have a job name prefix that begins with TESTJOB:

```
GET /zosmf/restjobs/jobs?owner=IBMUSER&prefix=TESTJOB*&exec-data=Y
Host: zosmf1.yourco.com
```

## Example response

A sample response is shown in [Figure 344 on page 658](#).

```
HTTP/1.1 200 OK
Date: Sat, 03 Nov 2018 09:07:12 +0000GMT
Content-Type: application/json
Connection: close

[
  {
    "jobid": "JOB000023", "jobname": "TESTJOB2", "subsystem": null, "owner": "IBMUSER",
    "status": "OUTPUT", "type": "JOB", "class": "A", "retcode": "CC 0000",
    "url": "https://host:port/zosmf/restjobs/jobs/TESTJOB2/JOB000023",
    "files-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB2/JOB000023/files",
    "exec-system": "SY1", "exec-member": "SY1", "exec-submitted": "2018-11-03T09:05:15.000Z",
    "exec-started": "2018-11-03T09:05:18.010Z", "exec-ended": "2018-11-03T09:05:25.332Z",
    {
      "jobid": "JOB000024", "jobname": "TESTJOB3", "subsystem": null, "owner": "IBMUSER",
      "status": "OUTPUT", "type": "JOB", "class": "A", "retcode": "ABEND S000",
      "url": "https://host:port/zosmf/restjobs/jobs/TESTJOB3/JOB000024",
      "files-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB3/JOB000024/files",
      "exec-system": "SY1", "exec-member": "SY1", "exec-submitted": "2018-11-03T09:06:30.000Z",
      "exec-started": "2018-11-03T09:06:36.020Z", "exec-ended": "2018-11-03T09:06:50.007Z"
    }
  ]
```

*Figure 344. Example: Returned list of the jobs for a specific owner and job name prefix*

## List the spool files for a job

You can use this operation to list the spool files for a batch job on z/OS.

### HTTP method and URI path

---

```
GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files
GET /zosmf/restjobs/jobs/<correlator>/files
```

---

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job for which the spool files are to be listed.
  - **<correlator>** identifies the job for which the spool files are to be listed. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.
- **/files** indicates that the response is to list the spool files for the specified job.

### Custom headers

#### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

#### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

#### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

### Query parameters

None.

### Required authorizations

See [“Required authorizations” on page 651](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

In addition, note that this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid/files
```

where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

## Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with an array of zero or more JSON job file documents. For the contents, see [“Job file document”](#) on page 686.

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document”](#) on page 688.

## Example request

The following request lists the spool files for the job TESTJOB1, job ID JOB00023:

```
GET /zosmf/restjobs/jobs/TESTJOB1/JOB00023/files HTTP/1.1
```

## Example response

A sample response is shown in [Figure 345](#) on page 660.

```
HTTP/1.1 200 OK
Date: Thu, 17 Jan 2013 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

[
  {
    "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 1,
    "stepname": "JESE", "procstep": null, "class": "H",
    "ddname": "JESMSGGL", "record-count": 14, "byte-count": 1200,
    "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/1/records",
    {
      "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 2,
      "stepname": "JESE", "procstep": null, "class": "H",
      "ddname": "JESJCL", "record-count": 10, "byte-count": 526,
      "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/2/records",
      {
        "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 3,
        "stepname": "JESE", "procstep": null, "class": "H",
        "ddname": "JESYSMSG", "record-count": 14, "byte-count": 1255,
        "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/3/records",
        {
          "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 4,
          "stepname": "STEP57", "procstep": "COMPILE", "class": "H",
          "ddname": "SYSUT1", "record-count": 6, "byte-count": 741,
          "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/4/records",
          {
            "jobid": "JOB00023", "jobname": "TESTJOB1", "subsystem": null, "id": 5,
            "stepname": "STEP57", "procstep": "COMPILE", "class": "A",
            "ddname": "SYSPRINT", "record-count": 3, "byte-count": 209,
            "records-url": "https://host:port/zosmf/restjobs/jobs/TESTJOB1/JOB00023/5/records"
          }
        }
      }
    }
  ]
```

*Figure 345. Example: Returned list of spool files*

## Retrieve the contents of a job spool file

You can use this operation to retrieve the contents of a job spool file on z/OS. Also, you can use this service to retrieve the JCL that was used to submit the job.

### HTTP method and URI path

---

```
GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/<nnn>/records
GET /zosmf/restjobs/jobs/<correlator>/files/<nnn>/records
GET /zosmf/restjobs/jobs/<jobname>/<jobid>/files/JCL/records
GET /zosmf/restjobs/jobs/<correlator>/files/JCL/records
```

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job for which the spool file contents are requested.
  - **<correlator>** identifies the job for which the spool file contents are requested. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.
- **/files<nnn>/records** indicates that the request is to retrieve the contents of a job spool file for the specified job. The **<nnn>** parameter is the ID for the spool file from which the contents are to be retrieved.
- **/files/JCL/records** indicates that the request is to retrieve the JCL for the specified job.

### Custom headers

You can include the following optional custom HTTP header with this request:

#### X-IBM-Record-Range

Use this header to retrieve a range of records from a spool file. You can specify this range by using either of the following formats:

##### SSS-EEE

Where **SSS** identifies the start record and **EEE** identifies the end record to be retrieved. Both values are relative offsets (0-based). When **EEE** is set to 0, records through the end of the spool file are retrieved.

##### SSS,NNN

Where **SSS** identifies the start record and **NNN** identifies the number of records to be retrieved.

For an example of how this custom header is used, see [“Examples” on page 663](#).

#### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

#### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance”](#) on page 398 was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

You can specify one or more of the following optional query parameters on this request.

### **mode**

Use the mode parameter to specify conversion options for the returned data. The following values are valid for mode:

#### **text**

The z/OS jobs REST interface converts records from the server code page to the client code page and returns the records with Content-Type: plain/text. Trailing spaces are removed and newline characters are used as record separators. This value is the default if you omit the mode parameter.

#### **binary**

The z/OS jobs REST interface performs no conversion and returns the records with Content-Type: application/octet-stream.

#### **record**

The z/OS jobs REST interface performs no conversion and returns the records with Content-Type: application/octet-stream. The z/OS jobs REST interface prefixes each record with a 4-byte (big endian) length.

Specifying the mode parameter with any other value, or no value, results in the default: mode=text.

When mode=text, the following query parameters can be used to further qualify the request. These parameters cannot be used when mode is set to record or binary; doing so results in an error.

- fileEncoding
- search
- research
- insensitive
- maxreturnsize

These query parameters are described as follows.

### **fileEncoding=code-page**

Specifies an alternative code page (EBCDIC) for the spool file on z/OS; the encoded text is converted to the client's request encoding. If not specified, the default code page is IBM-1047.

### **search=<string>**

The spool file is searched for the first record that contains the string, without respect to case (by default). Optionally, insensitive=false can be specified for case-sensitive matching.

This parameter cannot be used with the research parameter.

### **research=<regular-expression>**

The spool file is searched for the first record that matches the given extended regular expression. For example, research=A|B finds A or B. By default, the search is case-insensitive. To search for case-sensitive matches, specify the research parameter with the query parameter insensitive=false.

This parameter cannot be used with the search parameter.

### **insensitive=true|false**

When 'true', searches (search and research) are case-insensitive. For case-sensitive searches, specify 'false'. The default is 'true'.

### **maxreturnsize=<integer>**

This parameter can be specified only with search= or research=.

The value given is the maximum number of records to return.

The default, if not specified, is 100.

For the search and research queries, records are returned starting with the first matching record. The X-IBM-Record-Range request header can be used to specify the range of records to be searched, but it does not restrict the number of records returned (see maxreturnsize).

If no X-IBM-Record-Range request header is present, the search begins with the first record. In all cases, an X-IBM-Record-Range=p,q response header is returned, where p is the first matching record and q is the number of records returned. If no matching records are found, the response header X-IBM-Record-Range=0,0 is returned.

The parameter cannot be used if the mode query parameter specifies any option except 'text'.

## **Required authorizations**

See [“Required authorizations” on page 651](#).

In addition, your user ID requires READ access to the JESSPOOL profile for the spool data set. If no profile exists, only the user who created the spool data set can access, modify, or delete it. For information about spool data set security considerations, see [z/OS JES Application Programming](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, observe the following considerations for this request:

- The response does not include the Content-Length header. Because the server streams the data rather than buffering it in memory, the server cannot determine the total content length of the data before it completes the transfer. For similar reasons, the response does not include the Content-Range header, either.
- This request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid/filesJCL/records
```

Where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

## **Expected response**

On completion, the z/OS jobs REST interface returns an HTTP response with content type that is defined by the mode query parameter.

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 688](#).

## **Examples**

In the following example, the GET method is used to retrieve the contents of spool file 1 for the job TESTJOB, job ID JOB00023:

```
GET /zosmf/restjobs/jobs/TESTJOB/JOB00023/files/1/records HTTP/1.1
Host: zosmf1.yourco.com
```

A sample response is shown in [Figure 346 on page 664](#).

```

HTTP/1.1 200 OK

Date: Thu, 17 Jan 2013 05:39:28 +0000GMT
Content-Type: text/plain
Connection: close

      J E S 2 J O B L O G  -- S Y S T E M E I M G  -- N O D E  D C E I M G W V

15.49.11 JOB00239 ---- MONDAY, 14 JAN 2013 ----
15.49.11 JOB00239 IRR010I USERID IBMUSER IS ASSIGNED TO THIS JOB.
15.49.11 JOB00239 ICH70001I IBMUSER LAST ACCESS AT 15:48:25 ON MONDAY, JANUARY 14, 2013
15.49.11 JOB00239 $HASP373 INSTALL STARTED - INIT 2 - CLASS A - SYS EIMG
15.49.11 JOB00239 IEF403I INSTALL - STARTED - TIME=15.49.11
15.49.16 JOB00239 IEF404I INSTALL - ENDED - TIME=15.49.16
15.49.16 JOB00239 $HASP395 INSTALL ENDED
----- JES2 JOB STATISTICS -----
      14 JAN 2013 JOB EXECUTION DATE
          71 CARDS READ
        287 SYSOUT PRINT RECORDS
          0 SYSOUT PUNCH RECORDS
          13 SYSOUT SPOOL KBYTES
          0.08 MINUTES EXECUTION TIME

```

*Figure 346. Example: Returned spool file content*

In the following example, the GET method is used to retrieve a range of records (the first 250) using the X-IBM-Record-Range custom header:

```

GET /zosmf/restjobs/jobs/TESTJOB/JOB00023/files/8/records HTTP/1.1
X-IBM-Record-Range: 0-249

```

A sample response is shown in [Figure 347 on page 664](#).

```

HTTP/1.1 200 OK

Date: Thu, 17 Jan 2013 05:39:28 +0000GMT
Content-Type: text/plain
Connection: close

...(the first 250 records)

```

*Figure 347. Example: Returned spool file content (a range of records)*

In the following example, the GET method is used to retrieve the JCL for the job TESTJOB, job ID JOB00060:

```

GET /zosmf/restjobs/jobs/TESTJOB/JOB00060/files/JCL/records HTTP/1.1

```

A sample response is shown in [Figure 348 on page 664](#).

```

HTTP/1.1 200 OK

//TESTJOB JOB (),MSGCLASS=H
// EXEC PGM=IEFBR14

```

*Figure 348. Example: Returned job content (the job JCL)*

## Submit a job

You can use this operation to submit a job to run on z/OS.

### HTTP method and URI path

---

```
PUT /zosmf/restjobs/jobs[/-<JESB>]
```

---

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- **<JESB>** represents an optionally specified secondary JES subsystem. If you omit this value, the request is processed by the primary JES subsystem.

### Standard headers

Use the following standard HTTP header with this request:

#### Content-Type

One of the following values:

- Set to `text/plain` when the optional header `X-IBM-Intrdr-Mode` is set to `TEXT` or is omitted, and the job JCL is included in the request.
- Set to `application/octet-stream` when optional header `X-IBM-Intrdr-Mode` is set to `RECORD` or `BINARY`, and the JCL for the job to be submitted is included in the HTTP request.
- Set to `application/json` when the job to be submitted resides in a data set or UNIX file, which is identified in a JSON document (included as input with this request).

### Custom headers

Optionally, you can include one of the following custom HTTP headers with this request:

#### X-IBM-Intrdr-Class

A single character that specifies the internal reader class; the default is `A`. This value defines the default message class (MSGCLASS) for the job.

#### X-IBM-Intrdr-Recfm

A single character that specifies the internal reader record format: `F` or `V`.

When submitting a job from a data set, you can omit this header. Otherwise, this value must match the record format of the data set.

When not submitting a job from a data set, if you omit this header or specify a value other than `F` or `V`, the default of `F` is used.

#### X-IBM-Intrdr-Lrecl

An integer value that specifies the internal reader logical record length (LRECL).

When submitting a job from a data set, you can omit this header. Otherwise, this value must match the LRECL of the data set.

When not submitting a job from a data set, if you omit this header or specify a non-integer value, the default of 80 is used.

#### X-IBM-Intrdr-Mode

A keyword that specifies the format of the input job: `TEXT`, `RECORD`, or `BINARY`.

When submitting a job from a data set, you can omit this header. Otherwise, this value must be set to `RECORD`.

When not submitting a job from a data set, observe the following rules:

- If you omit this header, the TEXT keyword is used by default.
- If you specify the BINARY keyword, the X-IBM-Intrdr-Recfm header must be omitted or set to F (the default).
- If you specify the RECORD keyword or BINARY keyword, you must set Content-Type to application/octet-stream.

### **X-IBM-User-Correlator**

Specifies the user portion of the job correlator. In z/OS, a job correlator can be used to associate each job with a unique 64-character value. The correlator provides you with a means to query a job in the system and track it through execution.

A job correlator consists of a 31-byte system-defined portion and a colon character (:), followed by a 32-byte user portion. The system-defined portion contains the following values:

- 8-byte job ID
- 8-byte MAS name for the system on which the job resides
- 8-byte sequence value
- 7-bytes of reserved space.

Following the system value is the colon character (:) separator and the second string: an optional 32-character user-defined value (the user portion). This value is 1 – 32 characters in length, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 31) can be any combination of uppercase alphabetic, numeric (0-9), or special. Blank characters are not supported.

If specified, the user portion is combined with the system portion, producing the full job correlator that will be returned in the job-correlator property of the JSON job document. If the user portion is not specified, the returned job correlator is the 32-byte system value, ending with the colon (:).

If this header is specified when JES3 is the primary job entry subsystem, an error results and no job is submitted.

For more information on the job correlator, see *z/OS JES2 Commands*.

### **X-IBM-JCL-Symbol-name**

Specifies the name and value for a JCL symbol. The symbol name is included in the header, at the name position. The characters that follow 'X-IBM-JCL-Symbol-' up to the colon separator (:) form the symbol name. The data that follows the colon specifies the value for the symbol.

A symbol name is 1–8 characters, where the first character must be uppercase alphabetic (A-Z) or special (\$, #, @). The remaining characters (up to 7) can be any combination of uppercase alphabetic, numeric (0-9), or special.

A symbol value is limited to 255 characters. Multiple symbol names and values can be specified, up to a limit of 128.

**Example:** X-IBM-JCL-Symbol-MBR: ABC specifies symbol name MBR with value ABC. Specifying this custom header and submitting a job that uses //MYDD DSN=MY.DATASET(&MBR.) , DISP=SHR in the JCL will cause ABC to be substituted as the member name.

If this header is specified when JES3 is the primary job entry subsystem, an error results and no job is submitted.

For more information on JCL symbols, see [Using system symbols and JCL symbols in z/OS MVS JCL Reference](#).

### **X-IBM-Notification-URL**

Specifies a destination URL for receiving an HTTP POST when the job is no longer eligible for execution (that is, when the job reaches the output queue or purge queue). The notification is in the form of a JSON document (Content-Type: application/json), which contains job status information. For the contents of the JSON document, see [“Job completed document” on page 684](#).

### **X-IBM-Intrdr-File-Encoding**

This optional header specifies that the EBCDIC code page is to be used for encoding the data that is written to the internal reader. If not specified, the default is IBM-1047. This header is ignored when optional header X-IBM-Intrdr-Mode is set to RECORD or BINARY.

This header is effective only when the JCL for the job to be submitted is included in the HTTP request body. This header is ignored when the job to be submitted resides in a data set or UNIX file.

### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

None.

## **Input to this request**

- Internet media type: text/plain, application/octet-stream, or application/json
- HTTP request with optional headers, followed by job to be submitted or a JSON document identifying the location of the job to be submitted (a data set or UNIX file).

To submit a job, you can include the job JCL in the HTTP request itself, or you can have the request refer to a job that resides in a data set or UNIX file. Here, you include a JSON document ("Content-Type: application/json" with the HTTP request. The JSON document contains the property "file": "<file-name>" where <file-name> identifies the data set or UNIX file that contains the job to be submitted.

Use the JSON document to identify the data set or UNIX file containing the job to be submitted, as follows:

- For a data set, specify the qualified data set name, prefixing the data set name with two leading forward slash characters ("//").

If not fully qualified, the current z/OSMF user ID is prefixed to the data set name. Supported data set types include sequential data sets and members of partitioned data sets.

Data sets must be catalogued.

- For a z/OS UNIX file, specify the absolute path name of the file.

Code page conversion is not performed on the contents of the file.

For a migrated data set, this operation does not cause the data set to be retrieved, unless you specify otherwise. To request that a data set be recalled without waiting, you can specify the "recall" property with the value "yesnowait" to the input JSON document. Unique error responses are provided when a migrated data set is requested to be recalled without waiting and for when a migrated data set is not requested to be recalled. In both cases, no job is submitted. If you have asked for a recall, without waiting, when you try the submit again, you should do so without adding the "recall" property to the JSON document or by changing the "recall" property to the value "no."

## Required authorizations

See [“Required authorizations” on page 651](#).

In addition, your user ID must be authorized to run jobs on the system and be able to access any protected resources that the job might require. For information about the security considerations for job submission, see [Controlling job submission in z/OS JES2 Initialization and Tuning Guide](#) or [Authorizing the Use of Input Sources in z/OS JES3 Initialization and Tuning Guide](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, observe the following considerations for this request:

- This request can be directed to a secondary JES subsystem. To do so, use the following request format:

```
https://host:port/zosmf/restjobs/jobs/-JESB
```

Where *JESB* is the name of the secondary JES subsystem in these examples.

## Expected response

On completion, the z/OS jobs REST interface returns an HTTP response with a JSON job document. For the contents, see [“Job document” on page 682](#).

For errors, z/OS jobs REST interface returns an appropriate HTTP status code and error information as a JSON error report document. See [“Error report document” on page 688](#).

## Example of submitting a job from a data set or UNIX file

Table 378 on page 668 shows variations of a PUT request that submits the job TESTJOBX to run on z/OS. In each variation, the PUT request contains a JSON statement that identifies the location of the job to be submitted.

Table 378. Variations of a PUT request for submitting a job from a data set or UNIX file.	
Location of the job	Example
<b>Partitioned data set (fully qualified)</b>	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A  { "file" : "'/'MYJOBS.TEST.CNTL(TESTJOBX)'" }</pre>
<b>Partitioned data set (non-fully qualified)</b>	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A  { "file" : "'/TEST.CNTL(TESTJOBX)'" }</pre>

Table 378. Variations of a PUT request for submitting a job from a data set or UNIX file. (continued)

Location of the job	Example
<b>Sequential data set</b>	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A  { "file" : "'MYJOBS.TEST.JOB1'" }</pre>
<b>UNIX file</b>	<pre>PUT /zosmf/restjobs/jobs HTTP/1.1 Content-Type: application/json X-IBM-Intrdr-Class: A X-IBM-Intrdr-Recfm: V X-IBM-Intrdr-Lrecl: 255 X-IBM-Intrdr-Mode: TEXT  { "file" : "/u/myjobs/job1" }</pre>

### Example of a request that contains the job JCL

The following request submits the job TESTJOBX to run on z/OS. Here, the JCL for the job to be submitted is contained in the PUT request.

```
PUT /zosmf/restjobs/jobs HTTP/1.1
Host: zosmf1.yourco.com
Content-Type: text/plain
X-IBM-Intrdr-Class: A
X-IBM-Intrdr-Recfm: F
X-IBM-Intrdr-Lrecl: 80
X-IBM-Intrdr-Mode: TEXT

//TESTJOBX JOB ( ),MSGCLASS=H
// EXEC PGM=IEFBR14
```

A sample response is shown in [Figure 349 on page 669](#).

```
HTTP/1.1 201 Created
Date: Fri, 17 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00025", "jobname": "TESTJOBX", "subsystem": null, "owner": "IBMUER",
  "status": "INPUT", "type": "JOB", "class": "A", "retcode": null,
  "url": "https://host:port/zosmf/restjobs/jobs/TESTJOBX/JOB00025",
  "files-url": "https://host:port/zosmf/restjobs/jobs/TESTJOBX/JOB00025/files"
}
```

Figure 349. Example: Returned results of a job submission

## Hold a job

For a job that has been submitted to run on z/OS, but not yet selected for processing, you can use this operation to hold the job. When held, a job is not be eligible for selection.

You can use a similar method to release the job and make is available for selection; see [“Release a job” on page 672](#).

## HTTP method and URI path

```
PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>
```

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job to be held.
  - **<correlator>** identifies the job to be held. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

## Custom headers

### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### X-IBM-Target-System-Password

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## Query parameters

None.

## Input to this request

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

### "request": "hold"

Indicates a request to hold a job.

### "version": "n.n"

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so,

the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 650](#).

## Required authorizations

See [“Required authorizations” on page 651](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, observe the following consideration for this request:

- This request can be directed to a secondary JES subsystem. To do so, use one of the following request formats:

---

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid  
https://host:port/zosmf/restjobs/jobs/-JESB/correlator
```

---

where:

- *JESB* is the name of the secondary JES subsystem.
- The job to be held is identified by either the job name and job ID (*jobname/jobid*) or the job correlator (*correlator*).

## Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 652](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
  - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 685](#).
  - Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 688](#).

## Example request

The following request specifies that the job TESTJOBW, job ID JOB00023, is to be held. With the inclusion of the "version" property set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
PUT /zosmf/restjobs/jobs/TESTJOBW/JOB00023 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "request": "hold",
  "version": "2.0"
}
```

### Example response

A sample response is shown in [Figure 350 on page 672](#).

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00023",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00023",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000023SY1....CC20F378.....",
  "status": "0"
}
```

*Figure 350. Example: Returned results of a job hold request*

## Release a job

For a job that has been held from execution on z/OS, you can use this operation to release the job. When released, a job is made eligible for selection to execute.

### HTTP method and URI path

```
PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>
```

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job to be released.
  - **<correlator>** identifies the job to be released. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

### Custom headers

#### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-

System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

#### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

#### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

### **Query parameters**

None.

### **Input to this request**

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

#### **"request": "release"**

Indicates a request to release a job.

#### **"version": "n.n"**

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 650](#).

### **Required authorizations**

See [“Required authorizations” on page 651](#).

### **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, observe the following consideration for this request:

- This request can be directed to a secondary JES subsystem. To do so, use one of the following request formats:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
https://host:port/zosmf/restjobs/jobs/-JESB/correlator
```

where:

- JESB is the name of the secondary JES subsystem.

- The job to be released is identified by either the job name and job ID (*jobname/jobid*) or the job correlator (*correlator*).

## Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 652](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
  - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 685](#).
  - Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 688](#).

## Example request

The following request specifies that the job TESTJOBW, job ID JOB00023, is to be released. With the inclusion of the "version" property set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
PUT /zosmf/restjobs/jobs/TESTJOBW/JOB00023 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "request": "release"
  "version": "2.0"
}
```

## Example response

A sample response is shown in [Figure 351 on page 674](#).

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00023",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00023",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000023SY1.....CC20F378.....",
  "status": "0"
}
```

*Figure 351. Example: Returned results of a job release request*

## Change the job class

You can use this operation to change the class of a job on z/OS.

### HTTP method and URI path

---

```
PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>
```

---

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job for which the class is to be changed.
  - **<correlator>** identifies the job for which the class is to be changed. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

### Custom headers

#### **X-IBM-Target-System = <string>**

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

#### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

#### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

### Query parameters

None.

### Input to this request

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

**"class": "<new\_job\_class>"**

Indicates a request to change the job class to the value <new\_job\_class>.

### **"version": "n.n"**

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 650](#).

## **Required authorizations**

See [“Required authorizations” on page 651](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, observe the following considerations for this request:

- The specified job class is not validated on input. To verify the success of this request, your program can issue a GET request for the job status, and check the class value in the returned JSON Job document. See [“Obtain the status of a job” on page 652](#).
- This request can be directed to a secondary JES subsystem. To do so, use the following request format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

---

where *JESB* is the name of the secondary JES subsystem.

- A request to a secondary JES subsystem must include the job name and job id, rather than a job correlator.

## **Expected response**

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 652](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
  - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 685](#).
  - Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 688](#).

## **Example request**

The following request specifies job class A for the job TESTJOBW, job ID JOB00023. With the inclusion of the "version" property set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

---

```
PUT /zosmf/restjobs/jobs/TESTJOBW/JOB00023 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "class": "A",
  "version": "2.0"
}
```

---

### Example response

A sample response is shown in [Figure 352 on page 677](#).

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00023",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00023",
  "owner": "IBMUUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000023SY1.....CC20F378.....",
  "status": "0"
}
```

*Figure 352. Example: Returned results of a job class change*

## Cancel a job

You can use this operation to cancel a job on z/OS.

### HTTP method and URI path

---

```
PUT /zosmf/restjobs/jobs/<jobname>/<jobid>
PUT /zosmf/restjobs/jobs/<correlator>
```

---

Where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job to be canceled.
  - **<correlator>** identifies the job to be canceled. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

### Custom headers

#### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

### **X-IBM-Target-System-User**

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance” on page 398](#) was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

None.

## **Input to this request**

- Internet media type: application/json
- HTTP request with JSON document containing the following properties:

**"request": "cancel"**

Indicates a request to cancel a job.

**"version": "n.n"**

Specifies the version of the service to be used, either 1.0 or 2.0.

To request asynchronous processing for this service, set the "version" property to 1.0. To request synchronous processing (the default), set "version" to 2.0 or omit the property from the request. If so, the system will attempt to process the request synchronously, if such processing is supported on the target JES subsystem.

For further considerations, see [“Synchronous support for the job modify operations” on page 650](#).

## **Required authorizations**

See [“Required authorizations” on page 651](#).

In addition, your user ID must be authorized to cancel the job on the system. For information about the security considerations for job cancellation, see [Controlling job modification and cancellation in z/OS JES2 Initialization and Tuning Guide](#) or [Controlling Who Can Cancel Jobs by Job Name in z/OS JES3 Initialization and Tuning Guide](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

In addition, note that this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

where *JESB* is the name of the secondary JES subsystem in these examples. A request to a secondary JES subsystem must include the job name and job id, rather than a job correlator.

## Expected response

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service described in [“Obtain the status of a job” on page 652](#).
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
  - Status code 200 indicates that the synchronous request was processed successfully. This status, however, does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document” on page 685](#).
  - Status code of 4nn or 5nn indicates that an HTTP error has occurred.

For errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document” on page 688](#).

## Example request

The following request cancels the job TESTJOB2, job ID JOB00084 on z/OS. To request synchronous processing by the target JES subsystem, the request includes the specification "version": "2.0".

```
PUT /zosmf/restjobs/jobs/TESTJOB2/JOB00084 HTTP/1.1
Host: zosmf1.yourco.com
Content-Length: 40
Content-Type: application/json

{
  "request": "cancel",
  "version": "2.0"
}
```

## Example response

A sample response is shown in [Figure 353 on page 679](#).

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00084",
  "jobname": "TESTJOB2",
  "original-jobid": "JOB00084",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000084SY1.....CC20F378.....",
  "status": "0"
}
```

*Figure 353. Example: Returned results of a job cancellation*

## Cancel a job and purge its output

You can use this operation to cancel a job and purge its output.

### HTTP method and URI path

---

```
DELETE /zosmf/restjobs/jobs/<jobname>/<jobid>
DELETE /zosmf/restjobs/jobs/<correlator>
```

---

where:

- **/zosmf/restjobs/jobs/** identifies the z/OS jobs REST interface.
- To identify the job in the request, use either the combination of the job name and job ID, or the job correlator, as follows:
  - **<jobname>/<jobid>** identifies the job to be canceled and purged.
  - **<correlator>** identifies the job to be canceled and purged. Specify the full job correlator for the job: The 31-byte system portion, a semicolon, and the user portion (up to 32 bytes). The correlator can be one that you obtained from the "job-correlator" property in a returned JSON job document.

### Custom headers

You can include the following optional custom HTTP header with this request:

#### X-IBM-Job-Modify-Version

Use this header to specify whether the request is to be processed asynchronously or synchronously, as follows:

##### 1.0

Specifies that the request is to be processed asynchronously. In response, the caller receives an HTTP status code of 202 Accepted, with no indication of the success or failure of the request. To verify that the initial request was performed, the caller can issue the service that is described in [“Obtain the status of a job” on page 652](#).

##### 2.0

Specifies that the request is to be processed synchronously. In response, the caller receives an HTTP status code, which indicates the results of the request. For a successful request, the caller also receives the JSON job feedback document, which includes details about the job that was cancelled.

If this header is omitted, the request is processed asynchronously by default.

Synchronous processing is supported for JES2 only. On systems running JES3, the z/OS jobs REST interface services must run asynchronously.

For an example of how this header is specified, see [“Example request” on page 681](#).

#### X-IBM-Target-System = <string>

This header indicates the target system name (nick name) for this request, where the system name (nick name) is defined in the local system Systems table. The target host system must support single-sign-on by using either an LTPA token or a valid X-IBM-Target-System-User and X-IBM-Target-System-Password is provided for the target system. If the target system is the local system, this header is ignored and has no effect.

#### X-IBM-Target-System-User

This header indicates the z/OS user ID that allows the user to access the target system. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

If this header is not provided in the current request, the current request uses the authenticated user credentials to access the target system if either of the following conditions are true:

- The X-IBM-Target-System-User header was provided in a previous request
- The service described in [“Authenticate with a secondary z/OSMF instance”](#) on page 398 was issued in a previous request.

### **X-IBM-Target-System-Password**

This header indicates the password that is associated with the z/OS user ID. If the X-IBM-Target-System header is not supplied, this header is ignored. Both X-IBM-Target-System-Password and X-IBM-Target-System-User must be provided together; otherwise, this header is ignored.

## **Query parameters**

None.

## **Required authorizations**

See [“Required authorizations”](#) on page 651.

In addition, your user ID must be authorized to cancel the job on the system, which allows the user to delete the job SYSOUT data sets. For security considerations for job cancellation, see [Controlling job modification and cancellation in z/OS JES2 Initialization and Tuning Guide](#) or [Controlling Who Can Cancel Jobs by Job Name in z/OS JES3 Initialization and Tuning Guide](#).

## **Usage considerations**

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

In addition, this request can be directed to a secondary JES subsystem. To do so, use the following URL format:

---

```
https://host:port/zosmf/restjobs/jobs/-JESB/jobname/jobid
```

---

where *JESB* is the name of the secondary JES subsystem. A request to a secondary JES subsystem must include the job name and job ID, rather than a job correlator.

## **Expected response**

The response depends on whether the request is processed synchronously or asynchronously, as follows:

- For an asynchronous request, the caller receives only the HTTP status code 202 ACCEPTED. To determine whether the request was successful, the caller can issue the service that is described in [“Obtain the status of a job”](#) on page 652.
- For a synchronous request, the caller receives an HTTP status code, which indicates the results of the request, as follows:
  - Status code 200 indicates that the synchronous request was processed successfully. However, this status does not mean that the operation was successful. To determine the success of the operation, check the "status" property in the JSON job feedback document for a value of 0 (zero). See [“Job feedback document”](#) on page 685.
  - Status code of 4nn or 5nn indicates that an HTTP error occurred.

For HTTP errors, z/OS jobs REST interface returns error information as a JSON error report document. See [“Error report document”](#) on page 688.

## **Example request**

The following request cancels the job TESTJOBW, job ID JOB00085 and purges its output on the z/OS system. With the inclusion of the **X-IBM-Job-Modify-Version** header set to 2.0, the request is eligible to be processed synchronously, if supported on the target JES subsystem.

```
DELETE /zosmf/restjobs/jobs/TESTJOBW/JOB00085 HTTP/1.1
X-IBM-Job-Modify-Version: 2.0
```

### Example response

A sample response is shown in Figure 354 on page 682. Because the request was processed synchronously by the target JES subsystem, the response body includes the job feedback document with details about the job that was cancelled.

```
HTTP/1.1 200 OK
Date: Thu, 16 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
  "jobid": "JOB00085",
  "jobname": "TESTJOBW",
  "original-jobid": "JOB00085",
  "owner": "IBMUSER",
  "member": "JES2",
  "sysname": "SY1",
  "job-correlator": "J0000085SY1.....CC20F380.....",
  "status": "0"
}
```

Figure 354. Example: Results of a job delete request

## JSON document specifications for z/OS jobs REST interface requests

This section describes the contents of the JSON documents that are used with z/OS jobs REST interface requests.

The following JSON documents are described:

- “Job document” on page 682
- “Job completed document” on page 684
- “Job feedback document” on page 685
- “Job file document” on page 686
- “Job step data document” on page 687
- “Error report document” on page 688.

### Job document

Table 379 on page 682 shows the contents of the JSON job document.

Table 379. Contents of the JSON job document	
Property	Description
<b>jobid</b>	Job ID.
<b>jobname</b>	Job name.
<b>subsystem</b>	The primary or secondary JES subsystem. If this value is null, the job was processed by the primary subsystem.
<b>owner</b>	The z/OS user ID associated with the job.

Table 379. Contents of the JSON job document (continued)

Property	Description
<b>status</b> ={ <b>INPUT</b>   <b>ACTIVE</b>   <b>OUTPUT</b> }	Job status. One of the following values:  <b>INPUT</b> Job is in input processing.  <b>ACTIVE</b> Job is running.  <b>OUTPUT</b> Job is on the hardcopy output queue.  If this value is null, the job status cannot be determined.
<b>type</b> ={ <b>JOB</b>   <b>STC</b>   <b>TSU</b> }	Job type. One of the following values:  <b>JOB</b> Batch job.  <b>STC</b> Started task.  <b>TSU</b> TSO/E user.
<b>class</b>	Job execution class.
<b>retcode</b> ={ <b>ABENDUnnnn</b>   <b>ABEND Sxxx</b>   <b>CANCELED</b>   <b>CC nnnn</b>   <b>CONV ABEND</b>   <b>CONV ERROR</b>   <b>JCL ERROR</b>   <b>SEC ERROR</b>   <b>SYS FAIL</b> }	Job completion code. One of the following values:  <b>ABENDUnnnn</b> Job ended with the user abend code <i>nnnn</i> .  <b>ABEND Sxxx</b> Job ended with the system abend code <i>xxx</i> .  <b>CANCELED</b> Job was canceled.  <b>CC nnnn</b> Job ended with the completion code <i>nnnn</i> .  <b>CONV ABEND</b> Converter ended abnormally when processing the job.  <b>CONV ERROR</b> Converter error when processing the job.  <b>JCL ERROR</b> Job encountered a JCL error.  <b>SEC ERROR</b> Job failed a security check.  <b>SYS FAIL</b> System failure.  If this value is null, the job was not yet completed.
<b>url</b>	Resource URL based on original HTTP request.
<b>files-url</b>	Resource URL for listing the spool files for the job.
<b>job-correlator</b>	Job correlator. If this value is null, the job was submitted to JES3.
<b>phase</b>	Job phase. Provides a numeric indicator of the current state of the job.
<b>phase-name</b>	Job phase name. Provides a text description of the specific phase of the job.

Table 379. Contents of the JSON job document (continued)

Property	Description
<b>step-data</b>	Step data information. Provides information about each step in the job, such as the step name, step number, and completion code. For more information, see “Job step data document” on page 687.
<b>exec-system</b>	System name of the z/OS system on which the job ran (up to 8 characters).
<b>exec-member</b>	Member name of the z/OS system on which the job ran (up to 8 characters).
<b>exec-submitted</b>	Time when the job was submitted to run (the input end time).
<b>exec-started</b>	Time when job execution started.
<b>exec-ended</b>	Time when job execution ended.
<b>reason-not-running</b>	Text identifying one or more reasons why the job is not running.

## Job completed document

Table 380 on page 684 shows the contents of the JSON job completed document.

Table 380. Contents of the JSON job completed document

Property	Description
<b>job-correlator</b>	Job correlator. If this value is null, the job was submitted to JES3.
<b>jobid</b>	Job ID.
<b>jobname</b>	Job name.
<b>owner</b>	The z/OS user ID associated with the job.
<b>class</b>	Job execution class.
<b>retcode={ ABENDUnnnn   ABEND Sxxx   CANCELED   CC nnnn   CONV ABEND   CONV ERROR   JCL ERROR   SEC ERROR   SYS FAIL }</b>	<p>Job completion code. One of the following values:</p> <p><b>ABENDUnnnn</b> Job ended with the user abend code <i>nnnn</i>.</p> <p><b>ABEND Sxxx</b> Job ended with the system abend code <i>xxx</i>.</p> <p><b>CANCELED</b> Job was canceled.</p> <p><b>CC nnnn</b> Job ended with the completion code <i>nnnn</i>.</p> <p><b>CONV ABEND</b> Converter ended abnormally when processing the job.</p> <p><b>CONV ERROR</b> Converter error when processing the job.</p> <p><b>JCL ERROR</b> Job encountered a JCL error.</p> <p><b>SEC ERROR</b> Job failed a security check.</p> <p><b>SYS FAIL</b> System failure.</p>

Table 380. Contents of the JSON job completed document (continued)	
Property	Description
<b>completion-type</b>	Specific completion type: <b>0</b> No completion information was received. <b>1</b> Job ended normally. <b>2</b> Job ended with a completion code. <b>3</b> Job encountered a JCL error. <b>4</b> Job was canceled. <b>5</b> Job abended. <b>6</b> Converter error when processing the job. <b>7</b> Job encountered a security error. <b>8</b> Job failed in EOM. <b>9</b> Job failed a security check. <b>10</b> System failure.
<b>completion-code</b>	Completion code. Set for completion-type values 1 and 2. Otherwise, null.
<b>abend-code</b>	Job completed with abend code. Set for completion-type values 5 and 8. Otherwise, null. When set, one of the following values: <b>Unnnn</b> Job ended with the user abend code <i>nnnn</i> . <b>Sxxx</b> Job ended with the system abend code <i>xxx</i> .

## Job feedback document

Table 381 on page 685 shows the contents of the JSON job feedback document.

Table 381. Contents of the JSON job feedback document	
Property	Description
<b>jobid</b>	Job ID.
<b>jobname</b>	Job name.
<b>original-jobid</b>	Original job ID. If the job was processed on another system, this value represents the original job identifier that was assigned when the job was submitted on the host system. If the target system cannot assign the original job identifier, the target system assigns a new ID to the job, which is indicated as "jobid" in this document.

<i>Table 381. Contents of the JSON job feedback document (continued)</i>	
Property	Description
<b>owner</b>	z/OS user ID associated with the job.
<b>member</b>	JES2 multi-access spool (MAS) member name.
<b>sysname</b>	z/OS system name.
<b>job-correlator</b>	Job correlator. If this value is null, the job was submitted to JES3.
<b>status={n}</b>	job processing status. If set to zero (0), the request was processed successfully. Otherwise, there was an error. See the message property for a description of the error.
<b>internal-code</b>	If job processing status indicates an error (a value other than 0), this property contains the internal service routine return code. Otherwise, this property is omitted.
<b>message</b>	If job processing status indicates an error (a value other than 0), this property contains a description of the error. Otherwise, this property is omitted.

## Job file document

Table 382 on page 686 shows the contents of the JSON job file document.

<i>Table 382. Contents of the JSON job file document</i>	
Property	Description
<b>jobname</b>	Job name.
<b>recfm</b>	<p>Record format of the file. The first character of the returned string is one of the following:</p> <p><b>F</b> Fixed length records</p> <p><b>V</b> Variable length records</p> <p><b>U</b> Undefined length records.</p> <p>One or more subsequent characters might also be present in the returned string (in this order):</p> <p><b>B</b> File has blocked records.</p> <p><b>S</b> File has standard records (if fixed length format) or spanned records (if variable length format).</p> <p><b>M</b> File has machine print-control characters.</p> <p><b>A</b> File has ASA (ANSI) print-control characters.</p> <p>Generally, the B (blocked) and S (standard or spanned) characters are not present for JES spool files. Also, the M (machine) and A (ASA) characters are mutually exclusive.</p>
<b>byte-count</b>	Number of bytes on spool consumed by the spool file. The value can be zero (0). This field is integer data type.

Table 382. Contents of the JSON job file document (continued)

Property	Description
<b>record-count</b>	Number of records in the spool file. The value can be zero (0). This field is integer data type.
<b>job-correlator</b>	Job correlator. If this value is null, the job was submitted to JES3.
<b>class</b>	Class that is assigned to the spool file.
<b>jobid</b>	Job ID.
<b>id</b>	Data set number (key). This field is integer data type.
<b>ddname</b>	DDNAME for the data set creation.
<b>records-url</b>	Resource URL for retrieving the spool file contents for the job.
<b>lrecl</b>	Specifies the length, in bytes, for fixed-length records and the maximum length for variable-length records.
<b>subsystem</b>	The primary or secondary JES subsystem. If the value is null, the job was processed by the primary subsystem.
<b>stepname</b>	Step name for the step that created this data set. The value can be null.
<b>procstep</b>	Procedure name for the step that created this data set. The value can be null.

## Job step data document

Table 383 on page 687 shows the contents of the JSON job step data document.

Table 383. Contents of the JSON job step data document

Property	Description
<b>active</b>	Value is set to <i>true</i> if the step is running. Otherwise, the value is <i>false</i> .
<b>smfid</b>	The SMF ID of the system where the step is running.
<b>step-number</b>	Step number.
<b>selected-time</b>	Date and time the step started. Not returned if the step is still active.
<b>owner</b>	The z/OS user ID associated with the job.
<b>program-name</b>	Name of the program to be run by the job step. This value is retrieved from the EXEC statement.
<b>step-name</b>	Name specified for the step on the EXEC statement.
<b>path-name</b>	Path to the program in the z/OS UNIX System Services (z/OS UNIX) file system that is run by the job step. Not returned if the step is active.
<b>substep-number</b>	Step number in the sequence of steps that run z/OS UNIX programs. Not returned if the step is still active.
<b>end-time</b>	Date and time the step completed. Not returned if the step is still active.
<b>proc-step-name</b>	Name of the procedure to be run by the job step. This value is retrieved from the EXEC statement.

Table 383. Contents of the JSON job step data document (continued)	
Property	Description
<b>completion={ ABENDUnnnn   ABEND Sxxx   CANCELED   CC nnnn   FLUSHED }</b>	<p>Step completion code. One of the following values:</p> <p><b>ABENDUnnnn</b> Step ended with the user abend code <i>nnnn</i>.</p> <p><b>ABEND Sxxx</b> Step ended with the system abend code <i>xxx</i>.</p> <p><b>CANCELED</b> Step was canceled.</p> <p><b>CC nnnn</b> Step ended with the completion code <i>nnnn</i>.</p> <p><b>FLUSHED</b> Step was not processed.</p> <p>Not returned if the step is active.</p>
<b>abend-reason-code</b>	Abend reason code. Returned if the step is completed and an abend occurs. Not returned if the step is active.

## Error report document

Table 384 on page 688 shows the contents of the JSON error report document.

Table 384. Contents of the JSON error report document	
Property	Description
<b>category</b>	Error category. This field is integer data type.
<b>rc</b>	Return code. This field is integer data type.
<b>reason</b>	Reason code. This field is integer data type.
<b>message</b>	Message that describes the error.
<b>details</b>	(Optional) Array of strings containing additional message details.

For the meanings of the category, return code, and reason code fields, see [“Error reporting categories” on page 688](#).

## Error reporting categories

This section describes the error categories and associated error codes that can be returned in the JSON error report document, described in [“Error report document” on page 688](#).

### Categories

Table 385 on page 688 shows the error categories that are defined for errors that are returned in z/OS jobs REST interface operations.

Table 385. Error categories for z/OS jobs REST interface operations			
Category	Ordinal Value	Description	Where the error details are described
<b>Dynalloc</b>	1	Dynamic allocation errors.	<a href="#">“Category 1 — Dynamic allocation error” on page 689</a>

Table 385. Error categories for z/OS jobs REST interface operations (continued)

Category	Ordinal Value	Description	Where the error details are described
<b>VSAM API</b>	3	Errors that are produced or detected by the Java/ JNI/ C/ HLASM/ VSAM layer.	<a href="#">“Category 3 – VSAM API error” on page 691</a>
<b>VSAM system</b>	4	Errors that are produced or detected by VSAM. The return code and reason code are VSAM-specific.	<a href="#">“Category 4 – VSAM system error” on page 691</a>
<b>VSAM ABEND</b>	5	ABEND information that results from VSAM failures.	<a href="#">“Category 5 – VSAM ABEND error” on page 691</a>
<b>Service</b>	6	Errors that are produced or detected in the service layer.	<a href="#">“Category 6 – Service error” on page 691</a>
<b>Unexpected</b>	7	Unexpected errors detected.	<a href="#">“Category 7 – Unexpected error” on page 697</a>
<b>SSI extended status</b>	8	Errors that are produced or detected by the extended status function call of the subsystem interface (SSI Function Code 80).	<a href="#">“Category 8 – SSI extended status error” on page 697</a>
<b>CIM</b>	9	Errors that are produced or detected by the Common Information Model (CIM) interface.	<a href="#">“Category 9 – Common Information Model (CIM) error” on page 697</a>
<b>SSI job modify</b>	10	Errors that are produced or detected by the job modify function call of the subsystem interface (SSI Function Code 85).	<a href="#">“Category 10 – SSI job modify error” on page 698</a>

## Category 1 – Dynamic allocation error

[Table 386 on page 690](#) shows the possible conditions for this error category.

Table 386. Category 1 errors

Return code (rc)	Reason	Message	Description
<b>n</b>	0	Error allocating internal reader, RC=%d (0x%08X)	<p>An error occurred when z/OS attempted to allocate the internal reader for job submission. In the message, <i>RC</i> is error data from the dynamic allocation request (SVC 99).</p> <p>To diagnose the error, convert the <i>RC</i> value from decimal to a 4-byte hexadecimal value, which provides the dynamic allocation error code and information code, as follows:</p> <ul style="list-style-type: none"> <li>High-order two bytes indicate the error code from the dynamic allocation request (field S99ERROR in the input request block S99RB).</li> <li>Low order two bytes indicate the information code from the dynamic allocation request (field S99INFO in the input request block S99RB).</li> </ul> <p>For information about dynamic allocation and its related error and information codes, see <i>Dynamic allocation in z/OS MVS Programming: Authorized Assembler Services Guide</i>.</p>
<b>n</b>	1	Error allocating input data set: %s, RC=%d (0x%08X)	<p>An error occurred when z/OS attempted to allocate a ddname for the data set specified as the source for the input job. In the message, <i>RC</i> is the return code from the BPXWDYN service.</p> <p>For information about BPXWDYN and its related return codes, see <i>BPXWDYN: A text interface to dynamic allocation and dynamic output in z/OS Using REXX and z/OS UNIX System Services</i>.</p>
<b>n</b>	2	Error allocating spool file: job '%s' spool file id %d, RC=%d (0x%08X)	<p>An error occurred when z/OS attempted to allocate the requested spool file. Perhaps, a thread is attempting to allocate the spool file while another thread is requesting to cancel the job and purge its output.</p> <p>In the message, <i>RC</i> is error data from the dynamic allocation request (SVC 99). Both decimal and hexadecimal values are provided in the message.</p> <p>The hexadecimal value provides the dynamic allocation error code and information code, as follows:</p> <ul style="list-style-type: none"> <li>High-order two bytes indicate the error code from the dynamic allocation request (field S99ERROR in the input request block S99RB).</li> <li>Low order two bytes indicate the information code from the dynamic allocation request (field S99INFO in the input request block S99RB).</li> </ul> <p>For information about dynamic allocation and the meanings of the error code and information code, see <i>Dynamic allocation in z/OS MVS Programming: Authorized Assembler Services Guide</i>.</p>

### Category 3 – VSAM API error

Table 387 on page 691 shows the possible conditions for this error category.

Table 387. Category 3 errors			
Return code (rc)	Reason	Message	Description
4	1	Incorrect JesVsam handle	
4	2	VSAM file is not open	
4	3	Record length %d > lrecl %d	Writing a record to a VSAM file failed because an incorrect record length was specified.
4	4	Could not write JCL to internal reader	An I/O exception occurred when JCL to the internal reader.
8	0	JesVsam get failed	Buffer too small to hold the VSAM record.
255	0	JesVsam native buffer malloc failed	

### Category 4 – VSAM system error

Table 388 on page 691 shows the possible conditions for this error category.

Table 388. Category 4 errors			
Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	For descriptions of the specific return and reason codes, see the VSAM publications.

### Category 5 – VSAM ABEND error

Table 389 on page 691 shows the possible conditions for this error category.

Table 389. Category 5 errors			
Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	The values <i>n</i> and <i>m</i> indicate the ABEND return code and reason code.

### Category 6 – Service error

Table 390 on page 692 shows the possible conditions for this error category.

Table 390. Category 6 errors

<b>Return code (rc)</b>	<b>Reason</b>	<b>Message</b>	<b>Description</b>
4	1	Incorrect Internal Reader mode: %s. Must be one of TEXT   RECORD   BINARY	Request header X-IBM-Intrdr-Mode specified a value that is not valid. Valid values are TEXT, BINARY, or RECORD.
4	2	Incorrect Internal Reader parameters: %s. Fixed records are required for binary mode	The internal reader characteristics form a combination that is not valid. If you specify the value BINARY for the X-IBM-Intrdr-Mode request header, you must specify the value F for the X-IBM-Intrdr-Recfm request header.
4	3	Request does not contain '%s' content	Job modification requests must have a content type of application/json.
4	4	Value of %s query parameter is not valid	The query parameter that was identified in the message either contains incorrect characters or exceeds the allowable length. In the message, the query parameter is <i>owner</i> , <i>prefix</i> , <i>jobid</i> , or <i>job-correlator</i> .
4	5	Update request is not 'cancel'	In the job modification request, the "request" property is set to an incorrect value. The required value is "cancel."
4	6	Request does not contain a valid job update request	In the job modification request, the input document does not specify a valid property. The valid properties are: <ul style="list-style-type: none"> <li>• "request"</li> <li>• "class"</li> </ul>
4	7	No match for method %s and pathInfo='%s'	The supplied servlet path information (pathinfo) does not match any expected string for the HTTP method that was specified.
4	8	POST requests not supported	For standard REST requests, the POST HTTP method is not allowed. To avoid this message, include the X-IBM-Requested-Method header to send the request through the POST verb.
4	9	Job submission error. Record length %d too long for JCL submission, maxlen=%d	The check for record mode job submission failed.
4	10	No job found for reference: '%s'	The job modification request specifies a job that does not exist.

Table 390. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	11	Record range '%s' is not valid for spool file record request	Request header X-IBM-Record-Range specified a value that is not valid. The content range must be specified by using one the following formats:  <b>SSS-EEE</b> Where <i>SSS</i> identifies the start record and <i>EEE</i> identifies the end record to be retrieved. Both values are relative offsets (0-based). When <i>EEE</i> is set to 0, records through the end of the spool file are retrieved.  <b>SSS,NNN</b> Where <i>SSS</i> identifies the start record and <i>NNN</i> identifies the number of records to be retrieved.
4	12	Job '%s' does not contain spool file id %d	
4	13	Job input was not recognized by system as a job	The job was submitted without a job statement or with unrecognized (non-JCL) content.
4	14	Unsupported encoding: %s	On a job submission request, the Content-Type request header specified an unsupported character set (charset).
4	15	DD names are not supported for submit input	The filename property in the JSON document started with //DD:, indicating the dd:ddname syntax. This syntax is not supported.
4	16	Data set not found	The data set specified in the JSON document that was provided on the submit job interface was not found. Possibly, the data set is not cataloged.
4	17	Submit input data does not start with a slash	This error occurs when the first character of the input job is not the EBCDIC slash character. Possible causes include: <ul style="list-style-type: none"><li>• The Content-Type request header is set to text/plain when a JSON document that names the source of the input job is also used.</li><li>• The input data set or file does not contain EBCDIC data.</li></ul>
4	18	Submit input filename must be absolute path: %s	The z/OS UNIX file specification in the JSON document was not an absolute path.
4	19	Internal reader mode must be RECORD for data set submission: %s	If specified, the internal reader mode must be set to RECORD when a job is submitted from a data set.
4	20	Service not implemented: %s	The requested service is not supported. The variable text %s contains additional information.

Table 390. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	22	Internal reader RECFM (%s) does not match data set RECFM (%s): %s	If specified, the internal reader record format must match the record format of the existing data set when a job is submitted from a data set.
4	23	Internal reader LRECL (%d) does not match data set LRECL (%d): %s	If specified, the internal reader logical record length must match the logical record length of the existing data set when a job is submitted from a data set.
4	24	Content type '%s' not valid for internal reader mode '%s'	The values that are specified for Content-Type and internal reader mode are not a supported combination.
4	25	JCL symbol name '%s' is not valid	The specified symbol name does not match the syntax rules for a JCL symbol or start with the characters 'SYS'.
4	26	JCL symbol '%s' value exceeds maximum length	The value that is supplied for the specified symbol name exceeds the maximum value length of 255 characters.
4	27	No value supplied for JCL symbol '%s'	For a JCL symbol to be defined, it must have a non-null, non-blank value.
4	28	Maximum number of JCL symbols exceeded	An attempt was made to define more than 128 symbols, which is not allowed.
4	29	User correlator '%s' is not valid	The specified user correlator (X-IBM-User-Correlator) does not match the syntax rules for a user correlator.
4	30	Notification URL '%s' exceeds maximum length	The specified notification URL exceeds the maximum value of 2083 characters.
4	31	Request header not supported by primary JES subsystem: %s	The request header that is identified in the message is not supported by the primary job entry subsystem.
4	32	Error parsing JSON input	<p>On a job submission request, the header Content-Type specified application/json. However, an exception occurred when the input JSON document was processed.</p> <p>Possible causes include:</p> <ul style="list-style-type: none"> <li>• JCL stream was provided instead of a JSON document.</li> <li>• JSON document was malformed.</li> <li>• Required "file" property was not provided.</li> <li>• Value for the "recall" property was not valid.</li> </ul> <p>For details about the exception, check the z/OSMF logs.</p>

Table 390. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
4	33	Data set is migrated: %s	The JSON document specifies a z/OS data set that was migrated. No recall was issued. No job was submitted.
4	34	Recall issued for migrated data set: %s	The JSON document specifies a z/OS data set that was migrated. A recall without waiting was issued. No job was submitted.
4	35	Error recalling data set, RC=%d	An error occurred during the recall of a migrated data set. The return code from the ARCHRCAL service is included in the message.  For information about the ARCHRCAL service and return codes from the DFSMSHsm user macros, see <a href="#">ARCHRCAL: Recalling a data set in z/OS DFSMSHsm Managing Your Own Data</a> .
4	36	Incorrect internal reader class: %s. Must be one character in length.	The internal reader class request header specified a value that is not valid. The class must be one character in length.
4	37	Incorrect job update version requested: %s.	In the job modification request, the "version" property or the X-IBM-Job-Modify-Version request header is set to an incorrect value. The valid values are "1.0" or "2.0".
4	38	The query parameters <i>search</i> and <i>research</i> cannot be used together.	The search and research query parameters are mutually exclusive; they cannot be used together. Specify either search or research, but not both.
4	39	The query parameters <i>search</i> and <i>research</i> are allowed only on <i>mode=text</i> requests.	The query parameters search and research cannot be used when mode is set to a value other than text, such as mode=record or mode=binary.
4	40	The query parameter maxreturnsize is not a valid integer.	The query parameter maxreturnsize must be set to a valid integer value, for example,maxreturnsize=100.
8	1	Unable to query information about submitted job: %s	The job status for the submitted job was not obtained within the timeout period (3 seconds).
8	2	EOF encountered before all requested bytes read (%d / %d)	Internal read state error. The expected number of bytes were not available to be read before the end of file (EOF) was reached.

Table 390. Category 6 errors (continued)

<b>Return code (rc)</b>	<b>Reason</b>	<b>Message</b>	<b>Description</b>
<b>8</b>	3	Range start is beyond end of spool file %d for job %s	In a GET request for a range of records for a spool file, the X-IBM-Record-Range header specified a record start value that is beyond the end of the spool file.
<b>8</b>	4	Cannot advance spool file more than Integer.MAX_VALUE. DD= %s	In a GET request for a range of records in a spool file, the X-IBM-Record-Range header specified a record start value that was greater than $2^{31}-1$ .
<b>8</b>	5	Error opening input data set: %s	An error occurred opening the input z/OS data set. For a message with additional information, check the z/OSMF logs.
<b>8</b>	6	Error reading submit input data	An error occurred reading the submit input data. For a message with additional information, check the z/OSMF logs.
<b>8</b>	7	Error opening input file: %s	An error occurred opening the input z/OS UNIX file. For a message with additional information, check the z/OSMF logs.
<b>8</b>	8	IAZSYMBL error defining %s	The JES symbol definition service (IAZSYMBL) failed during the attempt to define the specified information. In the message, %s is one of the following values: <ul style="list-style-type: none"> <li>• User correlator</li> <li>• Notification URL</li> <li>• One or more JCL symbols.</li> </ul> For details about the IAZSYMBL error, check the z/OSMF logs.
<b>12</b>	1	Not authorized to access spool file	An authorization check failed during the attempt to open the requested spool file.
<b>12</b>	2	Not authorized to submit job	An authorization check failed during the attempt to open the internal reader to submit a job.

Table 390. Category 6 errors (continued)

Return code (rc)	Reason	Message	Description
12	3	User not authorized to issue a CIM request	<p>CIM detected an authentication or authorization failure during the request. To use the requested service, the user must be authorized to use the CIM server and be permitted to the JES2-JES3Jobs CIM provider.</p> <p>The requested service was one of the following:</p> <ul style="list-style-type: none"> <li>• Hold a job.</li> <li>• Release a job.</li> <li>• Change the job class.</li> <li>• Cancel a job.</li> <li>• Delete a job.</li> </ul> <p>CIM provides jobs (CFZSEC and CFZRCUST) to help you configure the CIM server, including security authorizations and file system customization. For more information, see <a href="#">Quick guide: CIM server setup and verification in z/OS Common Information Model User's Guide</a>.</p>

## Category 7 – Unexpected error

Table 391 on page 697 shows the possible conditions for this error category.

Table 391. Category 7 errors

Return code (rc)	Reason	Message	Description
16	1	Server error occurred	For details about the exception, check the z/OSMF logs.

## Category 8 – SSI extended status error

Table 392 on page 697 shows the possible conditions for this error category.

Table 392. Category 8 errors

Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	The return and reason codes ( <i>n,m</i> ) are set from the extended status function of subsystem interface (SSI) Function Code 80 return code and the subsystem options block (SSOB) return code. The details property of the JSON error report document contains a message with more information. See <a href="#">“Error report document” on page 688</a> .

## Category 9 – Common Information Model (CIM) error

Table 393 on page 698 shows the possible conditions for this error category.

Table 393. Category 9 errors			
Return code (rc)	Reason	Message	Description
4	2	Incorrect jobname: "%s"	Before the CIM service call, the job name was found to be null or an empty string.
4	3	Incorrect jobid: "%s"	Before the CIM service call, the job ID was found to be null or an empty string.
4	4	Incorrect JES type	Before the CIM service call, an incorrect JES type (not JES2 or JES3) was detected.
4	5	Incorrect job class: "%s"	Before the CIM service call, the job class was found to be null or an empty string.
8	—	Varies	CIM internal error. An error occurred during setup or invocation of the CIM service.
12	<i>m</i>	Error returned from CIM job {Cancel Hold Release Request Property Change}service	CIM response error. Reason ( <i>m</i> ) is the reason code that was returned from CIM. The "details" property of the JSON error report document contains the CIM response text, if any. See <a href="#">“Error report document” on page 688</a> .
16	—	CIM connection failure	<p>A connection exception was encountered when the request was processed. This error can occur during periods of concurrent high usage of the REST interfaces. Usually, the reason for the failure is a connection refused due to overload of the server. The application can try the request again. The number of retry attempts that are needed depends on how much work is being requested of the server.</p> <p>One of the following services was requested:</p> <ul style="list-style-type: none"> <li>• Hold a job.</li> <li>• Release a job.</li> <li>• Change the job class.</li> <li>• Cancel a job.</li> <li>• Delete a job.</li> </ul>

## Category 10 – SSI job modify error

Table 394 on page 698 shows the possible conditions for this error category.

Table 394. Category 10 errors			
Return code (rc)	Reason	Message	Description
<i>n</i>	<i>m</i>	Varies	The return and reason codes ( <i>n,m</i> ) are set from the job modify function call of the subsystem interface (SSI Function Code 85) return code and the subsystem options block (SSOB) return code. The "details" property of the JSON error report document contains a message with more information. See <a href="#">“Error report document” on page 688</a> .

## z/OSMF information retrieval service

---

The z/OSMF information retrieval service is an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. This service allows the caller to query the version and other details about the instance of z/OSMF running on a particular system.

z/OSMF information includes the following details:

- SAF realm
- z/OSMF listening port
- z/OSMF version and release
- Installed plug-ins and plug-in build levels
- Indicates the z/OS operating system level.

With this information, a calling program can determine which z/OSMF plug-ins and API functions are available for use on a given system. For information, see [“Retrieve z/OSMF information” on page 700](#).

### Using the Swagger interface

You can use the Swagger interface to display information about the z/OSMF information retrieval service REST API. For more information, see [“Using the Swagger interface” on page 1](#).

### Required authorizations

None.

### Error handling

For errors that occur during the processing of a request, the API returns an appropriate hypertext transfer protocol (HTTP) status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

#### **HTTP 200 OK**

Success.

#### **HTTP 400 Bad request**

Request contained incorrect parameters.

#### **HTTP 401 Unauthorized**

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the information retrieval service.

#### **HTTP 500 Internal server error**

Programming error.

### Error logging

Errors from the information retrieval service are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Retrieve z/OSMF information

You can use this operation to retrieve information about z/OSMF on a particular z/OS system.

### HTTP method and URI path

---

```
GET /zosmf/info
```

---

where **zosmf/info** identifies the z/OSMF information retrieval service.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Content type used for HTTP response data

The JSON content type ("Content-Type: application/json") is used for response data. The following JSON object is received as output from the request.

```
{
  "zosmf_saf_realm": "SAF-profile-prefix",
  "zosmf_port": "zosmf-server-port-number",
  "zosmf_full_version": "zosmf-release-level",
  "plugins": [
    {
      "pluginVersion": "plugin-fmid-build-level",
      "pluginStatus": "plugin-status",
      "pluginDefaultName": "plugin-name"
    },
    {
      "pluginVersion": "plugin-fmid-build-level",
      "pluginStatus": "plugin-status",
      "pluginDefaultName": "plugin-name"
    }
  ],
  "api_version": "api-version",
  "zos_version": "zos-release",
  "zosmf_version": "zosmf-version",
  "zosmf_hostname": "host-system-URL"
}
```

where:

#### **zosmf\_saf\_realm**

Realm associated with the system on which z/OSMF is installed. Usually, this is the sysplex name.

#### **zosmf\_port**

Port number for SSL encrypted traffic for the active instance of z/OSMF on the z/OS system.

#### **zosmf\_full\_version**

Indicates the z/OSMF version, further qualified by a service level.

#### **plugins**

Array of zero, one, or more elements that contain information about each of the installed z/OSMF plug-ins. If no plug-ins are installed, this area is empty.

Each element contains the following attributes:

**pluginVersion**

Indicates the plug-in version (FMID) and build level.

**pluginStatus**

Indicates the status of the plug-in. The status is reported for IBM-supplied plug-ins only. For an external application, the status is blank.

The following values are valid:

**ACTIVE**

The plug-in is running.

**INSTALLED**

The plug-in is installed, but not running.

**UNINSTALLED**

The plug-in was installed in a previous z/OSMF configuration, but is not installed in the current configuration. This status can result when a plug-in is removed from z/OSMF.

After a plug-in is started, its status remains as ACTIVE, even if the plug-in is later stopped.

**pluginDefaultName**

Indicates the plug-in name.

**api\_version**

Version of the z/OSMF information retrieval service and the JSON object structure used for this request. The version sequence starts at 1, and is incremented if the service or the JSON structure changes.

**zos\_version**

Indicates the z/OS operating system level. The following values are valid:

**04.24.00**

Indicates that the z/OS level is V2R1.

**04.25.00**

Indicates that the z/OS level is V2R2.

**04.26.00**

Indicates that the z/OS level is V2R3.

**zosmf\_version**

Indicates the z/OSMF level. The following values are valid:

**24**

Indicates that the z/OSMF level is V2R1.

**25**

Indicates that the z/OSMF level is V2R2.

**26**

Indicates that the z/OSMF level is V2R3.

**zosmf\_hostname**

Indicates the hostname or IP address of the z/OS system on which z/OSMF is installed

**Usage considerations**

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

**Required authorizations**

None.

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred. For more details, see [“Error handling” on page 699](#).

The response also includes a JSON object that contains the retrieved data. For details, see [“Content type used for HTTP response data” on page 700](#).

## Example request

In the following example, the GET method is used to retrieve information about z/OSMF.

```
GET /zosmf/info HTTP/1.1
Host: host.name.com
```

*Figure 355. Sample request to retrieve z/OSMF information*

## Example response

For a successful request, the HTTP response includes a JSON document containing the requested information.

```
HTTP/1.1 200 OK
Date: Wed, 06 Mar 2013 06:39:28 +0000GMT
Content-Type: text/plain
Connection: close

{
  "zosmf_saf_realm": "SAFRealm",
  "zosmf_port": "443",
  "zosmf_full_version": "24.02",
  "plugins": [
    {
      "pluginVersion": "hsma210.spe2;driver05;2014-02-11T03:21:53",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Import Manager",
      "pluginVersion": "hsma213.spe2;driver05;2014-02-11T10:17:27",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "WorkloadManagement",
      "pluginVersion": "Hqx7790;driver122;2014-02-18T00:00:00Z",
      "pluginDefaultName": "IBM SDSF",
      "pluginVersion": "hsma216;DRIVER04;2014-01-14T19:03:40",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Capacity Provisioning",
      "pluginVersion": "hsma214.spe2;driver05;2014-02-11T08:28:24",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Software Deployment",
      "pluginVersion": "hsma21a;pm93903;2013-08-12T03:52:53",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "NetworkConfigurationAssistant",
      "pluginVersion": "hsma215.spe2;driver05;2014-02-11T10:16:30",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "IncidentLog",
      "pluginVersion": "hsma211.spe2;driver05;2014-02-11T03:29:52",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "ISPF",
      "pluginVersion": "hsma212;DRIVER4B;2014-01-14T12:43:43",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "ResourceMonitoring",
      "pluginVersion": "hsma217.spe2;driver05;2014-02-11T12:01:40",
      "pluginStatus": "ACTIVE",
      "pluginDefaultName": "Workflow"
    }
  ],
  "api_version": "1",
  "zos_version": "04.24.00",
  "zosmf_version": "24",
  "zosmf_hostname": "host.name.com"
}
```

## z/OSMF settings services

The z/OSMF settings services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. A set of REST services is provided for working with the z/OSMF settings for z/OSMF plug-ins and the z/OSMF server. Depending on the changes specified, the changes can be scoped to individual users, plug-ins, or the z/OSMF server.

[Table 395 on page 703](#) lists the operations that the z/OSMF settings services provide.

Table 395. Operations provided through the z/OSMF settings services

Operation	HTTP method and URI path
<b>Put</b>	PUT /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
	PUT /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
<b>Get</b>	GET /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
	GET /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
<b>Delete</b>	DELETE /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
	DELETE /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>

## Required authorizations

The user must be logged in to z/OSMF, and must have READ access to the SAF profile that was registered for the plug-in and task making the request.

For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

## Content type used for HTTP request and response data

The JSON content type ("Content-Type: application/json") is used for request and response data. The following JSON object is used by all z/OSMF settings services as input and output for the requested operations. The attributes that are provided in the JSON object depend on the requested operation.

```
{
  "value": "data-value",
  "version": "structure-version",
  "messages": "z/OSMF-messages",
  "update": true|false
}
```

Where:

### data-value

The value that is added, updated, retrieved, or removed by the z/OSMF settings services. Any data type is supported including JSON objects, JSON arrays, and scalars. The value is required.

### structure-version

Version of the z/OSMF settings services and the JSON object structure that is used for this request. The version sequence starts at 1.0.0, and is incremented only if the services or the JSON structure changes. The version the client supports is required as input to the request. The z/OSMF settings services are backward compatible for *n*-2 versions, and accepts requests for each version it supports. If the version specified by the client is not supported or if no version is specified, the service returns an error message.

### z/OSMF-messages

z/OSMF messages received during the request. The *messages* attribute is included in the JSON object only if an error occurred during the request. The message ID and message text are provided for each z/OSMF message received.

### update

An optional input attribute, which indicates that the service is updating or replacing an existing JSON object. If you set the value to *true*, the service updates the key-value pairs you specified for the *value* attribute and preserves any other data persisted in the JSON object. You can set this attribute to *true* only when the data type is a JSON object or JSON array. If you omit this attribute or set it to *false*, the service deletes the existing JSON object and creates a new JSON object that contains only the key-value pairs you specified for the *value* attribute.

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a *4nn* code or a *5nn* code. Some errors might also include a returned JSON object that contains a message that describes the error.

The following HTTP status codes are valid:

### HTTP 200 OK

Success.

### HTTP 400 Bad request

Request contained incorrect parameters.

### HTTP 401 Unauthorized

Submitter of the request did not authenticate to z/OSMF or is not authorized to use the z/OSMF settings services.

### HTTP 404 Bad URL

Target of the request (a URL) was not found.

### HTTP 500 Internal server error

Programming error.

## Error logging

Errors from the z/OSMF settings services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required.

For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Persist z/OSMF settings for a user or plug-in

You can use this operation to persist z/OSMF settings for a user, z/OSMF plug-in, or the z/OSMF server.

### HTTP method and URI path

```
PUT /zosmf/settings/system/<resourcePath>?version=1.0.0
PUT /zosmf/settings/user/system/<resourcePath>?version=1.0.0

PUT /zosmf/settings/zosmf/<resource-path>?version=1.0.0
PUT /zosmf/settings/user/zosmf/<resource-path>?version=1.0.0

PUT /zosmf/settings/app/<plugin-id>/<task-id>/<resource-path>?version=1.0.0
PUT /zosmf/settings/user/<plugin-id>/<task-id>/<resource-path>?version=1.0.0
```

Where:

- **/zosmf/settings/** identifies the z/OSMF settings services.
- **user** indicates that the service persists the data only for the user who is logged in to z/OSMF when the service is invoked.
- **app** indicates that the service persists the data globally for the application.
- **<pluginId>** is the unique identifier that you assigned to the plug-in.
- **<taskId>** is the unique identifier that you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object by using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

## Query parameters

None.

## Standard headers

Use the following standard HTTP headers with this request:

```
Accept: application/json
Content-Type: application/json
```

## Custom headers

None.

## Request content

Your request must include a JSON object that contains the value to be persisted and the version. For more details, see [“Content type used for HTTP request and response data” on page 366](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 366](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error occurred. For more details, see [“Error handling” on page 484](#).

The response also includes a JSON object that contains the current data after being modified. For more details, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

In this example, assume that you want to enable the z/OSMF services "z/OS jobs REST interface" and "z/OS data set and file REST interface." Also, you want to disable the Sysplex Management plug-in. To persist data that satisfies this criteria, submit the request that is depicted in [Figure 356 on page 705](#):

```
PUT /zosmf/settings/zosmf/services?version=1.0.0
Host: zosmf1.yourco.com
Accept: application/json
Content-Type: application/json
{
  "value": {
    "IZU_REST_FILE": "Y",
    "IZU_SYSPLEX_CONFIGURE": "N",
    "IZU_REST_JOB": "Y"
  }
}
```

*Figure 356. Sample request to persist user-specific z/OSMF settings data*

## Retrieve z/OSMF settings data

You can use this operation to retrieve z/OSMF settings data that is persisted for a specific user or application.

### HTTP method and URI path

---

```
GET /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
GET /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
```

---

Where:

- **/zosmf/settings/** identifies the z/OSMF settings services.
- **user** indicates that the service persists the data only for the user who is logged in to z/OSMF when the service is invoked.
- **app** indicates that the service persists the data globally for the application.
- **<pluginId>** is the unique identifier that you assigned to the plug-in.
- **<taskId>** is the unique identifier that you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object by using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

### Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

### Custom headers

None.

### Request content

None.

### Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

### Required authorizations

See [“Required authorizations” on page 703](#).

### Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For more information, see [“Error handling” on page 484](#).

The response also includes a JSON object that contains the retrieved data. For more information, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

In this example, assume that you want to query the enablement status of the z/OSMF services and plug-ins on your system. To retrieve the status, submit a request like the one that is shown in [Figure 357](#) on page 707:

```
GET: /zosmf/settings/zosmf/services?version=1.0.0
Host: zosmf1.yourco.com
```

*Figure 357. Sample request to retrieve z/OSMF settings data*

A sample response is shown in [Figure 358](#) on page 707. In the example, the enablement status is returned for the following z/OSMF services:

- Sysplex Management plug-in is disabled.
- z/OS jobs REST interface is enabled.
- z/OS data set and file REST interface is enabled.

```
HTTP/1.1 200 OK
Date: Mon, 27 July 2020 05:39:28 +0000GMT
Connection: close
```

```
{
  "version": "1.0.0",
  "value": {
    "IZU_SYSPLEX_CONFIGURE": "N",
    "IZU_REST_FILE": "Y",
    "IZU_REST_JOB": "Y"
  }
}
```

*Figure 358. Sample response from a request to retrieve z/OSMF settings data*

## Delete z/OSMF settings data

You can use this operation to delete z/OSMF settings data that is persisted for a specific user or application.

### HTTP method and URI path

```
DELETE /zosmf/settings/user/<pluginId>/<taskId>/<resourcePath>
DELETE /zosmf/settings/app/<pluginId>/<taskId>/<resourcePath>
```

Where:

- **/zosmf/settings/** identifies the z/OSMF settings services.
- **user** indicates that the service persists the data only for the user who is logged in to z/OSMF when the service is invoked.
- **app** indicates that the service persists the data globally for the application.
- **<pluginId>** is the unique identifier that you assigned to the plug-in.
- **<taskId>** is the unique identifier that you assigned to the task.
- **<resourcePath>** is the path in the JSON object to the attribute where you want the data to be stored. The persisted data is stored in a JSON object by using a tree structure. To persist data, specify all the nodes or branches that must be traversed in the JSON structure to access that data. Use a forward slash (/) to separate each node or branch, and specify the nodes in the order in which they are listed in the structure.

## Standard headers

Use the following standard HTTP header with this request:

Content-Type: application/json

## Custom headers

None.

## Request content

None.

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Required authorizations

See [“Required authorizations” on page 703](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 indicates success. A status code of *4nn* or *5nn* indicates that an error occurred. For more information, see [“Error handling” on page 484](#).

The response also includes the updated JSON object. For more information, see [“Content type used for HTTP request and response data” on page 483](#).

## Example

In this example, assume that you want to delete the persisted data for the z/OS jobs REST interface. These z/OSMF services are represented by the IZU\_REST\_JOB key under the /services portion of the request URI. To delete the persisted data for this service, submit a request like the one shown in [Figure 359 on page 708](#).

```
DELETE /zosmf/settings/zosmf/services/IZU_REST_JOB?version=1.0.0
Host: zosmf1.yourco.com
```

*Figure 359. Sample request to delete persisted z/OSMF settings data*

A sample response is shown in [“Delete z/OSMF settings data” on page 707](#). On completion, the delete request returns the value of "IZU\_REST\_JOB" key, which is now null.

```
HTTP/1.1 200 OK
Date: Mon, 27 July 2020 05:39:28 +0000GMT
Connection: close
{
  "version": "1.0.0",
  "value": null
}
```

*Figure 360. Sample response from a request to retrieve z/OSMF settings data*

Following this deletion, assume that you were to query the persisted data for the z/OSMF services, as shown in [“Retrieve z/OSMF settings data” on page 706](#).

```
GET: /zosmf/settings/zosmf/services?version=1.0.0
Host: zosmf1.yourco.com
```

Figure 361. Sample request to retrieve z/OSMF settings data

The response body no longer contains data for the z/OS jobs REST interface.

```
HTTP/1.1 200 OK
Date: Mon, 27 July 2020 05:39:28 +0000GMT
Connection: close
{
  "version": "1.0.0",
  "value": {
    "IZU_SYSPLEX_CONFIGURE": "N",
    "IZU_REST_FILE": "Y"
  }
}
```

Figure 362. Sample response from a request to retrieve z/OSMF settings data

## z/OSMF system variable services

The z/OSMF system variable services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage z/OSMF system variables.

Table 396 on page 709 lists the operations that the system variable services provide.

Table 396. z/OSMF system variable services: operations summary	
Operation name	HTTP method and URI path
<a href="#">“Create or update system variables” on page 710</a>	POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
<a href="#">“Get system variables” on page 712</a>	GET /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
<a href="#">“Import system variables” on page 714</a>	POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/import
<a href="#">“Export system variables” on page 715</a>	POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/export
<a href="#">“Delete system variables” on page 717</a>	DELETE /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>

Table 397 on page 709 describes the variables that can be specified in the system variable services URI paths.

Table 397. z/OSMF system variable services: URI path variables	
URI path variable	Description
<version>	The version of the system variable services API. The following value is valid: 1.0.
<sysplex-name>	The name of the sysplex.
<system-name>	The name of the system.

## Authorization requirements

To use the system variable services API, the client must be authenticated. For more information, see [“Authenticating to z/OSMF” on page 2](#).

Also, for some of the system variable services, the client requires READ access to the ZOSMF.VARIABLES.SYSTEM.ADMIN resource profile in the ZMFAPLA class. See the description of the individual APIs for details.

## Error logging

Errors from the system variable services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For more information, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded.

### HTTP 204 No content

The request was processed successfully; however, no content was returned. This status is normal for some types of requests, such as creating or updating system variables.

### HTTP 400 Bad request

The request was missing required input, had errors in the provided input, included extraneous input, or cannot be otherwise served. Additional information regarding the error is provided in an error response body that includes a reason code with additional information. Do not repeat the request without first correcting it.

### HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

### HTTP 404 Not found

The requested resource does not exist.

### HTTP 500 Server error

A server error occurred during processing of the request.

## Create or update system variables

Use this operation to create or update z/OSMF system variables in the system variable pool.

## HTTP method and URI path

---

```
POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
```

---

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF system variables service. The following value is valid: 1.0.
- **<sysplex-name>** identifies the sysplex.
- **<system-name>** identifies the system.

## Description

This operation creates or updates system variables specified in the request body. If the system variable pool does not exist, this operation creates the pool and adds the variables to it. If there is no request body and the pool does not already exist, the operation creates an empty pool. If there is no request body and the pool already exists, no action is taken. If a variable appears in the request body multiple times, the value of the last occurrence is used as the value of the variable.

On successful completion, HTTP status code 204 (No content) is returned, indicating that the system variables were created or updated with the new value.

## Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class: ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 710](#).

## Request content

The request content is expected to contain an array of JSON objects. See Table 398 on page 711 and Table 399 on page 711. A request with no request body creates an empty pool if the pool does not already exist.

Table 398. Request content for the create or update system variables request		
Field name	Type	Description
<b>system-variable-list</b>	Array of objects	List of variables to be added or updated to the system variable pool.

Table 399. Fields in a JSON object for the create or update system variables request			
Field name	Type	Required or optional	Description
<b>name</b>	String	Required	Descriptive name for the variable
<b>value</b>	String	Required	Value for the variable
<b>description</b>	String	Optional	Description of the variable

## HTTP status codes

On successful completion, HTTP status code 204 (no content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 400. HTTP error response codes for a create or update system variables request	
HTTP error status code	Description
<b>400</b>	The request is missing a required property or the value of the property was null or empty, or the JSON request is incorrectly formatted.
<b>401</b>	Submitter of the request is not authorized to invoke the task to create the system variables.
<b>404</b>	Requested <i>sysplex-name.sysname-name</i> was not found.
<b>404</b>	Requested <i>system-name</i> was not found.

Table 400. HTTP error response codes for a create or update system variables request (continued)

HTTP error status code	Description
500	A server error occurred during processing of the request.

## Response content

None.

## Example HTTP interaction

In the following example, the POST method is used to create the system variables.

```
POST /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE
```

Figure 363. Sample request to create system variables

The request body is as follows:

```
{
  "system-variable-list": [
    {
      "name": "var1",
      "value": "value1",
      "description": "description of the variable"
    },
    {
      "name": "var2",
      "value": "value2",
      "description": "description of the variable"
    }
  ]
}
```

Figure 364. Sample request body for a create system variables request

## Get system variables

Use this operation to get the z/OSMF system variables from a selected system.

### HTTP method and URI path

```
GET /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF system variables service. The following value is valid: 1.0.
- **<sysplex-name>** identifies the sysplex.
- **<system-name>** identifies the system.

### Description

This operation retrieves system variables from the system variable pool and returns them in a list.

On successful completion, HTTP status code 200 is returned, along with a response body, which is described in [“Response content” on page 713](#).

### Request content

None.

### Authorization requirements

For general requirements, see [“Authorization requirements” on page 710](#).

Unlike the other System Variables REST services, a get system variables request does not require the client to have READ access to ZOSMF.VARIABLES.SYSTEM.ADMIN resource profile in the ZMFAPLA class.

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 200 is returned with an empty array of variables.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 401. HTTP error response codes for a get system variables request	
HTTP error status code	Description
404	Requested system was not found.
404	The system identifier in the URI of the request is not valid.
500	A server error occurred during processing of the request.

## Response content

On successful completion, the service returns a response body, which contains a JSON object with details about the system variables. See [Table 402 on page 713](#) and [Table 403 on page 713](#). If no system variables match the filter criteria, HTTP status code 200 (OK) is returned with an empty array.

Table 402. Get system variables request response body		
Field name	Type	Description
system-variable-list	Array of objects	List of variables to be added or updated to the system variable pool.

Table 403. Get system variables request: objects		
Field name	Type	Description
name	String	Descriptive name for the variable
value	String	Value for the variable
description	String	Description of the variable

## Example HTTP interaction

In the following example, the GET method is used to get all of the system variables on a system.

```
GET /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE
```

Figure 365. Sample request to get system variables

```
{system-variable-list: [
  {"name": "sample1", "value": "20", "Description": "value of sample1"},
  {... }
]}
```

Figure 366. Sample response from a get system variables request

# Import system variables

Use this operation to import z/OSMF system variables from a file.

## HTTP method and URI path

```
POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/import
```

In this request, the URI path variables are described, as follows:

- <version> identifies the version of the z/OSMF system variables service. The following value is valid:  
1.0.
- <sysplex-name> identifies the sysplex.
- <system-name> identifies the system.

## Description

This operation imports system variables from a file. The file must be accessible by the authenticated user. The file contains variable definitions in comma-separated value (CSV) format, where each row consists of the variable name, value and description. There should be no header row in the file. The variables imported from the file are processed in the same way as variables that are specified with the create system variables API.

On successful completion, HTTP status code 204 (No content) is returned.

## Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class: ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 710.](#)

## Request content

The request content is expected to contain a JSON object. See [Table 404 on page 714.](#)

Table 404. Request content for the import system variables request			
Field name	Type	Required or optional	Description
variables-import-file	String	Required	Path to the CSV-formatted file containing the variables to import

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 204 is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 405. HTTP error response codes for a create system variables request	
HTTP error status code	Description
400	Request body is not syntactically correct
400	Specified file was either not found or could not be opened

Table 405. HTTP error response codes for a create system variables request (continued)

HTTP error status code	Description
400	Specified file has an incorrect format
401	Submitter of the request is not authorized to add or delete system variables
404	Requested system was not found
500	A server error occurred during processing of the request.

## Response content

None.

## Example HTTP interaction

In the following example, the POST method is used to import system variables from a file.

```
POST /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE/actions/import
```

*Figure 367. Sample request to import system variables*

The request body is as follows:

```
{ "variables-import-file": "/u/testuser/variables.csv" }
```

*Figure 368. Sample request body for an import system variables request*

## Export system variables

Use this operation to export z/OSMF system variables for a specific system to a file.

### HTTP method and URI path

```
POST /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>/actions/export
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF system variables service. The following value is valid:  
1.0.
- **<sysplex-name>** identifies the sysplex.
- **<system-name>** identifies the system.

### Description

This operation exports system variables, for the system identified in the URI, to a CSV file specified by the request body. It creates the file if it does not exist. Files created by this API can be imported with the import system variables API.

On successful completion, HTTP status code 204 (No content) is returned.

## Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class:  
ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 710](#).

## Request content

The request content is expected to contain a JSON object. See [Table 406 on page 716](#).

Table 406. Request content for the export system variables request			
Field name	Type	Required or optional	Description
<b>variables-export-file</b>	String	Required	Path to the file to contain the exported system variables. The file must be accessible to the authenticated user.
<b>overwrite</b>	Boolean	Optional	Indicates whether or not the file should be written to if it already exists. If the value is false and the file exists, the call returns with a status code 400. The value defaults to false if it is not specified.

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 204 is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 407. HTTP error response codes for a create system variables request	
HTTP error status code	Description
<b>400</b>	Request body is not syntactically correct
<b>400</b>	Path is not accessible for writing
<b>400</b>	File exists, but the request did not indicate that it should be written to
<b>404</b>	Requested system was not found
<b>500</b>	A server error occurred during processing of the request

## Response content

None.

## Example HTTP interaction

In the following example, the POST method is used to export system variables from a file.

```
POST /zosmf/variables/rest/1.0/systems/TESTPLEX.TESTNODE/actions/export
```

*Figure 369. Sample request to export system variables*

The request body is as follows:

```
{ "variables-export-file": "/u/testuser/backup-variables.csv", "overwrite":true }
```

Figure 370. Sample request body for an export system variables request

## Delete system variables

Use this operation to delete z/OSMF system variables from the system variable pool.

### HTTP method and URI path

```
DELETE /zosmf/variables/rest/<version>/systems/<sysplex-name>.<system-name>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF system variables service. The following value is valid:  
1.0.
- *<sysplex-name>* identifies the sysplex.
- *<system-name>* identifies the system.

### Description

This operation removes system variables from the system variable pool.

If all variables are removed, the system variable pool is empty. If there is no request body, this operation deletes the system variable pool. If the request body contains an empty array ([ ]), no action is taken. If the request body contains no variables in the array, no action is taken. If the request body contains variables that does not exist in the pool, those variables are ignored.

On successful completion, HTTP status code 204 (No content) is returned.

### Authorization requirements

Use of this API requires READ access to the following resource profile in the ZMFAPLA class:  
ZOSMF.VARIABLES.SYSTEM.ADMIN.

See also [“Authorization requirements” on page 710](#).

### Request content

The request content is expected to contain an array of strings. Each string represents the name of a system variable to delete.

### HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

If the system variable pool does not exist for the requested system, HTTP status code 204 is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors.

Table 408. HTTP error response codes for a create system variables request	
HTTP error status code	Description
400	Request body is not formatted correctly
401	Submitter of the request is not authorized to delete the system variables

Table 408. HTTP error response codes for a create system variables request (continued)

HTTP error status code	Description
404	Requested system was not found
500	A server error occurred during processing of the request.

## Response content

None.

## Example HTTP interaction

In the following example, the DELETE method is used to delete system variables from a system variable pool.

```
DELETE /zosmf/variables/rest/1.0/variables/systems/<sysplex-name>.<system-name>
```

Figure 371. Sample request to delete system variables

The request body is as follows:

```
["var1", "var2", ...]
```

Figure 372. Sample request body for a delete system variables request

## z/OSMF authentication services

The z/OSMF authentication services API is provided for z/OSMF tasks and vendor applications. This API is used to obtain or delete authentication tokens (a JSON Web Token and an LTPA token) on the user's authentication request when logging in to or out of z/OSMF. This API can also be used to change a z/OSMF user's password.

Table 409. z/OSMF authentication services method

Operation	HTTP method and URI path
<a href="#">“Log in to the z/OSMF server” on page 719</a>	POST /zosmf/services/authenticate
<a href="#">“Change the user password or passphrase” on page 721</a>	PUT /zosmf/services/authenticate
<a href="#">“Log out of the z/OSMF server” on page 724</a>	DELETE /zosmf/services/authenticate

For information about enabling the z/OSMF server to produce JSON Web Tokens, see [Enabling JSON Web Token support in IBM z/OS Management Facility Configuration Guide](#).

## Error handling

For errors that occur during the processing of a request, the API returns an appropriate HTTP status code to the calling client. An error is indicated by a 4nn code or a 5nn code. For example, HTTP/1.1 400 Bad Request or HTTP/1.1 500 Internal Server Error.

In addition, some errors might also include a returned JSON object that contains a message that describes the error. You can use this information to diagnose the error or provide it to IBM Support, if required.

The following HTTP status codes are valid:

### HTTP 200 OK

Request was processed successfully.

### HTTP 400 Bad request

Request could not be processed because it contains a syntax error or an incorrect parameter.

### HTTP 401 Unauthorized

Request could not be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both, or the client did not authenticate to z/OSMF.

### HTTP 500 Internal server error

Server encountered an error. See the response body for a JSON object with information about the error.

### HTTP 503 Service unavailable

Server is not available.

## Error logging

Errors from the z/OSMF authentication services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Log in to the z/OSMF server

You can use the POST method to log in to the z/OSMF server and obtain authentication tokens. This service creates a JSON Web Token, an LTPA token, or both, and returns the tokens to the requester.

### HTTP method and URI path

---

```
POST /zosmf/services/authenticate
```

---

Where:

- **/zosmf/services** specifies the z/OSMF REST services API.
- **/authenticate** indicates an authentication request.

### Query parameters

None.

### Standard headers

Use the following standard HTTP headers with this request:

#### **Content-Type: application/x-www-form-urlencoded**

Indicates that the body of the HTTP message is a query string, consisting of name=value pairs, with each pair separated by a single ampersand (&).

#### **Authorization: Basic <credentials>**

In the context of an HTTP transaction, basic access authentication is a method for an HTTP user agent (such as a web browser) to provide a user ID and password with a request. In basic HTTP authentication, a request contains a header field in the form of **Authorization: Basic <credentials>**, where **<credentials>** is the Base64 encoding of ID and password, joined by a single colon (:).

## Customer headers

### X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or an empty string (" "). For more information, see [“Allowing cross-site access to REST services” on page 5](#).

## Request body

None.

## Response Body

Table 410. Response body for a "log in to the z/OSMF server" request	
Field name	Description
<b>returnCode</b>	Identifies the category of errors
<b>reasonCode</b>	Specified file is either not found or cannot be opened.
<b>message</b>	Describes the text information of the login result.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error occurred.

Table 411. HTTP error response codes for a "log in to the z/OSMF server" request				
HTTP error status code	Return code	Reason code	Message	Description
200	0	0	Success.	User logs in to z/OSMF server successfully.
401	8	1	Login failed. Check whether the user ID and password you use for the Basic Auth is correct, and if the user ID has the required SAF permissions.	Check whether the user ID and password you use for the Basic Auth is correct, and if the user ID has the required SAF permissions.
401	8	12	The password or passphrase is expired.	Your password or passphrase is expired. Reset the password or passphrase.
401	8	28	The user ID was revoked.	Your user ID was revoked. Contact your system administrator.
401	4	1	Login failed. The Basic Auth information in the request header is incorrect.	Correct any errors in the Basic Auth information, which is in the request header.

Table 411. HTTP error response codes for a "log in to the z/OSMF server" request (continued)

HTTP error status code	Return code	Reason code	Message	Description
401	4	40	The request failed because an internal error occurred.	The request failed because an internal error occurred. Contact your system administrator.

**Note:** In the **General Settings > Home Page**, if the option **Display error details when login fails** is selected, the error message indicates when a password is expired (return code 8, reason code 12), or a user ID was revoked (return code 8, reason code 28). Otherwise, if the option **Display error details when login fails** is not selected, an expiration or revocation error returns only the general failure message: return code 8, reason code 1.

### Example

In the following example, the POST method is used to log in to the z/OSMF server and obtain a JSON Web Token for the user. On completion, the token is saved as a web cookie.

```
POST /zosmf/services/authenticate HTTP/1.1
Host: your.company.com
Referer: https://your.company.com/zosmf
Content-Type: application/x-www-form-urlencoded
cookie: "jwtToken=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2t1b190eXB1IjojQmVhcmVyIiwic3ViIjoiem9zbWZhZCIsInVwbiI6Inpvc21mYWQlLCJncm91cHMlOi00VBR1AilCJDUE9DVFJMIiwia1BPVVVU1k1LCJlVBR1JTtIsIldMTUdSUCIsIkjN00ZHI10sInJlYWxtIjojU0FGUmVhbG0iLCJpc3MiOiJ6T1NNRiIsImV4cCI6MTU0ODg2MTM0NiwiawF0IjoxNTQ4ODMxOTQ2fQ.Bfc9MqPSRfn-rz0Gryf_24klKD8rqZB0TwwYwYB9osdKpFDbS2wKtqKMyBx6gcPX649Uk9mhSj1VEAz71A0gOLi9KA28rMj1mQZbimkzNzXEPpQp37HG5Ve8aGvtxCdsyPKUQAQC9DRCdB-MGLjwCaLokRI7BjtDhriwkK17yBVAx0GV7Gs8arFQzsTZS1Rj3VWJ-wpu0JATRUCsjBLPE7inZzxzGIlh0eyM0K_0EH6YB_RTEsdf21SbNy36dA2aJaaYHLw9j1-fJ1VNwMf4ipFy_x2em8bdLYIhVT0ujLtjWmd2M9dEnYGs_rt6xpKfBCILN0yXowrrWEchEPE1Q"
cache-control: no-cache
response body:
{"returnCode":0,"reasonCode":0,"message":"Success."}
```

Figure 373. Log in and obtain a JSON Web Token for user authentication

## Change the user password or passphrase

You can use a PUT request to change a user password or password phrase (passphrase). This service is available when you install the PTF for APAR PH34912.

### HTTP method and URI path

```
PUT /zosmf/services/authenticate
```

Where:

- **/zosmf/services** specifies the z/OSMF REST services API.
- **/authenticate** indicates an authentication request.

### Query parameters

None.

### Standard headers

Use the following standard HTTP header with this request:

## Customer headers

### X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or the empty string (" "). For more information, see [“Allowing cross-site access to REST services” on page 5](#).

## Request content

The request content is expected to contain a JSON object. See [Table 412 on page 722](#).

*Table 412. Request content for a change user password or passphrase request*

Field name	Type	Required or optional	Description
<b>userID</b>	String	Required	User ID.
<b>oldPwd</b>	String	Required	The current (or old) password or passphrase for the user ID.
<b>newPwd</b>	String	Required	The new password or passphrase for the user ID. Ensure that the same type of authentication value is used: Replace a password with a password; replace a passphrase with a passphrase.

## Response Body

*Table 413. Response body for a change user password or passphrase request*

Field name	Description
<b>returnCode</b>	Identifies the category of errors
<b>reasonCode</b>	Identifies the specific errors.
<b>message</b>	Describes the text information of the change request.

## Required authorizations

See [“Required authorizations” on page 567](#).

## Usage considerations

See [“Usage considerations for the z/OSMF REST services” on page 3](#).

## Expected response

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 200 OK indicates success. A status code of 4nn or 5nn indicates that an error occurred.

Table 414. HTTP error response codes for a change user password or passphrase request

HTTP error status code	Return code	Reason code	Message	Description
200	0	0	Success.	The password or passphrase is changed.
400	4	4	The user ID is not defined to RACF.	The user ID is not defined to RACF. Verify that the user ID is spelled correctly. This value is case-sensitive.
400	4	21	The user ID is not valid. It is either empty or exceeds the character limit.	The user ID is not valid. It is either empty or exceeds the character limit.
400	4	22	The current (old) password cannot be null.	The current (old) password cannot be null.
400	4	23	The new password cannot be null.	The new password cannot be null.
500	8	2	Change password failed. Check whether the user ID and oldPwd you provide is correct.	Determine whether the user ID and old password you provided are correct. You cannot change a password to a passphrase, or vice versa.
500	8	16	Change password failed. newPwd is invalid.	At least one of the following conditions is true: <ul style="list-style-type: none"> <li>• The new password or passphrase is not valid.</li> <li>• A new password was specified with a current password, or a new passphrase was specified with a current passphrase.</li> <li>• A new passphrase was specified with a PassTicket as the current password, but the user does not currently have a passphrase.</li> <li>• A password or passphrase change is disallowed now because the minimum password-change interval has not elapsed.</li> </ul>
500	8	28	The user ID was revoked.	Your user ID was revoked. Contact your system administrator.
500	8	29	The request body is not in JSON format.	The request body is required to be in JSON format. Convert the request body to JSON format.
500	4	40	The request failed because an internal error occurred.	The request failed because an internal error occurred. Contact your system administrator.

**Note:** In the **General Settings > Home Page**, if the option **Display error details when login fails** is selected, the error message indicates when a user ID was revoked (return code 8, reason code 28) or a user ID profile is not defined to RACF (return code 4, reason code 4). Otherwise, if the option **Display error details when login fails** is not selected, an undefined or revoked user ID returns only the general failure message: return code 8, reason code 2.

## Example

In the following example, the PUT method is used to change a user password. On completion, the user password is changed.

```
PUT https://your.company.com/zosmf/authenticate HTTP/1.1
Host: your.company.com
Referer: https://your.company.com/zosmf
Content-Type: application/x-www-form-urlencoded
{"userID": "ZOSMFT1", "oldPassword": "QWER1234", "newPassword": "ZOSMFT1"}

response:
HTTP/1.1 200 OK
response body:
{"returnCode":0,"reasonCode":0,"message":"Success."}
```

*Figure 374. Change a user password*

## Log out of the z/OSMF server

You can use the DELETE method to log out of the z/OSMF server and delete the user's authentication tokens (JSON Web Tokens and LTPA tokens). Your request cookie must include a valid JSON Web Token or LTPA token (or both).

### HTTP method and URI path

```
DELETE /zosmf/services/authenticate
```

Where:

- **/zosmf/services** specifies the z/OSMF REST services API.
- **/authenticate** indicates an authentication request.

### Query parameters

None.

### Standard headers

Use the following standard HTTP header with this request:

```
Content-Type: application/x-www-form-urlencoded
```

### Customer headers

#### X-CSRF-ZOSMF-HEADER

This header is required for both browser and non-browser applications. Set the header to any value or an empty string (""). For more information, see [“Allowing cross-site access to REST services” on page 5](#).

### Request body

None.

### Response Body

None.

See [“Required authorizations”](#) on page 567.

See [“Usage considerations for the z/OSMF REST services”](#) on page 3.

On completion, the service returns an HTTP response, which includes a status code that indicates whether your request completed. Status code 204 indicates success. A status code of 4nn or 5nn indicates that an error occurred.

In the following example, the DELETE method is used to delete all the authenticated tokens when user logs out.

Figure 375. Delete the JSON Web Tokens for this user

The z/OSMF workflow services are an application programming interface (API), which is implemented through industry standard Representational State Transfer (REST) services. These services allow the caller to create and manage z/OSMF workflows on a z/OS system.

Table 415. z/OSMF workflow services: operations summary

Chapter 1. Using the z/OSMF REST services **725**

Table 415. z/OSMF workflow services: operations summary (continued)	
Operation name	HTTP method and URI path
<a href="#">“Delete a workflow” on page 768</a>	DELETE /zosmf/workflow/rest/<version>/workflows/<workflowKey>
<a href="#">“Retrieve a workflow definition” on page 770</a>	GET /zosmf/workflow/rest/<version>/workflowDefinition
<a href="#">“Archive a workflow instance” on page 783</a>	POST /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/archive
<a href="#">“List the archived workflows for a system” on page 785</a>	GET /zosmf/workflow/rest/<version>/archivedworkflows
<a href="#">“Get the properties of an archived workflow” on page 787</a>	GET /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>
<a href="#">“Delete an archived workflow” on page 801</a>	DELETE /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>

Table 416 on page 726 describes the variables that can be specified in the z/OSMF workflow services URI paths.

Table 416. z/OSMF workflow services: URI path variables	
URI path variable	Description
<version>	The version of the z/OSMF workflow services API. The following value is valid: 1.0.
<workflowKey>	The identifier of a unique instance of a workflow, as returned in the response of the operation that created the workflow.

## Using the Swagger interface

You can use the Swagger interface to display information about the z/OSMF workflow REST APIs. For more information, see [“Using the Swagger interface” on page 1](#).

## Authorization requirements

Use of the z/OSMF workflow services API requires the client to be authenticated. For information about client authentication in z/OSMF, see [“Authenticating to z/OSMF” on page 2](#).

In addition, the user's z/OS user ID must have:

- READ access to the <SAF-PREFIX>.ZOSMF.WORKFLOW.WORKFLOWS profile in the ZMFAPLA class.
- READ access to the <SAF\_PREFIX>.\*.izuUsers profile in the EJBROLE class. Or, at a minimum, READ access to the <SAF\_PREFIX>.IzuManagementFacilityWorkflow.izuUsers resource name in the EJBROLE class.

## Error response content

For most 4nn and 5nn HTTP error status codes, additional diagnostic information beyond the HTTP status code is provided in the response body for the request. This information is provided in the form of a JSON object containing the following fields:

Table 417. Error response body elements for the z/OSMF workflow services API

Field name	Type	Description
<b>messageID</b>	String	The message identifier identifying the reason for the error.
<b>messageText</b>	String	The message text that describes the error.

## Error logging

Errors from the z/OSMF workflow services are logged in the z/OSMF log. You can use this information to diagnose the problem or provide it to IBM Support, if required. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## HTTP status codes

The following HTTP status codes are valid:

### HTTP 200 OK

The request succeeded. A response body is provided, which contains the results of the request.

### HTTP 201 Created

The request succeeded and resulted in the creation of an object.

### HTTP 202 Accepted

The request was successfully validated and is performed asynchronously.

### HTTP 204 No content

The request succeeded, but no content is available to be returned.

### HTTP 400 Bad request

The request contained incorrect parameters.

### HTTP 401 Unauthorized

The request cannot be processed because the client is not authorized. This status is returned if the request contained an incorrect user ID or password, or both. Or, the client did not authenticate to z/OSMF by using a valid WWW-Authenticate header.

### HTTP 403 Forbidden

The server received the request, but rejected it.

### HTTP 404 Not found

The requested resource does not exist.

### HTTP 405 Method not allowed

The requested resource is a valid resource, but an incorrect method was used to submit the request. For example, the request used the POST method when the GET method was expected.

### HTTP 408 Request timed out

The client did not produce a request within the allowed time. The request can be submitted again later.

### HTTP 409 Request conflict

The request cannot be processed because of conflict in the request, such as an edit conflict between multiple updates.

### HTTP 500 Server error

The server encountered an error when it processed the request. For a more specific indication of the error, check the response for a reason code.

### HTTP 501 Not implemented

The request specifies an HTTP method that is not recognized by the server.

### HTTP 503 Service unavailable

The request cannot be carried out by the server because of a temporary condition. A suggested wait time might be indicated in a Retry-After header, if one is provided in the response. Otherwise, the requestor can treat the response as a 500 response.

### HTTP 504 Gateway timeout

The server, which is acting as a gateway or proxy, did not receive a timely response from the server that was specified in the URI path (for example, HTTP, FTP, LDAP) or an auxiliary server (such as DNS). This access is needed to complete the request. For example, the server was not able to start a remote REXX or UNIX shell interface.

## Create a workflow

You can use this operation to create a z/OSMF workflow on a z/OS system.

### HTTP method and URI path

---

```
POST /zosmf/workflow/rest/<version>/workflows
```

---

In this request, the URI path variable `<version>` identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.

### Query parameters

None.

### Description

This operation creates a workflow, based on the properties that are specified in the request body (a JSON object). For the properties that you can specify, see [“Request content” on page 729](#).

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the creation of a new workflow. The URI path for the workflow is provided in the Location response header and a response body is provided, as described in [“Response content” on page 735](#).

### Workflow access type

By default, general information about the workflow and its steps can be viewed by all users of the Workflows task. If you want to restrict access to a workflow or portions of a workflow, you can do so by specifying an *access type* in the request body for your Create Workflow request. The access type determines which users can view the workflow steps and edit the step notes. The access type is specified on the `accessType` property.

The valid values for the `accessType` property are summarized, as follows:

#### Public

Information about the workflow, including the steps and notes, can be viewed by all users.

#### Restricted

Information about steps, variables, and notes, is restricted to a subset of users — the workflow owner, step owners, and step assignees. Other users cannot access this information.

#### Private

Information is restricted to a subset of users, and is further limited among these users. The workflow owner can access information about steps, variables, and notes. Step owners and assignees can retrieve information about the steps for which they are assigned or own, and the associated variables for those steps. Other users cannot access this information.

The `accessType` property is optional. If you omit it from the request body, the workflow is created with public access.

Except for workflow notes and step notes, this information is also available to REST API requesters through the Get Workflow Properties service. For the types of information that are restricted by access type, see [“Get the properties of a workflow” on page 736](#).

## Request content

The request content is expected to contain a JSON object that describes the workflow to be created. Table 418 on page 729 lists the fields in the JSON object.

Table 418. Request content for the create workflow request			
Field name	Type	Required or optional	Description
<b>workflowName</b>	String	Required	Descriptive name for the workflow (up to 100 characters). The name cannot contain the symbols for less-than (<), greater-than (>), or ampersand (&). z/OSMF validates this name to ensure that it is unique across all of the existing workflows.
<b>workflowDefinitionFile</b>	String	Required	Location of the workflow definition file. This file is the primary XML file for the workflow definition.  Specify this value, as follows: <ul style="list-style-type: none"> <li>• If the workflow definition file resides in a data set member, specify the fully qualified data set name, including the member name. Ensure that this data set is cataloged.</li> <li>• If the workflow definition file resides in a z/OS UNIX file, specify the fully qualified path name of the file, beginning with the forward slash (/) and including the file name. For example: /usr/lpp/zosmf/samples/workflow_sample_automation.xml.</li> </ul>
<b>workflowDefinitionFileSystem</b>	String	Optional	Nickname of the system on which the specified workflow definition file and any related files reside. The Workflows task obtains the workflow files from this system.  Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is a unique name for the system to differentiate it from existing systems that have the same system and sysplex name. The nickname is 1 - 40 characters long; the valid characters are alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and special characters (\$ _ # @). Nicknames are case-sensitive; for example, SYSTEM1 and System1 are unique values.  The system can be running in the local sysplex or in another sysplex in your enterprise. If you select a system in a remote sysplex, verify that the system is enabled for single sign-on (SSO). Otherwise, your request must include a valid user ID and password (in the request body) for basic authentication with the remote system.  If you omit this property, the Workflows task checks the z/OSMF system of the local sysplex for the workflow definition file and related files, by default.

Table 418. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
<b>variableInputFile</b>	String	Optional	<p>Specifies an optional properties file that you can use to pre-specify values for one or more of the variables that are defined in the workflow definition file.</p> <p>Specify this property, as follows:</p> <ul style="list-style-type: none"> <li>• If the workflow variable input file resides in a data set member, specify the fully qualified data set name, including the member name. Ensure that this data set is cataloged.</li> <li>• If the workflow variable input file resides in a z/OS UNIX file, specify the fully qualified path name of the file, beginning with the forward slash (/) and including the file name.</li> </ul> <p>For the format of the contents of the variable input file, see <a href="#">“Providing a workflow variable input file” on page 871</a>.</p>
<b>variables</b>	Array of objects	Optional	<p>A list of one or more variables for this workflow. The variables that you specify here take precedence over the variables that are specified in the workflow variable input file.</p> <p>Specify this property as a list of name-value objects, for example:</p> <pre>"variables": [   {"name": "user_name", "value": "IBMUSER"},   {"name": "file_name", "value": "textfile.txt"} ]</pre> <p>If you plan to include an <i>array</i> type variable on the variables property, specify the variable as a list of strings, with each string separated by the following escape character: \. In the following example, the variable array3 is an array variable:</p> <pre>"variables": [{"name": "user_name", "value": "IBMUSER"},  {"name": "file_name", "value": "textfile.txt"},  {"name": "array3", "value": clust   "[ \Z0SV23T\, \DB211T\,    {\\"property1\\":\\"tt1\\",     \\"dsname\\":\\"TEST.DSNAME.TT1\\"}]"}</pre> <p>For more information about array variables, see <a href="#">“Array variables” on page 869</a>.</p>

Table 418. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
<b>resolveGlobalConflictBy Using</b>	String	Optional	<p>On creation of the workflow, z/OSMF determines whether any of the variables that are supplied in this request (through the variable input file or variables array) would conflict with existing global variables in the Workflows task. In such cases, this property specifies which version of the variable is used, as follows:</p> <ul style="list-style-type: none"> <li>• When set to <code>input</code>, the global variable conflicts are overridden by the variables in specified input file. The global variable value is updated with the input variable value. Use caution with this setting; your selection affects any other workflows that refer to the same global variable.</li> <li>• When set to <code>global</code>, or omitted, the variable value that is supplied with the request (through the variable input file or variables array) is ignored and the current global value is used.</li> </ul> <p>The default is <code>global</code>.</p>

Table 418. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
<b>system</b>	String	Required	<p>Nickname of the system on which the workflow is to be created. Use the nickname that is specified for the system definition in the z/OSMF Systems task.</p> <p>The nickname is a unique name for the system to differentiate it from existing systems that have the same system and sysplex name. The nickname is 1 - 40 characters long; the valid characters are alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and special characters (\$ _ # @). Nicknames are case-sensitive; for example, SYSTEM1 and System1 are treated as different values.</p> <p>The workflow steps are performed on this system. Any jobs or scripts in the workflow are run on this system. Similarly, any work that you perform manually for the workflow is done on this system.</p> <p>If the workflow is to be created on a system in a remote sysplex:</p> <ul style="list-style-type: none"> <li>• If the system is running z/OSMF, verify that the system is enabled for single sign-on (SSO). Otherwise, your request must include a valid user ID and password (in the request body) for basic authentication with the remote system.</li> <li>• If the system is not running z/OSMF, it must be associated with the z/OSMF system for that sysplex. If so, set the z/OSMF system as the host system for the system on which the workflow is to be performed. Similarly, you must ensure that the z/OSMF system in the remote sysplex is enabled for single sign-on. Or, you must include a valid user ID and password (in the request body) for basic authentication with the remote system.</li> </ul> <p>For more information about defining z/OSMF systems and enabling them for single sign-on, see <a href="http://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zosmfcore.multisysplex.help.doc/izuG00hpSystemPage.html">Defining your systems to z/OSMF (www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zosmfcore.multisysplex.help.doc/izuG00hpSystemPage.html)</a>.</p>
<b>owner</b>	String	Required	<p>User ID of the workflow owner. This user can perform the workflow steps or delegate the steps to other users.</p> <p>Specify a valid user ID, as it is defined to your installation's z/OS security management product, such as RACF. A valid user ID consists of one to eight alphanumeric characters (A-Z, a-z, 0-9, #, \$, and @).</p>
<b>comments</b>	String	Optional	<p>Specifies any information that you want to associate with the creation of this workflow (up to 500 characters). This information is recorded in the workflow history. Consider including a meaningful comment on the workflow, for example: This workflow was created through the z/OSMF workflow services REST interface.</p>

Table 418. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
<b>assignToOwner</b>	Boolean	Optional	Indicates whether the workflow steps are assigned to the workflow owner when the workflow is created. If you set this property to true, or omit the property, z/OSMF assigns the steps to the user ID that is specified on the property owner. If you set this property to false, the workflow steps are left unassigned when the workflow is created. The default is true.
<b>accessType</b>	String	Optional	<p>Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in <a href="#">“Workflow access type”</a> on page 728.</p> <p>The following values are valid:</p> <ul style="list-style-type: none"> <li>• Public</li> <li>• Restricted</li> <li>• Private</li> </ul> <p>If you omit this property, the workflow is public, by default.</p>
<b>accountInfo</b>	String	Optional	For a workflow that submits a job, this property specifies the account information to use in the JCL JOB statement. This property can be null.
<b>jobStatement</b>	String	Optional	For a workflow that submits a job, this property specifies the JOB statement JCL that is used in the job. This property can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 - 8 characters.
<b>deleteCompletedJobs</b>	Boolean	Optional	<p>For a workflow that submits a job, this property specifies whether the job is deleted from the JES spool after it completes successfully. To retain the job, set this property to false, which is the default. If so, the job remains on the JES spool until it is removed by a user or automated process. To conserve space in the JES spool, consider setting this property to true.</p> <p>If you omit this property, the completed job is retained on the JES spool.</p>
<b>jobsOutputDirectory</b>	String	Optional	<p>For a workflow that submits a job, this property specifies the name of a UNIX directory that is to be used for automatically saving job spool files from the workflow. Specify a valid UNIX file path and directory on the user's system, beginning with a single forward slash (/). For example: /u/IBMUSER/jobFiles.</p> <p>If you omit this property, the job spool files are not saved.</p>

Table 418. Request content for the create workflow request (continued)

Field name	Type	Required or optional	Description
<b>autoDeleteOnCompletion</b>	Boolean	Optional	<p>Indicates whether the workflow is automatically deleted from the local system when all of its steps are marked complete or skipped. If so, include this property in your request and set it to <code>true</code>.</p> <p>When a workflow is deleted, it no longer appears in the Workflows table in the Workflows task user interface. You might use this option to avoid reaching the system limit of 200 workflows.</p> <p>Otherwise, if you want to retain the workflow after it is complete, set this property to <code>false</code> or omit the property from your request. Here, the workflow is retained in the Workflows table until you explicitly delete the workflow.</p> <p>If you omit this property, the workflow instance is retained.</p> <p><b>Note:</b> A called workflow cannot be deleted until all of its calling workflows are either deleted or archived.</p>
<b>targetSystemuid</b>	String	Optional	The user ID to be used for remote system basic authentication.
<b>targetSystempwd</b>	String	Optional	The password to be used for remote system basic authentication.

## Authorization requirements

See [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content” on page 735](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body with the reason code that is indicated and associated error message.

Table 419. HTTP error response codes for a create workflow request

HTTP error status code	Description
<b>HTTP 400 Bad request</b>	<p>The request contained incorrect parameters. For example:</p> <ul style="list-style-type: none"> <li>The specified workflow name contains errors or is not unique.</li> <li>Workflow definition file contains errors or does not exist.</li> <li>Variable input file does not exist.</li> <li>Validation error. The specified value does not match the validation criteria for one or more of the following properties: <code>variables</code>, <code>system</code>, <code>owner</code>, or <code>comments</code>.</li> <li>An incorrect value is specified for the property <code>resolveGlobalConflictByUsing</code>.</li> </ul>
<b>HTTP 401 UNAUTHORIZED</b>	The requester user ID is not permitted to perform the attempted action.

Table 419. HTTP error response codes for a create workflow request (continued)	
HTTP error status code	Description
<b>HTTP 403 Forbidden</b>	The requester user ID is not permitted to the workflow definition file or the variable input file.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the service returns the following:

- URI path of the created workflow in the Location response header.
- Response body, which contains a JSON object with details about the workflow. [Table 420 on page 735](#) lists the fields in the JSON object.

Table 420. Response from a create workflow request		
Field name	Type	Description
<b>workflowKey</b>	String	Workflow key. A string value, which is generated by z/OSMF to uniquely identify the workflow instance.
<b>workflowDescription</b>	String	Workflow description. This value is obtained from the element workflowDescription in the workflow definition file.
<b>workflowID</b>	String	Workflow ID. A short, arbitrary value that identifies the workflow. This value is obtained from the element workflowID in the workflow definition file.
<b>workflowVersion</b>	String	Version of the workflow definition file. This value is obtained from the element workflowVersion in the workflow definition file.
<b>vendor</b>	String	Name of the vendor that provided the workflow definition file. This value is obtained from the element vendor in the workflow definition file.

## Example HTTP interaction

In [Figure 376 on page 735](#), a request is submitted to create the workflow AutomationExample on the system SY1.

```
POST /zosmf/workflow/rest/1.0/workflows HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Content-Type: application/json
Content-Length: 203
Authorization: Basic em9zbWZhZDp6b3NtZmFk

{
  "workflowName": "AutomationExample",
  "workflowDefinitionFile": "/usr/lpp/zosmf/samples/workflow_sample_automation.xml",
  "system": "SY1",
  "owner": "zosmfad",
  "assignToOwner": true,
  "accessType": "Restricted",
  "deleteCompletedJobs": true
  "autoDeleteOnCompletion": true
}
```

*Figure 376. Sample request to create a workflow*

A sample response is shown in [Figure 377 on page 736](#).

```

HTTP/1.1 201 Created
content-length: 210
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
location: /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0
date: Mon, 21 Oct 2019 18:29:55 GMT
content-type: application/json; charset=UTF-8

{
  "vendor": "IBM",
  "workflowDescription": "Sample demonstrating the use of automated steps in workflow.",
  "workflowID": "automationSample",
  "workflowKey": "d043b5f1-adab-48e7-b7c3-d41cd95fa4b0",
  "workflowVersion": "1.0"
}

```

Figure 377. Sample response from a create workflow request

## Get the properties of a workflow

You can use this operation to retrieve the properties of a z/OSMF workflow.

### HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/workflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the workflow to be queried.

### Query parameters

You can specify the following query parameter on this request:

#### returnData

This optional query parameter is used to request information about the workflow steps and variables. Include one or both of the following attributes on the `returnData` parameter:

##### steps

Returns an array of step-info objects; one object for each step in the workflow. [Table 424 on page 745](#) lists the fields in the step-info JSON object.

##### variables

Returns an array of variable-info objects; one object for each variable that is referenced in the workflow. [Table 429 on page 755](#) lists the fields in the variable-info JSON object.

To specify both attributes, separate the attributes by a comma (','), as follows:

```
returnData=steps,variables
```

Do not enclose the attributes in quotation marks.

The response data is limited by the access type of the workflow. For more information, see [“Effects of access type on the returned data” on page 737](#).

### Description

This operation retrieves the properties of a z/OSMF workflow. You can optionally expand the returned information through the specification of query parameters. On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 422 on page 739](#).

For the format of this information, see the JSON objects that are described in [Table 424 on page 745](#) and [Table 429 on page 755](#).

## Authorization requirements

See [“Authorization requirements” on page 726](#).

## Effects of access type on the returned data

If you include the optional query parameter `returnData` on the request, the operation can return information about the workflow steps or variables, or both. The amount of data that can be retrieved about the workflow steps or variables can be restricted by the workflow *access type*. This value is specified by the workflow owner at workflow creation time.

Generally, a workflow with a public access type is less restricted in the amount of data that is available to the requestor. A workflow with a restricted or private access type is more secure, and requires the caller user ID to be a workflow owner, step owner, or step assignee to use or access areas in the workflow.

All requestors can retrieve the workflow common properties. This data includes the information that is shown in [Table 422 on page 739](#).

For variables data, the workflow owner can retrieve all of the variable properties for the workflow. The step owner and step assignees can retrieve the variable properties that are associated with steps they own. Other users cannot retrieve this data; requests for details about the variables from these users result in empty arrays being returned.

For steps data, the returned data depends on the type of step data that is requested. The access type allows the workflow creator to distinguish between the following types of steps data.

Step data type	Steps properties	Public access workflow	Restricted access workflow	Private access workflow
Step common properties	<ul style="list-style-type: none"><li>• title</li><li>• name</li><li>• owner</li><li>• stepNumber</li><li>• assignees</li><li>• state</li><li>• skills</li><li>• weight</li><li>• autoEnable</li><li>• hasCalledWorkflow</li><li>• userDefined</li><li>• optional</li></ul>	This data can be retrieved by all requestors.	This data can be retrieved by: <ul style="list-style-type: none"><li>• Workflow owner</li><li>• Step owner and assignees</li></ul>	This data can be retrieved by: <ul style="list-style-type: none"><li>• Workflow owner</li><li>• Step owner or assignee</li></ul>
Step restricted properties	<ul style="list-style-type: none"><li>• description</li><li>• prereqStep</li></ul>	This data can be retrieved by all requestors.	This data can be retrieved by: <ul style="list-style-type: none"><li>• Workflow owner</li><li>• Step owner and assignees</li></ul>	The workflow owner can retrieve this data for all steps.  The step owner and assignees can retrieve this data for their steps only.

Step data type	Steps properties	Public access workflow	Restricted access workflow	Private access workflow
Step detail properties	<ul style="list-style-type: none"> <li>calledInstanceURI</li> <li>calledWorkflowID</li> <li>calledWorkflowVersion</li> <li>calledWorkflowMD5</li> <li>calledWorkflowDescription</li> <li>calledWorkflowDefinitionFile</li> <li>failedPattern</li> <li>instructions</li> <li>instructionsSub</li> <li>isConditionStep</li> <li>jobInfo</li> <li>maxLrecl</li> <li>output</li> <li>outputSub</li> <li>outputVariablesPrefix</li> <li>procName</li> <li>regionSize</li> <li>returnCode</li> <li>saveAsDataset</li> <li>saveAsDatasetSub</li> <li>saveAsUnixFile</li> <li>saveAsUnixFileSub</li> <li>scriptParameters</li> <li>submitAs</li> <li>successPattern</li> <li>template</li> <li>templateSub</li> <li>timeout</li> <li>variable-references</li> </ul>	<p>The workflow owner can retrieve this data for all steps.</p> <p>The step owner and assignees can retrieve this data for their steps only.</p>	<p>The workflow owner can retrieve this data for all steps.</p> <p>The step owner and assignees can retrieve this data for their steps only.</p>	<p>The workflow owner can retrieve this data for all steps.</p> <p>The step owner and assignees can retrieve this data for their steps only.</p>

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 422 on page 739](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 421. HTTP error response codes for a get workflow properties request	
HTTP error status code	Description
<b>HTTP 404 Not found</b>	The specified workflow key was not found; the workflow does not exist.

Other standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 422 on page 739](#) lists the fields in the JSON object.

Table 422. JSON object that is returned to a get workflow properties request

Field name	Type	Description
<b>workflowName</b>	String	Descriptive name for the workflow.
<b>workflowKey</b>	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
<b>workflowDescription</b>	String	Description of the workflow.
<b>workflowID</b>	String	Workflow ID. A short, arbitrary value that identifies the workflow.
<b>workflowVersion</b>	String	Version of the workflow definition file.
<b>workflowDefinitionFileMD5Value</b>	String	The 128-bit hash value that is associated with the workflow definition file that was used to create the workflow.
<b>vendor</b>	String	Name of the vendor that provided the workflow definition file.
<b>owner</b>	String	User ID of the workflow owner.
<b>system</b>	String	Full name of the z/OS system on which the workflow is to be performed. This value is in the format <i>sysplex.sysname</i> .
<b>jobsOutputDirectory</b>	String	Name of the UNIX directory that is used for automatically saving job spool files from the workflow.
<b>category</b>	String	Category of the workflow, which is general, configuration, or provisioning.
<b>productID</b>	String	Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).
<b>productName</b>	String	Name of the product or component that is being configured through the workflow.
<b>productVersion</b>	String	Version and release of the product or component that is configured through the workflow.
<b>percentComplete</b>	Integer	Percentage of the workflow that is completed. z/OSMF calculates this value based on the number of steps in the workflow and the relative weighting value of each step.
<b>isCallable</b>	Boolean	Indicates whether a workflow is eligible to be called by another workflow. For more information, see <a href="#">“Callable workflows”</a> on page 815.
<b>containsParallelSteps</b>	Boolean	For a parallel-steps workflow, this property is <code>true</code> . If so, the automation ready steps can be run in parallel (concurrently), thus possibly completing more quickly.  Otherwise, if this property is <code>false</code> , automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.
<b>scope</b>	String	Restricts a workflow to one instance only. The scope attribute can be set to <code>system</code> , <code>sysplex</code> , or <code>none</code> . For more information, see <a href="#">“Setting the workflow scope”</a> on page 815.

Table 422. JSON object that is returned to a get workflow properties request (continued)

Field name	Type	Description
<b>statusName</b>	String	<p>Indicates the current workflow status, as follows:</p> <p><b>in-progress</b> One or more steps in the workflow are started.</p> <p><b>complete</b> Workflow is complete. All steps are marked complete or skipped.</p> <p><b>automation-in-progress</b> Workflow contains an automated step that is running.</p> <p><b>canceled</b> Workflow is canceled and cannot be resumed. However, you can view its properties or delete it.</p>
<b>deleteCompletedJobs</b>	Boolean	<p>For a workflow that submits a job, this property specifies whether the job is deleted from the JES spool after it completes successfully, as follows:</p> <ul style="list-style-type: none"> <li>• <b>false</b> means that the job is retained on the JES spool until it is removed by a user or automated process.</li> <li>• <b>true</b> means that the job is deleted from the JES spool after it completes or fails.</li> </ul>

Table 422. JSON object that is returned to a get workflow properties request (continued)

Field name	Type	Description
<b>automationStatus</b>	Object	<p>An automation-info object that contains details about the most recent start automation request for the workflow. The content of this property depends on the following factors:</p> <ul style="list-style-type: none"> <li>• If no automation was performed for the workflow, this property is null.</li> <li>• If automation processing is still in progress, this property indicates the step that is being processed.</li> <li>• If automation was restarted after it was stopped, this property indicates the status of the current start automation request.</li> <li>• If automation is stopped and the workflow status is complete, this property indicates that automation is completed.</li> <li>• If automation is stopped and the workflow status is not complete, this property identifies the step that is most closely related to the reason why automation was stopped.</li> </ul> <p><b>Notes about parallel-step workflows:</b></p> <ul style="list-style-type: none"> <li>– When a parallel-steps workflow is started, all of its automation ready steps are processed until they complete or fail, or automation is stopped. Failure of a step does not stop automation processing for other automation ready steps in the workflow.</li> <li>– In a parallel-steps workflow: <ul style="list-style-type: none"> <li>- The automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows.</li> <li>- If automation is currently stopped and the workflow is not yet complete, this property identifies the first uncompleted step that was returned to the Get Properties request.</li> </ul> </li> </ul> <p>Table 423 on page 743 lists the fields in the automation-info object.</p>
<b>autoDeleteOnCompletion</b>	Boolean	<p>Specifies whether the workflow is automatically deleted from the system after it completes successfully, as follows:</p> <ul style="list-style-type: none"> <li>• <b>false</b> means that the workflow is retained after it is complete, until it is removed by a user. A complete workflow is one in which all of its steps are marked complete or skipped.</li> <li>• <b>true</b> means that the workflow is automatically deleted from the system after it completes. As a result, the workflow is removed from the Workflows table in the Workflows task user interface.</li> </ul>

Table 422. JSON object that is returned to a get workflow properties request (continued)

Field name	Type	Description
<b>access</b>	String	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in <a href="#">“Workflow access type” on page 728</a> .  The following values are valid: <ul style="list-style-type: none"> <li>• Public</li> <li>• Restricted</li> <li>• Private</li> </ul>
<b>accountInfo</b>	String	For a workflow that submits a job, this property specifies the account information to use in the JCL JOB statement. This property can be null.
<b>jobStatement</b>	String	For a workflow that submits a job, this property specifies the JOB statement JCL that is used in the job. This property can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 - 8 characters.
<b>templateID</b>	String	Specifies the unique identifier for the template. Derived from a workflow internal variable, <code>\${_workflow-templateID}</code> .  This property is returned only for a provisioning workflow.
<b>actionID</b>	String	For an actions workflow, this property specifies the action ID for the action object. Derived from a workflow internal variable, <code>\${_workflow-actionID}</code> . For other types of provisioning workflows, this property is null.  This property is returned only for a provisioning workflow.
<b>registryID</b>	String	Specifies the ID of the software services registry. Derived from a workflow internal variable, <code>\${_workflow-registryID}</code> .  This property is returned only for a provisioning workflow.
<b>parentRegistryID</b>	String	Specifies the ID of the software instance parent registry entry. Derived from a workflow internal variable, <code>\${_workflow-parentRegistryID}</code> .  This property is returned only for a provisioning workflow.
<b>domainID</b>	String	Specifies the ID of the domain that is associated with the template. Derived from a workflow internal variable, <code>\${_workflow-domainID}</code> .  This property is returned only for a provisioning workflow.
<b>tenantID</b>	String	Specifies the ID of the tenant that is associated with the resource pool. Derived from workflow internal variable, <code>\${_workflow-tenantID}</code> .  This property is returned only for a provisioning workflow.

Table 422. JSON object that is returned to a get workflow properties request (continued)		
Field name	Type	Description
<b>software-service-instance-name</b>	String	Specifies the created software service instance name. Derived from workflow internal variable, <code>\${_workflow-softwareServiceInstanceName}</code> .  This property is returned only for a provisioning workflow.
<b>templateName</b>	String	Specifies the name of the template that is associated with the resource pool. Derived from workflow internal variable, <code>\${_workflow-templateName}</code> .  This property is returned only for a provisioning workflow.
<b>globalVariableGroup</b>	String	Global variable group for the workflow.
<b>isInstanceVariableWithoutPrefix</b>	String	Indicates whether the simplified format is used for references to instance variables. If true, variable references are simplified; they omit the prefix <code>instance-</code> . If false, variable references must include the prefix <code>instance-</code> .
<b>steps</b>	Array of objects	Array of one or more step-info objects that contain details about each of the steps in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code> .  The content of this array depends on what the requestor is permitted to see. For more information, see <a href="#">“Description” on page 736</a> .  Table 424 on page 745 lists the fields in the step-info object.
<b>variables</b>	Array of objects	Array of one or more variable-info objects that contain details about the variables that are used in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>variables</code> .  The content of this array depends on what the requestor is permitted to see. For more information, see <a href="#">“Description” on page 736</a> .  Table 429 on page 755 lists the fields in the variable-info object.

## Format of the automation-info object

Table 423 on page 743 lists the fields in the automation-info JSON object.

Table 423. Get Workflow Properties request: Format of the automation-info object		
Field name	Type	Description
<b>startUser</b>	String	User ID of the user who initiated the automation processing.
<b>startedTime</b>	Timestamp	Time that automation processing started. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time that is expressed as the number of milliseconds since midnight on January 1, 1970 UTC.

Table 423. Get Workflow Properties request: Format of the automation-info object (continued)		
Field name	Type	Description
<b>stoppedTime</b>	Timestamp	Time that automation processing stopped. If automation is still in progress, this property is set to null. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time that is expressed as the number of milliseconds since midnight on January 1, 1970 UTC.
<b>currentStepName</b>	String	Depending on the current phase of automation processing, this property contains one of the following values: <ul style="list-style-type: none"> <li>Name of the step that is being processed automatically.</li> <li>Name of the step that caused automation to stop.</li> <li>For a workflow that uses parallel processing (a <i>parallel-steps workflow</i>) this value is the name of the first step that is incomplete.</li> </ul> If automation is stopped and the workflow status is complete, this property is set to null.
<b>currentStepNumber</b>	String	The step number. If automation is stopped and the workflow status is complete, this property is set to null.
<b>currentStepTitle</b>	String	Step title. If automation is stopped and the workflow status is complete, this property is set to null.
<b>messageID</b>	String	Message identifier for the accompanying message. If automation is still in progress, this property is set to null.
<b>messageText</b>	String	Message text that describes the reason that automation is stopped. If automation is still in progress, this property is set to null.

## Format of the step-info object

Table 424 on page 745 lists the fields in the step-info JSON object. Not all of the properties are returned for every step. Some properties are returned or omitted depending on the step type, as noted in the **When returned** column. This information in this column indicates whether valid data is returned for the step, as follows:

### All step types

Properties that are returned for all step types.

### Calling steps

Properties that are returned for a step that calls another workflow for execution.

### Template steps

Properties that are returned for a step that runs a program, such as a batch job, REXX exec, or UNIX shell script.

### REST steps

Properties that are returned for a step that issues a REST API request, such as a GET or PUT request.

Regarding returned data, a template step and a REST step are mutually exclusive. The returned information for a template step does not include the properties for a REST step. Likewise, the returned information for a REST step does not include the properties for a template step. To help you identify which steps are REST steps, the step-info object includes the `isRestStep` property, set to `true` or `false`.

Table 424. Get Workflow Properties request: Format of the step-info object			
Field name	Type	When returned	Description
<b>name</b>	String	All step types	Name of the step.
<b>actualStatusCode</b>	String	REST steps only	The actual HTTP status code that is received from the REST API request. To obtain this value, map it to a workflow variable.
<b>assignees</b>	String	Calling steps and template steps	Step assignees. One or more user IDs that are assigned to the step. Multiple items are separated by commas.
<b>autoEnable</b>	Boolean	All step types	Indicates whether the step can be performed automatically when all prerequisite steps are completed, and no user inputs are required.
<b>calledInstanceKey</b>	String	Calling steps only	For a step that calls another workflow for execution, this property contains the key of the called workflow instance. You can use this value to retrieve information about the called workflow.  This property is null until the step is performed and either a new instance of the called workflow is created or an existing instance is found.
<b>calledInstanceScope</b>	String	Calling steps only	For a step that calls another workflow for execution, this property contains the scope of the called workflow instance. See <a href="#">“Setting the workflow scope” on page 815</a> for more information.
<b>calledInstanceURI</b>	String	Calling steps only	For a step that calls another workflow for execution, this property contains the URI path of the called workflow instance. You can use this value to retrieve information about the called workflow.  This property is null until the step is performed and either a new instance of the called workflow is created or an existing instance is found.
<b>calledWorkflowID</b>	String	Calling steps only	Specifies the workflow ID of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property calledWorkflowMD5 is specified.
<b>calledWorkflowVersion</b>	String	Calling steps only	Specifies the workflow version of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed.  This property: <ul style="list-style-type: none"> <li>• Is null when the property calledWorkflowMD5 is specified</li> <li>• Might be null when the property calledWorkflowID is specified.</li> </ul>

Table 424. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>calledWorkflowMD5</b>	String	Calling steps only	Specifies the 128-bit hash value of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property calledWorkflowID is specified.
<b>calledWorkflowDescription</b>	String	Calling steps only	Describes the workflow to be called, from the point of view of the calling workflow.
<b>calledWorkflowDefinitionFile</b>	String	Calling steps only	Specifies the name of the workflow definition file that is used to create a new workflow if an existing instance is not found when this step is performed. This property might be null.
<b>description</b>	String	All step types	Step description.
<b>expectedStatusCode</b>	String	REST steps only	The expected HTTP status code from the REST API request. If the expectedStatusCode value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a template step returns a return code that is greater than the allowed maximum return code.
<b>failedPattern</b>	Array of strings	Template steps only	Optional regular expression that can be returned for program execution failures. This property might be null.
<b>hasCalledWorkflow</b>	Boolean	Calling steps and template steps	Indicates whether this step calls another workflow (true or false). If true, this step is a "calling" step, that is, it calls another workflow for execution. If false, it is a template step.  This property is returned only when steps=null, which indicates a leaf step.
<b>hostname</b>	String	REST steps only	Indicates the hostname or IP address of the site to which the REST request is directed. For example: <code>www.ibm.com</code> .
<b>httpMethod</b>	String	REST steps only	Indicates the HTTP method that is used for issuing the REST API request. The possible values are: <ul style="list-style-type: none"> <li>• GET</li> <li>• PUT</li> <li>• POST</li> <li>• DELETE</li> </ul>
<b>instructions</b>	String	Template steps only	Detailed instructions on what the user must do to perform the step.
<b>instructionsSub</b>	Boolean	Template steps only	Indicates whether the step instructions contain variables (true or false).

Table 424. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>isConditionStep</b>	Boolean	Calling steps and template steps	Indicates whether this step is a conditional step (true or false).
<b>isRestStep</b>	Boolean	All step types	<p>Indicates whether this step is a REST API step (true or false).</p> <p>When set to true, the following properties contain details about the REST request. Otherwise, these properties are set to null.</p> <ul style="list-style-type: none"> <li>• actualStatusCode</li> <li>• expectedStatusCode</li> <li>• hostname</li> <li>• hostnameSub</li> <li>• httpMethod</li> <li>• port</li> <li>• portSub</li> <li>• queryParameters</li> <li>• queryParametersSub</li> <li>• requestBody</li> <li>• requestBodySub</li> <li>• schemeName</li> <li>• schemeNameSub</li> <li>• uriPath</li> <li>• uriPathSub</li> </ul> <p>The following step properties are not applicable for a REST step and thus, are omitted from the output:</p> <ul style="list-style-type: none"> <li>• template</li> <li>• templateSub</li> <li>• output</li> <li>• outputSub</li> <li>• saveAsDataset</li> <li>• saveAsDatasetSub</li> <li>• saveAsUnixFile</li> <li>• saveAsUnixFileSub</li> <li>• submitAs</li> <li>• maxLrecl</li> <li>• returnCode</li> </ul>

Table 424. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>jobInfo</b>	JSON object	Template steps only	<p>For a step that submits a job, this property contains the jobInfo object, which contains details about the job. Otherwise, this property is null.</p> <p>This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code>.</p> <p><a href="#">“Format of the jobInfo object” on page 752</a> describes the fields in the jobInfo object.</p>
<b>maxLrecl</b>	Integer	Template steps only	For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job. This value is an integer 80 - 1024. The default is 1024.
<b>optional</b>	Boolean	All step types	Indicates whether the step is optional (true or false).
<b>output</b>	String	Template steps only	Indicates the name of the output file that is produced by the step (a data set or UNIX file). The output file can contain variables and values that are used by subsequent steps.
<b>outputSub</b>	Boolean	Template steps only	Indicates whether the output file name contains variable substitution (true or false).
<b>outputVariablesPrefix</b>	String	Template steps only	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.
<b>owner</b>	String	Calling steps and template steps	User ID of the step owner.
<b>port</b>	String	REST steps only	Port number that is associated with the REST request.
<b>portSub</b>	Boolean	REST steps only	Indicates whether the port number contains variable substitution (true or false).
<b>prereqStep</b>	Array of strings	All step types	Lists the names of the steps that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.
<b>procName</b>	String	Template steps only	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.

Table 424. Get Workflow Properties request: Format of the step-info object (continued)			
Field name	Type	When returned	Description
<b>queryParameters</b>	String	REST steps only	For a REST request that includes query parameters, this property contains the query parameters. Otherwise, this property is null.
<b>queryParametersSub</b>	Boolean	REST steps only	Indicates whether the query parameters contain variable substitution (true or false). Otherwise, this property is null.
<b>regionSize</b>	String	Template steps only	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value was specified for the step, the default is 50000.
<b>requestBody</b>	String	REST steps only	For a REST request that includes a request body, this property contains the request body. Otherwise, this property is null.
<b>requestBodySub</b>	Boolean	REST steps only	Indicates whether the request body variable substitution (true or false). Otherwise, this property is null.
<b>returnCode</b>	String	Template steps only	For a step that submits a job to run, this property indicates the return code that was returned when the job was submitted.
<b>runAsUser</b>	String	All step types	<p>The user ID under which the step is to be performed (the runAsUser ID).</p> <p>When the property <code>runAsUserDynamic</code> is true, the runAsUser ID is determined by using variable substitution when the step is performed.</p> <p>This property is omitted if the runAsUser element is not specified for the step.</p>
<b>runAsUserDynamic</b>	Boolean	All step types	<p>Indicates whether the runAsUser ID value can change:</p> <p><b>true</b> The runAsUser ID value is not final and can change during the processing of the workflow. Its value is determined by using variable substitution when the step is performed.</p> <p><b>false</b> The runAsUser ID is final and cannot change during the processing of the workflow. Its value is determined when the workflow is created.</p> <p>This property is omitted if the runAsUser element is not specified for the step.</p>
<b>saveAsDataset</b>	String	Template steps only	Data set name (fully qualified, no quotation marks) that contains the saved JCL.

Table 424. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>saveAsDatasetSub</b>	Boolean	Template steps only	Indicates whether the data set name contains variable substitution (true or false).
<b>saveAsUnixFile</b>	String	Template steps only	UNIX file name (absolute name) that contains the saved JCL.
<b>saveAsUnixFileSub</b>	Boolean	Template steps only	Indicates whether the UNIX file name contains variable substitution (true or false).
<b>schemeName</b>	String	REST steps only	The scheme name that is used for the REST request. For example: http.
<b>schemeNameSub</b>	Boolean	REST steps only	Indicates whether the scheme name contains variable substitution (true or false).
<b>scriptParameters</b>	Array of strings	Template steps only	For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.
<b>skills</b>	String	Calling steps and template steps	The type of skills that are required to perform the step.

Table 424. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>state</b>	String	All step types	<p>State of the step. One of the following status indicators is displayed:</p> <ul style="list-style-type: none"> <li>• <b>Unassigned.</b> The step is in the <i>Unassigned</i> state; no users or groups are assigned to the step.</li> <li>• <b>Assigned.</b> Users or groups are assigned to the step, but no user accepted ownership of the step.</li> <li>• <b>Not Ready.</b> User accepted ownership of the step. However, a prerequisite step must be completed or a conditional dependency must be satisfied before the step can be performed.</li> <li>• <b>Ready.</b> The step is ready to be performed; all prerequisite steps and conditional dependencies are satisfied.</li> <li>• <b>In Progress.</b> The step is in progress. For a parent step, a state of <i>In Progress</i> means that at least one of the child steps is started, but is not yet complete, overridden, or skipped. For a leaf step, a state of <i>In Progress</i> means that the step is started, but is not yet complete, overridden, or skipped.</li> <li>• <b>Submitted.</b> The step included a job, which the step owner submitted.</li> <li>• <b>Complete.</b> The step was completed.</li> <li>• <b>Skipped.</b> The step was bypassed by the step assignee.</li> <li>• <b>Complete (Override).</b> The step was marked complete, but the work was performed outside of the Workflows task.</li> <li>• <b>Failed.</b> The step included a job that was submitted by the step owner. However, the job failed to complete successfully.</li> <li>• <b>Conflicts.</b> The step created an output file for use by a subsequent step. However, values in that file conflict with existing instance or global variables.</li> <li>• <b>Condition Not Satisfied.</b> The step is a conditional step, and the condition is not satisfied.</li> </ul>
<b>stepNumber</b>	String	All step types	The step number. Steps are numbered to indicate the sequence in which steps are to be performed. For example, the first step in a workflow is 1.
<b>steps</b>	Array of objects	All step types	<p>For a parent step, this is a nested array of step-info objects. For a leaf step, this property is null.</p> <p>Check this property first before you check the other, non-common step properties. A non-null value here means that the calling step properties are omitted, as are the template step properties and the REST step properties.</p>

Table 424. Get Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>submitAs</b>	String	Template steps only	Indicates the type of executable program: JCL job, a REXX exec, or a UNIX shell script, which includes a REXX exec that is written for the UNIX shell environment. The possible values are the following: <ul style="list-style-type: none"> <li>• "JCL "</li> <li>• "TSO-REXX"</li> <li>• "shell-JCL "</li> <li>• "TSO-REXX-JCL "</li> <li>• "TSO-UNIX-REXX"</li> <li>• "TSO-UNIX-shell "</li> </ul>
<b>successPattern</b>	String	Template steps only	Regular expression that is returned for a successful program execution.
<b>template</b>	String	Template steps only	Indicates the template that is used to run a program or batch job (inline or external file).
<b>templateSub</b>	Boolean	Template steps only	Indicates whether template contains variable substitution (true or false). The default is false.
<b>timeout</b>	String	Template steps only	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it is ended by a timeout condition.
<b>title</b>	String	All step types	Step title.
<b>uriPath</b>	String	REST steps only	The URI path to use for the REST request.
<b>uriPathSub</b>	Boolean	REST steps only	Indicates whether the URI path contains variable substitution (true or false).
<b>userDefined</b>	Boolean	All step types	Indicates whether the step was added manually to the workflow (true or false). If true, the step was added by the workflow owner, using the <b>Update Workflow Steps</b> action in the Workflows table. If false, the step was defined in the workflow definition that was used to create the workflow.
<b>variable-references</b>	Array of objects	Template steps only	An array of variable-reference objects, the format of which is described in <a href="#">Table 428 on page 755</a> .
<b>weight</b>	Integer	Calling steps and template steps	The relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).

## Format of the jobInfo object

Table 425 on page 753 lists the fields in the jobInfo JSON object.

Table 425. Get Workflow Properties request: Format of the jobInfo object		
Field name	Type	Description
<b>jobstatus</b>	JSON object	Contains the jobstatus object, which contains details about the job. Otherwise, this property is null.  “Format of the jobstatus object” on page 753 describes the fields in the jobstatus object.
<b>jobfiles</b>	Array	Contains an array of one or more objects that contain details about each of the files that are created by the job. Otherwise, this property is null.  “Format of the jobfiles object” on page 754 lists the fields in the jobfiles object.

## Format of the jobstatus object

Table 426 on page 753 lists the fields in the jobstatus JSON object.

Table 426. Get Workflow Properties request: Format of the jobstatus object		
Field name	Type	Description
<b>retcode</b>	String	Job completion code. One of the following values:  <b>ABENDUnnnn</b> Job ended with the user abend code <i>nnnn</i> .  <b>ABEND Sxxx</b> Job ended with the system abend code <i>xxx</i> .  <b>CANCELED</b> Job was canceled.  <b>CC nnnn</b> Job ended with the completion code <i>nnnn</i> .  <b>CONV ABEND</b> Converter ended abnormally when processing the job.  <b>CONV ERROR</b> Converter error when processing the job.  <b>JCL ERROR</b> Job encountered a JCL error.  <b>SEC ERROR</b> Job failed a security check.  <b>SYS FAIL</b> System failure.  If this value is null, the job has not yet completed.
<b>jobname</b>	String	Job name.

Table 426. Get Workflow Properties request: Format of the jobstatus object (continued)

Field name	Type	Description
<b>status</b>	String	Job status. One of the following values: <b>INPUT</b> Job is in input processing. <b>ACTIVE</b> Job is running. <b>OUTPUT</b> Job is on the hardcopy output queue. If this value is null, the job status could not be determined.
<b>owner</b>	String	The z/OS user ID associated with the job.
<b>subsystem</b>	String	The primary or secondary JES subsystem. If this value is null, the job was processed by the primary subsystem.
<b>class</b>	String	Job execution class.
<b>type</b>	String	Job type. One of the following values: <b>JOB</b> Batch job. <b>STC</b> Started task. <b>TSU</b> TSO/E user.
<b>jobid</b>	String	Job identifier.

## Format of the jobfiles object

Table 427 on page 754 lists the fields in the jobfiles JSON object.

Table 427. Get Workflow Properties request: Format of the jobfiles object

Field name	Type	Description
<b>id</b>	Integer	Data set number (key).
<b>ddname</b>	String	DDNAME for the data set creation.
<b>byte-count</b>	Integer	Number of bytes on spool that is consumed by the spool file. The value can be zero (0).
<b>record-count</b>	Integer	Number of records in the spool file. The value can be zero (0).
<b>class</b>	String	Class that is assigned to the spool file.
<b>stepname</b>	String	Step name for the step that created this data set. The value can be null.
<b>procstep</b>	String	Procedure name for the step that created this data set. The value can be null.

## Format of the variable-reference object

Table 428 on page 755 lists the fields in the variable-reference JSON object.

Table 428. Get Workflow Properties request: Format of the variable-reference object		
Field name	Type	Description
<b>name</b>	String	Name of the variable.
<b>scope</b>	String	Variable scope, which is either instance or global.

## Format of the variable-info object

Table 429 on page 755 lists the fields in the variable-info JSON object.

Table 429. Get Workflow Properties request: Format of the variable-info object		
Field name	Type	Description
<b>name</b>	String	Name of the variable.
<b>scope</b>	String	Variable scope, which is either instance or global.
<b>type</b>	String	Type of variable, which is one of the following values: <ul style="list-style-type: none"> <li>• boolean</li> <li>• string</li> <li>• number</li> <li>• date</li> <li>• time</li> <li>• array</li> </ul>
<b>value</b>	String	Variable value.
<b>visibility</b>	String	Public or private.

## Example HTTP interaction

In the following example, the GET method is used to retrieve information about a workflow. The workflow is uniquely identified by the workflow key, which is represented by the following string value: 26f1fd84-058b-443c-8e06-5ec318ecdb86. The query parameter `returnData=steps,variables` is included to request more information about the workflow steps and variables.

```
GET /zosmf/workflow/rest/1.0/workflows/26f1fd84-058b-443c-8e06-5ec318ecdb86?returnData=steps,variables
HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 378. Sample request to get workflow properties

An example of the response is shown in the figures that follow.







## List the workflows for a system or sysplex

You can use this operation to list the z/OSMF workflows for a system or sysplex.

### HTTP method and URI path

---

```
GET /zosmf/workflow/rest/<version>/workflows
```

---

In this request, the URI path variable *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.

### Query parameters

Optionally, your request can include one or more of the following query parameters to filter the results:

**workflowName**

Workflow name. You can specify a regular expression here to match desired workflow names.

**category**

Category of the workflow, which is either general or configuration.

**system**

Nickname of the system on which the workflow is to be performed.

**statusName**

Workflow status, which can be one of the following values:

- in-progress
- complete
- automation-in-progress
- canceled

**owner**

Workflow owner (a valid z/OS user ID).

**vendor**

Name of the vendor that provided the workflow definition file.

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark ('?') to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).

### Description

This operation retrieves a list of workflows that match your search criteria. You can filter the returned list of workflows through the specification of query parameters. You can, for example, limit the results to workflows by name, or the workflows for a particular z/OS system.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 430 on page 760](#).

### Authorization requirements

See [“Authorization requirements” on page 726](#).

For the requestor of this service, the response data is limited by the workflow *access type*. This value is selected by the workflow owner at workflow creation time. All workflows with a public access type are

included in the response. To list restricted or private access type workflows, however, the requestor user ID must be a workflow owner, step owner, or step assignee.

For more information, see [“Workflow access type” on page 728](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, a non-successful HTTP status code is returned and the response body is a standard error response body with the message ID and the associated error message. For the codes that can be returned, see [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the response body contains one property, called workflows. This property is an array of workflow-info objects. [Table 430 on page 760](#) lists the fields in the workflow-info object. If no workflows match the filter criteria, HTTP status code 200 (OK) is returned with an empty array.

Table 430. List workflows request: Format of the workflow-info object		
Field name	Type	Description
<b>workflowName</b>	String	Descriptive name for the workflow.
<b>workflowKey</b>	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
<b>workflowDescription</b>	String	Description of the workflow.
<b>workflowID</b>	String	Workflow ID. A short, arbitrary value that identifies the workflow.
<b>workflowVersion</b>	String	Version of the workflow definition file.
<b>workflowDefinitionFileMD5Value</b>	String	The 128-bit hash value that is associated with the workflow definition file that was used to create the workflow.
<b>instanceURI</b>	String	Workflow instance URI path, which you can use to retrieve information about the workflow.
<b>owner</b>	String	User ID of the workflow owner.
<b>vendor</b>	String	Name of the vendor that provided the workflow definition file.
<b>access</b>	String	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in <a href="#">“Workflow access type” on page 728</a> .  The following values are valid: <ul style="list-style-type: none"><li>• Public</li><li>• Restricted</li><li>• Private</li></ul>

## Example HTTP interaction

In the following example, the GET method is used to list the workflows on a system. Here, the query parameter workflowName is included to limit the results to workflows with names that begin with AutomationExample.

```
GET /zosmf/workflow/rest/1.0/workflows?workflowName=AutomationExample.* HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

*Figure 382. Sample request to list workflows*

For a successful request, the HTTP response includes a JSON document that contains the requested information. In the following example, one workflow was found to match the request query parameter.

```
HTTP/1.1 200 OK
content-length: 464
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:34 GMT
content-type: application/json; charset=UTF-8

{
  "workflows": [
    {
      "instanceURI": "/zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0",
      "owner": "zosmfad",
      "vendor": "IBM",
      "workflowDefinitionFileMD5Value": "a8825b7497793bc620b0edffa8b97cd9",
      "workflowDescription": "Sample demonstrating the use of automated steps in workflow.",
      "workflowID": "automationSample",
      "workflowKey": "d043b5f1-adab-48e7-b7c3-d41cd95fa4b0",
      "workflowName": "AutomationExample|Canceled|1423679433714",
      "workflowVersion": "1.0",
      "access": "Public"
    }
  ]
}
```

*Figure 383. Sample response from a list workflows request*

## Start a workflow

You can use this operation to start a z/OSMF workflow on a z/OS system. The workflow to be started must contain at least one automated step.

### HTTP method and URI path

```
PUT /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/start
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the workflow to be started.

### Query parameters

None.

### Description

This operation is used to start an automated workflow, that is, a workflow with at least one step that can be performed automatically. For information about designing an automated workflow, see [“Automated steps” on page 846](#).

On successful completion, HTTP status code 202 (Accepted) is returned. If you specify a notification URL in your request, a JSON object is posted to the URL when automation stops; the format of the object is described in [“Content that is posted to the notification URL” on page 765](#).

This request must be submitted from the user ID of the step owner for the automated step. Otherwise, the request ends with HTTP status code 400 and an error message. Also, if used with a workflow that contains no automated steps, the request ends with HTTP status code 400 and an error message.

## Parallel-steps workflows

A workflow with steps that can be run in parallel (concurrently) is called a *parallel-steps workflow*. When you start a parallel-steps workflow, the Workflows task locates the automation ready steps and attempts to run them concurrently.

A step is considered to be *automation ready* when it is:

- Enabled for automation. In the workflow definition file, the attribute `autoEnable=true` is specified on the step element (`<step>`).
- In an eligible state: *ready*, *in-progress*, or *failed*. For a failed step, the Workflows task performs the step again.

To use parallel processing, an automated workflow must include the attribute `parallelSteps=true` in the workflow metadata. Otherwise, if this attribute is set to `false` or omitted, automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.

In a parallel-steps workflow, the failure of an automated step does not stop automation processing for the other automated steps. Processing continues until all of the automated steps are completed or failed, or a condition occurs that stops automation processing, such as a user stopping automation by using the **Stop Automation** action in the Workflows task.

**Note:** The ability to suspend step processing is mutually exclusive with the ability to run steps in parallel. Therefore, if a workflow includes the suspend element (`<suspend>`) in the step definition, it is not eligible for parallel processing. The Workflows task enforces this restriction. An attempt to start a workflow that contains both the suspend element (`<suspend>`) and the attribute `parallelSteps=true` will result in an error.

## Request content

Your request can contain a JSON object that specifies options for starting the workflow. The request content is optional; it can be omitted or be empty. If so, default values are used for any optional properties that can be specified.

[Table 431 on page 763](#) lists the fields in the JSON object.

Table 431. Request content for the start workflow request

Field name	Type	Required or optional	Description
<b>resolveConflictByUsing</b>	String	Optional	<p>Indicates how variable conflicts, if any, are to be handled when the Workflows task reads in the output file from a step that runs a REXX exec or UNIX shell script.</p> <p>The following values are valid for this parameter:</p> <p><b>outputFileValue</b> Allow the output file values to override the existing values. This setting applies to instance variables only; global variables are not overridden by variables in the output file.</p> <p><b>existingValue</b> Use the existing variables values instead of the output file values.</p> <p><b>leaveConflict</b> Automation is stopped. The user must resolve the conflict manually.</p> <p>If you omit this property, the default is <code>outputFileValue</code>.</p>
<b>stepName</b>	String	Optional	<p>The name of the step at which automation is to begin. If you omit this property, automation processing begins with the first step in the workflow.</p> <p>In a parallel-steps workflow, the automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows. Therefore, automation cannot be directed to start at a specific step. If you specify this property for a parallel-steps workflow, the property is accepted but ignored.</p>
<b>performSubsequent</b>	Boolean	Optional	<p>If the workflow contains any subsequent automated steps, this property indicates whether z/OSMF is to perform the steps. If you set this property to <code>true</code>, or omit the property, z/OSMF attempts to perform the automated steps. If you set this property to <code>false</code>, z/OSMF attempts to perform the specified step only. The default is <code>true</code>.</p> <p>In a parallel-steps workflow, the automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows. Therefore, automation cannot be directed to start at a specific step. If you specify this property for a parallel-steps workflow, the property is accepted but ignored.</p>

Table 431. Request content for the start workflow request (continued)			
Field name	Type	Required or optional	Description
<b>notificationUrl</b>	String	Optional	<p>A notification URL (up to 2000 characters).</p> <p>Depending on the requirements of your application, you might want to receive a notification when the automated step or steps reach an end state (either completion or failure). Suppose, for example, that your application includes a servlet that is to be given control on completion of the automated steps. If so, you can optionally specify a notification URL to be posted when automation ends.</p> <p>Your installation must add this URL to the list of allowable sites. Work with your security administrator to create the appropriate authorization; see <a href="#">“Allowing cross-site access to REST services”</a> on page 5.</p> <p>If specified, this URL receives a JSON object with the result of the automation processing. For the format of the JSON object, see <a href="#">Table 433 on page 765</a>.</p>
<b>targetSystemuid</b>	String	Optional	The user ID to be used for remote system basic authentication.
<b>targetSystempwd</b>	String	Optional	The password to be used for remote system basic authentication.

## Authorization requirements

See [“Authorization requirements”](#) on page 726.

## HTTP status codes

On successful completion, HTTP status code 202 (Accepted) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 432. HTTP error response codes for a start workflow request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	<p>The request contained incorrect parameters. For example:</p> <ul style="list-style-type: none"> <li>Incorrect value is specified for the property <code>resolveConflictByUsing</code>.</li> <li>Specified workflow contains no automated steps.</li> <li>Request is submitted from a user ID that is not the step owner for the automated step. For a parallel-steps workflow, the request is submitted from a user ID that is not the step owner for any of the automated steps in the workflow.</li> <li>Automation processing attempts to begin at a step that is not automated.</li> <li>Incorrect value is specified for the property <code>stepName</code>.</li> </ul>

Table 432. HTTP error response codes for a start workflow request (continued)	
HTTP error status code	Description
<b>HTTP 404 Not found</b>	The specified workflow key was not found; the workflow does not exist.
<b>HTTP 409 Request conflict</b>	Request cannot be performed automatically because the specified workflow has one of the following statuses: <ul style="list-style-type: none"> <li>Automation-in-progress</li> <li>Canceled</li> <li>Complete (for a parallel-steps workflow). A start request is not applicable for a parallel-steps workflow that is complete.</li> </ul>
<b>HTTP 500 Internal server error</b>	The server encountered an error. See the response body for a JSON object with information about the error.

Additional standard status codes can be returned, as described in [“HTTP status codes”](#) on page 727.

## Content that is posted to the notification URL

If you specified a notification URL in your request, the URL receives a JSON object when automation stops. Table 433 on page 765 lists the fields in the JSON object.

Table 433. Structure of the JSON object that is returned to the notification URL		
Field name	Type	Description
<b>instanceURI</b>	String	Specifies the URI path of the workflow that was being automated. You can use the URI path to retrieve more details about the workflow.
<b>statusName</b>	String	Indicates the current workflow status after automation has stopped.  The following are the possible values that might be returned after a stopped automation:  <b>in-progress</b> The workflow automation is stopped. The reason is indicated in the messageID and messageText properties.  <b>complete</b> Workflow is complete. All steps are marked complete or skipped.
<b>startUser</b>	String	User ID of the user who initiated the automation processing.
<b>startedTime</b>	Timestamp	Time that automation processing started.
<b>stopUser</b>	String	User ID of the user who stopped automation.
<b>stoppedTime</b>	Timestamp	Time that automation processing stopped.
<b>currentStepName</b>	String	Name of the step that caused automation to stop. If the workflow status is complete, this property is set to null.
<b>currentStepNumber</b>	String	The step number. If the workflow status is complete, this property is set to null.

Table 433. Structure of the JSON object that is returned to the notification URL (continued)		
Field name	Type	Description
<b>currentStepTitle</b>	String	Step title. If the workflow status is complete, this property is set to null.
<b>messageID</b>	String	This property contains the message identifier that helps identify the reason that automation is stopped.
<b>messageText</b>	String	This property contains the message text associated with the message identifier found in messageID.

## Example HTTP interaction

In Figure 384 on page 766, a request is submitted to start a workflow. Here, the workflow is identified by the workflow key, which is the following string value: d043b5f1-adab-48e7-b7c3-d41cd95fa4b0. In this example, the request content is empty. As a result, default values are used for any properties that can be specified; see Table 431 on page 763.

```
PUT /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0/operations/start HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Content-Type: application/json
Content-Length: 3
Authorization: Basic em9zbWZhZDp6b3NtZmFk

{}
```

Figure 384. Sample request to start a workflow

A sample response is shown in Figure 385 on page 766.

```
HTTP/1.1 202 Accepted
content-length: 0
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:29:55 GMT
content-type: application/json; charset=UTF-8
```

Figure 385. Sample response from a start workflow request

## Cancel a workflow

You can use this operation to cancel a z/OSMF workflow on a z/OS system.

### HTTP method and URI path

```
PUT /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/cancel
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the workflow to be canceled.

### Query parameters

None.

## Description

This operation is used to cancel a workflow. Canceling a workflow does not undo any actions that were already performed on the system as part of the workflow.

When canceled, the workflow cannot be resumed. You can view the workflow properties through a GET request, as described in [“Get the properties of a workflow” on page 736](#). Also, you can delete a canceled workflow, as described in [“Delete a workflow” on page 768](#).

When canceled, the name of workflow is changed. The name is appended with the text |Canceled|, followed by a timestamp (the date and time expressed in milliseconds since midnight on January 1, 1970 UTC). An example is shown in [“Example HTTP interaction” on page 767](#).

This request is failed for a workflow with the status *Automation in Progress*. That is, it is not possible to cancel the workflow while an automated step is running. You must either allow the processing to complete, or you can stop the processing through the Workflows task.

## Request content

None.

## Authorization requirements

This request is available to the workflow owner only. A cancel request from another user is rejected with HTTP status code 403 (Forbidden) and an appropriate error message in the JSON response object.

For other authorization requirements, see [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned. Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 434. HTTP error response codes for a cancel workflow properties request	
HTTP error status code	Description
HTTP 403 Forbidden	The request was submitted from a user ID that is not the workflow owner.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.
HTTP 409 Request conflict	The request cannot be processed because the workflow has the status <i>Automation in Progress</i> .

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the response body contains one property, `workflowName`, which specifies the new name of the canceled workflow.

## Example HTTP interaction

In the following example, the PUT method is used to cancel an instance of a workflow. Here, the workflow is identified by the workflow key, which is the following string value: `d043b5f1-adab-48e7-b7c3-d41cd95fa4b0`.

```
PUT /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0/operations/cancel HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Content-Type: application/json
Content-Length: 0
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

*Figure 386. Sample request to cancel a workflow*

For a successful request, HTTP response code 200 is returned with the canceled workflow name in response body.

```
HTTP/1.1 200 OK
content-length: 59
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:33 GMT
content-type: application/json; charset=UTF-8

{
  "workflowName": "AutomationExample|Canceled|1423679433714"
}
```

*Figure 387. Sample response from a cancel workflow request*

## Delete a workflow

You can use this operation to remove a z/OSMF workflow from a z/OS system.

### HTTP method and URI path

```
DELETE /zosmf/workflow/rest/<version>/workflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- *<workflowKey>* identifies the workflow to be deleted.

### Query parameters

None.

### Description

This operation is used to delete a workflow from z/OSMF, including any notes that accompany the workflow and its steps, and the history log for the workflow. Deleting a workflow does not undo any actions that were performed on the system as part of the workflow. If you delete a workflow, you are responsible for undoing manually any changes on the system that you no longer require. Ensure that all applicable back-out procedures are followed. See your workflow provider for this information.

This request is failed for a workflow with the status *Automation in Progress*. That is, it is not possible to delete the workflow while an automated step is running. You must either allow the processing to complete, or you can stop the processing through the Workflows task.

You cannot delete a *called workflow*, that is, a workflow that is in-progress as a result of being called by another workflow for execution. For design considerations for called workflows, see [“Calling steps” on page 840](#).

## Request content

None.

## Authorization requirements

For a general workflow or configuration workflow, the ability to delete the workflow is limited to the current workflow owner and members of the z/OSMF workflow administrators group. For a provisioning workflow, the domain administrator is also able to delete a workflow. A delete request from another user is rejected with the HTTP status code 403 (Forbidden) and an appropriate error message in the JSON response object.

For other authorization requirements, see [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 435. HTTP error response codes for a delete workflow request	
HTTP error status code	Description
HTTP 403 Forbidden	The request was submitted from a user ID that is not the workflow owner.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.
HTTP 409 Request conflict	The request cannot be processed because the workflow has the status <i>Automation in Progress</i> .

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

None.

## Example HTTP interaction

In the following example, the DELETE method is used to delete a workflow. Here, the workflow is identified by the workflow key, which is the following string value: d043b5f1-adab-48e7-b7c3-d41cd95fa4b0.

```
DELETE /zosmf/workflow/rest/1.0/workflows/d043b5f1-adab-48e7-b7c3-d41cd95fa4b0 HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 388. Sample request to delete a workflow

For a successful request, the HTTP response 204 is returned.

```
HTTP/1.1 204 No Content
content-length: 0
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:34 GMT
content-type: application/json; charset=UTF-8
```

*Figure 389. Sample response from a delete workflow request*

## Retrieve a workflow definition

You can use this operation to retrieve the contents of a z/OSMF workflow definition from a z/OS system.

### HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/workflowDefinition
```

In this request, the URI path variable `<version>` identifies the version of the z/OSMF workflow service. The following value is valid: `1.0`.

### Query parameters

You can specify the following query parameters on this request:

#### **definitionFilePath**

Specifies the location of the workflow definition file, which is either a UNIX path name (including the file name) or a fully qualified z/OS data set name. This parameter is required.

#### **workflowDefinitionFileSystem**

Nickname of the system on which the specified workflow definition file and any related files reside. The Workflows task obtains the workflow files from this system.

Use the nickname that is specified for the system definition in the z/OSMF Systems task. The nickname is a unique name for the system to differentiate it from existing systems that have the same system and sysplex name. The nickname is 1 - 40 characters long; the valid characters are alphanumeric characters (A-Z, a-z, and 0-9), hyphens (-), and special characters (\$ \_ # @). Nicknames are case-sensitive; for example, `SYSTEM1` and `System1` are unique values.

The system can be running in the local sysplex or in another sysplex in your enterprise. If you select a system in a remote sysplex, verify that the system is enabled for single sign-on (SSO). Otherwise, your request must include a valid user ID and password (in the request body) for basic authentication with the remote system.

If you omit this parameter, the Workflows task checks the z/OSMF system of the local sysplex for the workflow definition file and related files, by default.

#### **returnData**

Use this optional parameter to request more information about the workflow definition file. Include one or both of the following attributes on the `returnData` parameter:

##### **steps**

Returns an array of step-definition objects; one object for each step in the workflow. [Table 438 on page 773](#) lists the fields in the step-definition JSON object.

##### **variables**

Returns an array of variable-definition objects, with one object for each variable that is referenced in the workflow. [Table 442 on page 779](#) lists the fields in the variable-definition JSON object.

To specify both attributes, separate the attributes by a comma (','), as follows:

```
returnData=steps,variables
```

Do not enclose the attributes in quotation marks.

## Description

This operation retrieves the content of a z/OSMF workflow definition. You can optionally expand the returned information through the specification of query parameters.

A workflow definition might consist of multiple XML files, including a primary file and possibly other files that are included by the primary file. The workflow definition resides in a z/OS UNIX file system or a z/OS data set.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 437 on page 771](#). If you include the optional query parameter `returnData` on the request, the operation can return more information about the workflow definition steps or variables, or both. For the format of this information, see the JSON objects that are described in [Table 438 on page 773](#) and [Table 442 on page 779](#).

## Authorization requirements

See [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 437 on page 771](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the indicated reason code and associated error message.

Table 436. HTTP error response codes for a retrieve a workflow definition request	
HTTP error status code	Description
<b>HTTP 403 Forbidden</b>	The requester user ID is not permitted to the workflow definition file.
<b>HTTP 404 Not found</b>	The specified workflow definition file was not found. This resource is specified on the query parameter <code>definitionFilePath</code> .

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 437 on page 771](#) lists the fields in the object.

Table 437. JSON object that is returned to a retrieve a workflow definition request		
Field name	Type	Description
<b>workflowDefaultName</b>	String	Identifies the default name for the workflow. This value is shown in the workflow name field of the Workflows task when the user creates a new workflow instance.
<b>workflowDescription</b>	String	Description of the workflow.
<b>workflowID</b>	String	Workflow ID. A short, arbitrary value that identifies the workflow.
<b>workflowVersion</b>	String	Version of the workflow definition file.
<b>vendor</b>	String	Name of the vendor that provided the workflow definition file.

Table 437. JSON object that is returned to a retrieve a workflow definition request (continued)

Field name	Type	Description
<b>workflowDefinitionFileMD5Value</b>	String	A 128-bit hash value that z/OSMF generates to uniquely identify the workflow definition file.
<b>isCallable</b>	String	Indicates the callable scope for the workflow, as follows:  <b>system</b> An instance of this workflow can be called only from a workflow in the same system.  <b>sysplex</b> An instance of this workflow can be called from a workflow anywhere in the sysplex.  This property is null when the workflow cannot be called by another workflow.
<b>containsParallelSteps</b>	Boolean	For a workflow with automated steps, this property indicates whether the automated steps can be run in parallel (concurrently), thus possibly completing more quickly. For a parallel-steps workflow, this property is <code>true</code> . Otherwise, if this property is <code>false</code> , automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.
<b>scope</b>	String	Indicates the singleton scope for the workflow, as follows:  <b>system</b> A maximum of one instance of this workflow can exist on any one system in the sysplex.  <b>sysplex</b> A maximum of one instance of this workflow can exist in the sysplex.  <b>none</b> An existing instance cannot be used. A new instance of this workflow is always created.
<b>jobsOutputDirectory</b>	String	Name of the UNIX directory that is used for automatically saving job spool files from the workflow.
<b>category</b>	String	Category of the workflow, which is general, configuration, or provisioning.
<b>productID</b>	String	Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).
<b>productName</b>	String	Name of the product or component that is being configured through the workflow.
<b>productVersion</b>	String	Version and release of the product or component that is configured through the workflow.
<b>globalVariableGroup</b>	String	Global variable group for the workflow.
<b>isInstanceVariableWithoutPrefix</b>	String	Indicates whether the simplified format is used for references to instance variables. If <code>true</code> , variable references are simplified; they omit the prefix <code>instance-</code> . If <code>false</code> , variable references must include the prefix <code>instance-</code> .

Table 437. JSON object that is returned to a retrieve a workflow definition request (continued)		
Field name	Type	Description
<b>steps</b>	Array of objects	Array of one or more step-definition objects that contain details about each of the steps in the workflow definition file. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code> . See the “Format of the step-definition object” on page 773 section for a list of the fields in the step-definition object.
<b>variables</b>	Array of objects	Array of one or more variable-definition objects that contain details about the variables that are defined or referenced in the workflow definition file. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>variables</code> . Table 442 on page 779 lists the fields in the variable-definition object.

## Format of the step-definition object

The following tables list the fields in the step-definition object:

- Table 438 on page 773
- Table 439 on page 773
- Table 440 on page 774

Table 438. Retrieve a workflow definition request: Fields included in every step-definition object		
Field name	Type	Description
<b>name</b>	String	Name of the step.
<b>title</b>	String	Step title.
<b>description</b>	String	Step description.
<b>prereqStep</b>	Array of strings	Lists the names of the steps that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.
<b>optional</b>	Boolean	Indicates whether the step is optional (true or false).
<b>steps</b>	Array of objects	For a parent step, this property is an array of one or more step-definition objects that contain details about each of the steps in the workflow.  For a leaf step, this property is null.

Table 439. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a calling step (a step that calls another workflow)		
Field name	Type	Description
<b>calledWorkflowDescription</b>	String	For a step that calls another workflow for execution, this property contains the description of the called workflow. This information might include details such as the name and location of the workflow definition file that is used to create the called workflow.

Table 439. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a calling step (a step that calls another workflow) (continued)

Field name	Type	Description
<b>calledWorkflowID</b>	String	For a step that calls another workflow for execution, this property contains the workflow ID for the called workflow.
<b>calledWorkflowMD5</b>	String	For a step that calls another workflow for execution, this property contains the 128-bit hash value that can be used to identify the called workflow.
<b>calledWorkflowDefinitionFile</b>	String	For a step that calls another workflow for execution, this property contains the path name of the workflow definition file for the called workflow.
<b>calledWorkflowVersion</b>	String	For a step that calls another workflow for execution, this property contains the version of the workflow definition file for the called workflow.
<b>callingStepAutoEnable</b>	Boolean	For a step that calls another workflow for execution, this property indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required (true or false).
<b>callingStepWeight</b>	Integer	For a step that calls another workflow for execution, this property indicates the relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).
<b>callingStepSkills</b>	String	For a step that calls another workflow for execution, this property indicates the type of skills that are required to perform the step.

Table 440. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step

Field name	Type	When returned	Description
<b>actualStatusCode</b>	String	REST steps only	The actual HTTP status code that is received from the REST API request. To obtain this value, map it to a workflow variable.
<b>approvers</b>	Array of objects	All step types	An array of objects that contain: <b>approver</b> One or more user IDs that are separated by spaces and can provide approval. <b>approverSub</b> This can be set to true or false regardless of whether the approver field uses substitution.
<b>autoEnable</b>	Boolean	All step types	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required (true or false).

Table 440. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
<b>expectedStatusCode</b>	String	REST steps only	The expected HTTP status code from the REST API request. If the expectedStatusCode value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a template step returns a return code that is greater than the allowed maximum return code.
<b>failedPattern</b>	Array of strings	Template steps only	Optional regular expression that can be returned for program execution failures. This property might be null.
<b>hostname</b>	String	REST steps only	Indicates the hostname or IP address of the site to which the REST request is directed. For example: <code>www.ibm.com</code> .
<b>httpMethod</b>	String	REST steps only	Indicates the HTTP method that is used for issuing the REST API request. The possible values are: <ul style="list-style-type: none"> <li>• GET</li> <li>• PUT</li> <li>• POST</li> <li>• DELETE</li> </ul>
<b>instructions</b>	String	All step types	Detailed instructions on what the user must do to perform the step.

Table 440. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
<b>isRestStep</b>	Boolean	All step types	<p>Indicates whether this step is a REST API step (true or false).</p> <p>When set to true, the following properties contain details about the REST request. Otherwise, these properties are set to null.</p> <ul style="list-style-type: none"> <li>• actualStatusCode</li> <li>• expectedStatusCode</li> <li>• hostname</li> <li>• httpMethod</li> <li>• port</li> <li>• propertyMappings</li> <li>• queryParameters</li> <li>• requestBody</li> <li>• schemeName</li> <li>• uriPath</li> </ul> <p>The following step properties are not applicable for a REST step, and are therefore omitted from the output:</p> <ul style="list-style-type: none"> <li>• template</li> <li>• output</li> <li>• saveAsDataset</li> <li>• saveAsUnixFile</li> <li>• submitAs</li> <li>• maxLrecl</li> </ul>
<b>maxLrecl</b>	Integer	Template steps only	For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job. This value is an integer 80 - 1024. The default is 1024.
<b>output</b>	String	Template steps only	Indicates the default name of the output file that is produced by the step (a data set or UNIX file). The output file can contain variables and values that are used by subsequent steps.
<b>outputVariablesPrefix</b>	String	Template steps only	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.
<b>port</b>	String	REST steps only	Port number that is associated with the REST request.

Table 440. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
<b>procName</b>	String	Template steps only	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log in to the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.
<b>propertyMappings</b>	Array of objects	REST steps only	<p>An array of property mappings, the format of which is:</p> <pre>{   "mapFrom": "property",   "mapTo": "variable" },</pre> <p>In the mappings:</p> <p><b>mapFrom</b> Is the property from the REST request. The value of this property is assigned to the specified "mapTo" workflow variable.</p> <p><b>mapTo</b> Is the workflow variable that is assigned the value from "mapFrom" property.</p>
<b>queryParameters</b>	String	REST steps only	For a REST request that includes query parameters, this property contains the query parameters. Otherwise, this property is null.
<b>regionSize</b>	String	Template steps only	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value was specified for the step, the default is 50000.
<b>requestBody</b>	String	REST steps only	For a REST request that includes a request body, this property contains the request body. Otherwise, this property is null.
<b>saveAsDataset</b>	String	Template steps only	Data set name (fully qualified, no quotation marks) that contains the saved JCL.
<b>saveAsUnixFile</b>	String	Template steps only	UNIX file name (absolute name) that contains the saved JCL.
<b>schemeName</b>	String	REST steps only	The scheme name that is used for the REST request. For example: http.
<b>scriptParameters</b>	Array of strings	Template steps only	For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.
<b>skills</b>	String	All step types	The type of skills that are required to perform the step.

Table 440. Retrieve a workflow definition request: Additional fields included in the step-definition object only for a normal (non-calling) step (continued)

Field name	Type	When returned	Description
<b>submitAs</b>	String	Template steps only	Indicates the type of executable program: JCL job, a REXX exec, or a UNIX shell script, which includes a REXX exec that is written for the UNIX shell environment. The following values are valid: <ul style="list-style-type: none"> <li>• "JCL "</li> <li>• "TSO-REXX"</li> <li>• "shell-JCL "</li> <li>• "TSO-REXX-JCL "</li> <li>• "TSO-UNIX-REXX"</li> <li>• "TSO-UNIX-shell "</li> </ul>
<b>successPattern</b>	String	Template steps only	Regular expression that is returned for a successful program execution.
<b>template</b>	String	Template steps only	Indicates the template that is used for a JCL job, a REXX exec program, or a UNIX shell script.
<b>timeout</b>	String	Template steps only	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it times out.
<b>uriPath</b>	String	REST steps only	The URI path to use for the REST request.
<b>variable-specifications</b>	Array of objects	All step types	An array of variable-specification-info objects, the format of which is described in <a href="#">Table 441 on page 778</a> .
<b>weight</b>	Integer	All step types	The relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).

### Format of the variable-specification-info object

[Table 441 on page 778](#) lists the fields in the variable-specification-info object.

Table 441. Retrieve a workflow definition request: Format of the variable-specification-info object

Field name	Type	Description
<b>name</b>	String	Name of the variable.
<b>scope</b>	String	Variable scope, which is either instance or global.
<b>required</b>	Boolean	Indicates whether the variable is required (true or false).

### Format of the variable-definition object

[Table 442 on page 779](#) lists the fields in the variable-definition object.

Table 442. Retrieve a workflow definition request: Format of the variable-definition object

Field name	Type	Description
<b>name</b>	String	Name of the variable.
<b>scope</b>	String	Variable scope, which is either instance or global.
<b>abstract</b>	String	A brief description of the variable.
<b>category</b>	String	Name of the logical grouping to which the variable belongs.
<b>choice</b>	Array of strings	The choice value for the variable.
<b>decimalPlaces</b>	Integer	Maximum number of decimal places that can be specified.
<b>default</b>	String	Default value of the variable.
<b>description</b>	String	Description of the variable.
<b>exposeToUser</b>	Boolean	Indicates whether the variable is displayed to the user in the Workflows task.
<b>maxLength</b>	Integer	Maximum length of the variable value.
<b>maxValue</b>	String	Maximum value of the variable.
<b>minLength</b>	Integer	Minimum length of the variable value.
<b>minValue</b>	String	Minimum value of the variable.
<b>promptAtCreate</b>	Boolean	Indicates whether the user is prompted to specify a value for the variable during the create workflow process.
<b>regularExpression</b>	String	Provides a standard regular expression that constrains the variable value, as an alternative to the available validation types.
<b>requiredAtCreate</b>	Boolean	Indicates whether a value must be specified for the variable during the create workflow process.
<b>type</b>	String	Type of variable.
<b>validationType</b>	String	Specifies the validation type for the variable.
<b>valueMustBeChoice</b>	Boolean	Indicates whether the variable value must come from the provided choices. If <code>true</code> , the user must choose from the predefined values. If <code>false</code> , the user can enter a custom value.
<b>visibility</b>	String	Indicates whether the variable is displayed to the Workflows task user (either public or private).

### Example HTTP interaction

In the following example, the GET method is used to retrieve a workflow definition. The location of the workflow definition is specified on the query parameter `definitionFilePath`. The query parameter `returnData=steps,variables` is included to request more information about the workflow steps and variables.

```

GET
/zosmf/workflow/rest/1.0/workflowDefinition
?definitionFilePath=/usr/lpp/zosmf/samples/workflow_sample_program_execution.xml
&returnData=steps,variables HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk

```

Figure 390. Sample request to get a workflow definition

An example of the response is shown in the figures that follow.

```

HTTP/1.1 200 OK
content-length: 5822
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Thur, 12 Mar 2020 18:30:34 GMT
content-type: application/json; charset=UTF-8
{
  "workflowDefaultName": null,
  "isInstanceVariableWithoutPrefix": false,
  "variables": [
    {
      "visibility": "private",
      "exposeToUser": false,
      "promptAtCreate": false,
      "description": "This value is used to specify a procedure name (proc name) for the TSO/E address space \n\t
        that is used to run the program.\n\t",
      "abstract": "Enter a procedure name for running the program.",
      "type": "string",
      "requiredAtCreate": false,
      "default": null,
      "decimalPlaces": null,
      "valueMustBeChoice": false,
      "scope": "instance",
      "name": "procNameVariable",
      "category": "TSO procName",
      "choice": null
    },
    {
      "visibility": "private",
      "exposeToUser": false,
      "regularExpression": "^[A-Z$#@]{1}[0-9A-Z$#@]{0,7}$",
      "promptAtCreate": false,
      "validationType": "GROUP",
      "description": "The group name under whose authority the started task will run.\n\t",
      "abstract": "Group name for the started task.",
      "type": "string",
      "requiredAtCreate": false,
      "default": "SYS1",
      "decimalPlaces": null,
      "valueMustBeChoice": false,
      "scope": "instance",
      "name": "st_group",
      "category": "Started",
      "choice": null
    },
    {
      "visibility": "private",
      "exposeToUser": false,
      "regularExpression": "[0-9A-Z$#@]{1,8}$",
      "promptAtCreate": false,
      "validationType": "USERID",
      "description": "The user ID under whose authority the new started task will run.\n\t",
      "abstract": "User ID for the started task.",
      "type": "string",
      "requiredAtCreate": false,
      "default": "MYSTUSER",
      "decimalPlaces": null,
      "valueMustBeChoice": false,
      "scope": "instance",
      "name": "st_user",
      "category": "Started",
      "choice": null
    }
  ],
}

```

Figure 391. Sample response from a get workflow definition request (Part 1 of 3)

```

"productID": "ABC123",
"workflowDescription": "Sample that demonstrates how to run an executable program from a step.\n\t",
"steps": [
{
  "template": "\n#!/bin/sh\nnecho \"this is a sample to submit shell script to run immediately\"\n\nnecho \"the first parameter is :\" $1 \t\n\nnecho ${instance-st_user}\necho prefix:st_group = SYS123\nnecho prefix:st_user = USERS\nnecho \"This symbol is used to indicate success\"\n\nnecho \"The program ran successfully !!\"\n\n",
  "instructions": "This step outputs some variables and prints a few words.\n",
  "autoEnable": "false",
  "maxLrecl": 1024,
  "submitAs": "TSO-UNIX-shell",
  "failedPattern": [
    "failed.*"
  ],
  "description": "In this step, you submit an inline UNIX shell script for immediate processing \n\t\nton the host system. In this example, the step is expected to complete successfully.\n\t\t",
  "weight": 1,
  "outputVariablesPrefix": "prefix:",
  "optional": false,
  "procName": "${instance-procNameVariable}",
  "title": "A step that runs a UNIX shell script.",
  "timeout": 60,
  "regionSize": 50000,
  "skills": "System Programmer",
  "output": null,
  "scriptParameters": "para1",
  "isRestStep": false,
  "saveAsUnixFile": "/u/${instance-st_user}/savedStuff/myScript.sh",
  "outputSysoutDD": null,
  "variable-specifications": [
    {
      "scope": "instance",
      "name": "st_group",
      "required": true
    },
    {
      "scope": "instance",
      "name": "st_user",
      "required": true
    },
    {
      "scope": "instance",
      "name": "procNameVariable",
      "required": true
    }
  ],
  "name": "TSO-UNIX-shell_Execution",
  "successPattern": "success.*",
  "saveAsDataset": null
},
{
  "template": "/* rexx */\nparse arg arg1\n\nSAY \"this is a sample to submit UNIX REXX script to run immediately\"\n\nSAY \"the first parameter is :\" arg1\n\nSAY ${instance-st_user}\n\nSAY \"prefix:st_group =\" SYS123\n\nSAY \"prefix:st_user =\" USERS\n\nSAY \"This symbol is used to indicate failed\"\n\n",
  "instructions": "This step outputs some variables and prints a few words.\n",
  "autoEnable": "false",
  "maxLrecl": 1024,
  "submitAs": "TSO-UNIX-REXX",
  "failedPattern": [
    "failed.*"
  ],
  "description": "In this step, you submit an inline UNIX REXX exec for immediate processing \n\t\nton the host system. In this example, the step is expected to fail.\n\t\t",
  "weight": 1,
  "outputVariablesPrefix": "prefix:",
  "optional": false,
  "procName": "${instance-procNameVariable}",
  "title": "A step that runs a UNIX REXX exec program.",
  "timeout": 60,
  "regionSize": 50000,
  "skills": "System Programmer",

```

Figure 392. Sample response from a get workflow definition request (Part 2 of 3)

```

        "output": null,
        "scriptParameters": "para1",
        "isRestStep": false,
        "saveAsUnixFile": "/u/${instance-st_user}/savedStuff/myScript.sh",
        "outputSysoutDD": null,
        "variable-specifications": [
            {
                "scope": "instance",
                "name": "st_group",
                "required": true
            },
            {
                "scope": "instance",
                "name": "st_user",
                "required": true
            },
            {
                "scope": "instance",
                "name": "procNameVariable",
                "required": true
            }
        ],
        "name": "TSO-UNIX-REXX_Execution",
        "successPattern": "success.*",
        "saveAsDataset": null
    },
    {
        "template": "/* rexx */\nparse arg arg1\nSAY \"this is a sample to submit TSO REXX script to run immediately\"\nSAY \"the first parameter is :\" arg1\nSAY ${instance-st_user}\nSAY \"prefix:st_group =\" SYS123\nSAY \"prefix:st_user =\" USERS\nSAY \"This execution will meets timeout.\"\n",
        "instructions": "This step outputs some variables and prints a few words.\n",
        "autoEnable": "false",
        "maxLrecl": 1024,
        "submitAs": "TSO-REXX",
        "failedPattern": [
            "failed.*"
        ],
        "description": "In this step, you submit an inline REXX exec for immediate processing \n\t\tton the host system.\n\t\tIn this example, the processing is ended by a time-out condition.\n\t\t",
        "weight": 1,
        "outputVariablesPrefix": "prefix:",
        "optional": false,
        "procName": "${instance-procNameVariable}",
        "title": "A step that runs a REXX exec program.",
        "timeout": 60,
        "regionSize": 50000,
        "skills": "System Programmer",
        "output": null,
        "scriptParameters": "para1",
        "isRestStep": false,
        "saveAsUnixFile": "/u/${instance-st_user}/savedStuff/myScript.sh",
        "outputSysoutDD": null,
        "variable-specifications": [
            {
                "scope": "instance",
                "name": "st_group",
                "required": true
            },
            {
                "scope": "instance",
                "name": "st_user",
                "required": true
            },
            {
                "scope": "instance",
                "name": "procNameVariable",
                "required": true
            }
        ],
        "name": "TSO-TSO-REXX_Execution",
        "successPattern": "success.*",
        "saveAsDataset": null
    }
],
    "productName": "Product ABC",
    "globalVariableGroup": null,
    "containsParallelSteps": false,
    "workflowDefinitionFileMD5Value": "5c5dd66eb3ca3cd1c578ccf323d57cc0",
    "isCallable": null,
    "productVersion": "Version 1",
    "jobsOutputDirectory": null,
    "vendor": "IBM",
    "scope": "none",
    "workflowVersion": "1.0",
    "category": "configuration",
    "workflowID": "programExecutionSample"
}

```

Figure 393. Sample response from a get workflow definition request (Part 3 of 3)

## Archive a workflow instance

You can use this operation to archive a z/OSMF workflow instance on a z/OS system.

### HTTP method and URI path

---

```
POST /zosmf/workflow/rest/<version>/workflows/<workflowKey>/operations/archive
```

---

In this request, the URI path variables are described, as follows:

- *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- *<workflowKey>* identifies the workflow to be archived.

### Query parameters

None.

### Description

This operation archives a workflow instance, which is identified by the workflow key that is specified in the request URI.

You can archive any workflows that are completed or that you no longer need. Doing so removes the workflow from the Workflows table in the Workflows task and places it in an archive for your reference. An archived workflow is no longer active, but its information can be viewed by you at any time. You cannot undo this action.

After you archive a workflow, you can list it or delete it, or retrieve its properties.

When you no longer want to retain an archived workflow, you can delete it permanently from z/OSMF.

To be archived, a workflow must be in one of the following states:

- In-progress
- Complete
- Canceled.

It is not possible to archive a workflow while it is involved in a workflow activity. Specifically, you cannot archive a workflow when it:

- Is locked for an update operation
- Contains an automated step that is running
- Is called for processing by another workflow.

To do so, you must allow the processing to complete first.

On successful completion, HTTP status code 201 (Created) is returned, indicating that the request resulted in the archival of the workflow. The URI path for the workflow is provided in the Location response header and a response body is provided, as described in the [“Response content” on page 784](#).

### Request content

None.

### Authorization requirements

See [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 201 (Created) is returned and the response body is provided, as described in [“Response content”](#) on page 784.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body that provides the reason code and the associated error message.

Table 443. HTTP error response codes for an archive workflow request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained an incorrect parameter, such as an incorrect workflow key.
<b>HTTP 403 Forbidden</b>	The requestor user ID is not permitted to archive the workflow.
<b>HTTP 404 Not found</b>	The specified workflow key was not found; the workflow does not exist.
<b>HTTP 409 Request conflict</b>	Request cannot be performed because the specified workflow has a status that makes the workflow ineligible to be archived.

Additional standard status codes can be returned, as described in [“HTTP status codes”](#) on page 727.

## Response content

On successful completion, the service returns the response body, which contains a JSON object with the workflow key. [Table 444 on page 784](#) describes the contents of the response body.

Table 444. Response from an archive workflow request		
Field name	Type	Description
<b>workflowKey</b>	String	Workflow key. A string value, generated by z/OSMF, which is used to uniquely identify the archived workflow instance.

## Example HTTP interaction

In [Figure 394 on page 784](#), a request is submitted to archive the workflow that is identified by the workflow key 2535b19e-a8c3-4a52-9d77-e30bb920f912.

```
POST /zosmf/workflow/rest/1.0/workflows/2535b19e-a8c3-4a52-9d77-e30bb920f912/operations/archive
Host: zosmf1.yourco.com
Connection: close
```

*Figure 394. Sample request to archive a workflow*

A sample response is shown in [Figure 395 on page 784](#).

```
HTTP/1.1 201 Created
{
  "workflowKey": "2535b19e-a8c3-4a52-9d77-e30bb920f912"
}
```

*Figure 395. Sample response from an archive workflow request*

## List the archived workflows for a system

You can use this operation to list the archived z/OSMF workflows for a system or sysplex.

### HTTP method and URI path

---

```
GET /zosmf/workflow/rest/<version>/archivedworkflows
```

---

In this request, the URI path variable *<version>* identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.

### Query parameters

Optionally, your request can include one or more of the following query parameters to filter the results:

#### Orderby

To sort the returned instances by time, specify either of the following values:

**"desc":**

From the newest to the oldest

**"asc":**

From the oldest to the newest

#### View

An string type to select the list instances by view:

**"user":**

Return the archived workflow instances that are owned by the user, up to a maximum of 200 workflow instances. This value is the default.

**"domain":**

For archived provisioning workflows, return the instances that the user is authorized to view. The user must be a domain owner. The results are grouped by domain, with up to 200 instances per domain.

Observe the following conventions:

- Query parameters are optional; you can specify one or more query parameters, as needed.
- You use a question mark ('?') to separate the first query parameter from the resource.
- To specify multiple query parameters in combination, use an ampersand (&).

### Description

This operation retrieves a list of archived workflows that you are authorized to view.

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 446 on page 786](#).

### Authorization requirements

See [“Authorization requirements” on page 726](#).

### HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 445. HTTP error response codes for a get archived workflow properties request	
HTTP error status code	Description
<b>HTTP 400 Bad request</b>	The request contained an error, such as an incorrect query parameter.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the response body contains one property, which is called archived workflows. This property is an array of workflow-info objects. Table 446 on page 786 lists the fields in the workflow-info object. If no workflows match the filter criteria, HTTP status code 200 (OK) is returned with an empty array.

Table 446. List archived workflows request: Format of the workflow-info object		
Field name	Type	Description
<b>workflowName</b>	String	Descriptive name for the workflow.
<b>workflowKey</b>	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
<b>archivedInstanceURI</b>	String	Workflow instance URI path, which you can use to retrieve information about the archived workflow.

## Example HTTP interaction

In the following example, the GET method is used to list the archived workflows on a system. Here, the query parameter `?orderBy=desc` is included to order the results in descending order.

```
GET /zosmf/workflow/rest/1.0/archivedworkflows?orderBy=desc HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

*Figure 396. Sample request to list archived workflows*

For a successful request, the HTTP response includes a JSON document that contains the requested information. In the following example, three archived workflows were found for the requestor user ID.

```

HTTP/1.1 200 OK
content-length: 464
content-language: en-US
x-powered-by: Servlet/3.0
server: WebSphere Application Server
connection: Close
date: Wed, 11 Feb 2015 18:30:34 GMT
content-type: application/json; charset=UTF-8

HTTP/1.1 200 OK
{
  "archivedWorkflows": [
    {
      "workflowName": "Sample demonstrating variable substitution and the use of a wizard. - Workflow_5",
      "workflowKey": "2535b19e-a8c3-4a52-9d77-e30bb920f912",
      "archivedInstanceURI": "\\zosmf\\workflow\\rest\\1.0\\archivedworkflows\\
        /2535b19e-a8c3-4a52-9d77-e30bb920f912"
    },
    {
      "workflowName": "Sample demonstrating variable substitution and the use of a wizard. - Workflow_0",
      "workflowKey": "8f0f572a-0eb5-493e-91b2-3d549374e07d",
      "archivedInstanceURI": "\\zosmf\\workflow\\rest\\1.0\\archivedworkflows\\
        /8f0f572a-0eb5-493e-91b2-3d549374e07d"
    },
    {
      "workflowName": "Sample demonstrating variable substitution and the use of a wizard. - Workflow_5",
      "workflowKey": "1aead54d-3507-4473-9cda-e4fd25eb21b8",
      "archivedInstanceURI": "\\zosmf\\workflow\\rest\\1.0\\archivedworkflows\\
        /1aead54d-3507-4473-9cda-e4fd25eb21b8"
    }
  ]
}

```

Figure 397. Sample response from a list archived workflows request

## Get the properties of an archived workflow

You can use this operation to retrieve the properties of an archived z/OSMF workflow.

### HTTP method and URI path

```
GET /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the archived workflow to be queried.

### Query parameters

You can specify the following query parameter on this request:

#### returnData

This optional query parameter is used to request information about the workflow steps and variables. Include one or both of the following attributes on the `returnData` parameter:

##### steps

Returns an array of step-info objects; one object for each step in the workflow. [Table 450 on page 793](#) lists the fields in the step-info JSON object.

##### variables

Returns an array of variable-info objects; one object for each variable that is referenced in the workflow. [Table 451 on page 800](#) lists the fields in the variable-info JSON object.

To specify both attributes, separate the attributes by a comma (','), as follows:

```
returnData=steps,variables
```

Do not enclose the attributes in quotation marks.

## Description

This operation retrieves the properties of an archived z/OSMF workflow. You can optionally expand the returned information through the specification of query parameters. On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 448 on page 788](#).

For the format of this information, see the JSON objects that are described in [Table 450 on page 793](#) and [Table 451 on page 800](#).

## Authorization requirements

See [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 200 (OK) is returned and the response body is provided, as described in [Table 448 on page 788](#).

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 447. HTTP error response codes for a get archived workflow properties request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained an incorrect parameter, such as an incorrect workflow key.
HTTP 403 Forbidden	The requestor user ID is not permitted to retrieve the workflow properties.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.

Additional standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

On successful completion, the response body is a JSON object that contains the retrieved data. [Table 448 on page 788](#) lists the fields in the JSON object.

Table 448. JSON object that is returned to a get archived workflow properties request		
Field name	Type	Description
workflowName	String	Descriptive name for the workflow.
workflowKey	String	Workflow key. A string value, generated by z/OSMF to uniquely identify the workflow instance.
workflowDescription	String	Description of the workflow.
workflowID	String	Workflow ID. A short, arbitrary value that identifies the workflow.
workflowVersion	String	Version of the workflow definition file.

Table 448. JSON object that is returned to a get archived workflow properties request (continued)		
Field name	Type	Description
<b>workflowDefinitionFileMD5Value</b>	String	The 128-bit hash value that is associated with the workflow definition file that was used to create the workflow.
<b>vendor</b>	String	Name of the vendor that provided the workflow definition file.
<b>owner</b>	String	User ID of the workflow owner.
<b>system</b>	String	Full name of the z/OS system on which the workflow is to be performed. This value is in the format <i>sysplex.sysname</i> .
<b>category</b>	String	Category of the workflow, which is either general or configuration.
<b>productID</b>	String	Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).
<b>productName</b>	String	Name of the product or component that is being configured through the workflow.
<b>productVersion</b>	String	Version and release of the product or component that is configured through the workflow.
<b>percentComplete</b>	Integer	Percentage of the workflow that is completed. z/OSMF calculates this value based on the number of steps in the workflow and the relative weighting value of each step.
<b>isCallable</b>	Boolean	Indicates whether a workflow is eligible to be called by another workflow. For more information, see <a href="#">“Callable workflows”</a> on page 815.
<b>containsParallelSteps</b>	Boolean	For a parallel-steps workflow, this property is <code>true</code> . If so, the automation ready steps can be run in parallel (concurrently), thus possibly completing more quickly.  Otherwise, if this property is <code>false</code> , automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.
<b>scope</b>	String	Restricts a workflow to one instance only. The scope attribute can be set to <code>system</code> , <code>sysplex</code> , or <code>none</code> . For more information, see <a href="#">“Setting the workflow scope”</a> on page 815.
<b>statusName</b>	String	Indicates the current workflow status, which is <i>archived</i> .
<b>deleteCompletedJobs</b>	Boolean	For a workflow that submits a job, this property specifies whether the job is deleted from the JES spool after it completes successfully, as follows: <ul style="list-style-type: none"> <li><code>false</code> means that the job is retained on the JES spool until it is removed by a user or automated process.</li> <li><code>true</code> means that the job is deleted from the JES spool after it completes or fails.</li> </ul>

Table 448. JSON object that is returned to a get archived workflow properties request (continued)

Field name	Type	Description
<b>automationStatus</b>	Object	<p>An automation-info object that contains details about the most recent start automation request for the workflow. The content of this property depends on the following factors:</p> <ul style="list-style-type: none"> <li>• If no automation was performed for the workflow, this property is null.</li> <li>• If automation processing is still in progress, this property indicates the step that is being processed.</li> <li>• If automation was restarted after it was stopped, this property indicates the status of the current start automation request.</li> <li>• If automation is stopped and the workflow status is complete, this property indicates that automation is completed.</li> <li>• If automation is stopped and the workflow status is not complete, this property identifies the step that is most closely related to the reason why automation was stopped.</li> </ul> <p><b>Notes about parallel-step workflows:</b></p> <ul style="list-style-type: none"> <li>– When a parallel-steps workflow is started, all of its automation ready steps are processed until they complete or fail, or automation is stopped. Failure of a step does not stop automation processing for other automation ready steps in the workflow.</li> <li>– In a parallel-steps workflow: <ul style="list-style-type: none"> <li>- The automation ready steps are processed in an unpredictable order, not sequentially as is done for other types of workflows.</li> <li>- If automation is currently stopped and the workflow is not yet complete, this property identifies the first uncompleted step that was returned to the Get Properties request.</li> </ul> </li> </ul> <p>Table 449 on page 792 lists the fields in the automation-info object.</p>
<b>jobsOutputDirectory</b>	String	Name of the UNIX directory that is used for automatically saving job spool files from the workflow.
<b>autoDeleteOnCompletion</b>	Boolean	<p>Specifies whether the workflow is automatically deleted from the system after it completes successfully, as follows:</p> <ul style="list-style-type: none"> <li>• <code>false</code> means that the workflow is retained after it is complete, until it is removed by a user. A complete workflow is one in which all of its steps are marked complete or skipped.</li> <li>• <code>true</code> means that the workflow is automatically deleted from the system after it completes. As a result, the workflow is removed from the Workflows table in the Workflows task user interface.</li> </ul>

Table 448. JSON object that is returned to a get archived workflow properties request (continued)

Field name	Type	Description
<b>access</b>	String	Specifies the access type for the workflow. The access type determines which users can view the workflow steps and edit the step notes, as described in <a href="#">“Workflow access type”</a> on page 728.  The following values are valid: <ul style="list-style-type: none"> <li>• Public</li> <li>• Restricted</li> <li>• Private</li> </ul>
<b>archivedTime</b>	String	Date and time on the system when the workflow was archived. The date and time is in Greenwich Mean Time (GMT).
<b>accountInfo</b>	String	For a workflow that submits a job, this property specifies the account information to use in the JCL JOB statement. This property can be null.
<b>jobStatement</b>	String	For a workflow that submits a job, this property specifies the JOB statement JCL that is used in the job. This property can be null, or a list of JCL cards, each up to 72 characters long. Columns 1 and 2 of each record must be "/" or "/"* and the job name must be 1 to 8 characters.
<b>steps</b>	Array of objects	Array of one or more step-info objects that contain details about each of the steps in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>steps</code> .  The content of this array depends on what the requestor is permitted to see. For more information, see <a href="#">“Description”</a> on page 788.  <a href="#">Table 450 on page 793</a> lists the fields in the step-info object.
<b>variables</b>	Array of objects	Array of one or more variable-info objects that contain details about the variables that are used in the workflow. This property is returned only when the query parameter <code>returnData</code> specifies the attribute <code>variables</code> .  The content of this array depends on what the requestor is permitted to see. For more information, see <a href="#">“Description”</a> on page 788.  <a href="#">Table 452 on page 800</a> lists the fields in the variable-info object.

## Format of the automation-info object

[Table 449 on page 792](#) lists the fields in the automation-info JSON object.

Table 449. Get Archived Workflow Properties request: Format of the automation-info object		
Field name	Type	Description
<b>startUser</b>	String	User ID of the user who initiated the automation processing.
<b>startedTime</b>	Timestamp	Time that automation processing started. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time expressed as the number of milliseconds since midnight on January 1, 1970 UTC.
<b>stoppedTime</b>	Timestamp	Time that automation processing stopped. If automation is still in progress, this property is set to null. The timestamp data type is used to mean a non-negative Long integer quantity where the value represents a date and time expressed as the number of milliseconds since midnight on January 1, 1970 UTC.
<b>currentStepName</b>	String	Depending on the current phase of automation processing, this property contains one of the following values: <ul style="list-style-type: none"> <li>Name of the step that is being processed automatically.</li> <li>Name of the step that caused automation to stop.</li> <li>For a workflow that uses parallel processing (a <i>parallel-steps workflow</i>) this value is the name of the first step that is incomplete.</li> </ul> If automation is stopped and the workflow status is complete, this property is set to null.
<b>currentStepNumber</b>	String	The step number. If automation is stopped and the workflow status is complete, this property is set to null.
<b>currentStepTitle</b>	String	Step title. If automation is stopped and the workflow status is complete, this property is set to null.
<b>messageID</b>	String	Message identifier for the accompanying message. If automation is still in progress, this property is set to null.
<b>messageText</b>	String	Message text that describes the reason that automation is stopped. If automation is still in progress, this property is set to null.

## Format of the step-info object

Table 450 on page 793 lists the fields in the step-info JSON object. Not all of the properties are returned for every step. Some properties are returned or omitted depending on the step type, as noted in the **When returned** column. This information in this column indicates whether valid data is returned for the step, as follows:

### All step types

Properties that are returned for all step types.

### Calling steps

Properties that are returned for a step that calls another workflow for execution.

### Template steps

Properties that are returned for a step that runs a program, such as a batch job, REXX exec, or UNIX shell script.

### REST steps

Properties that are returned for a step that issues a REST API request, such as a GET or PUT request.

With regard to returned data, a template step and a REST step are mutually exclusive. The returned information for a template step does not include the properties for a REST step. Likewise, the returned information for a REST step does not include the properties for a template step. To help you identify which steps are REST steps, the step-info object includes the `isRestStep` property, set to true or false.

Table 450. Get Archived Workflow Properties request: Format of the step-info object			
Field name	Type	When returned	Description
<b>name</b>	String	All step types	Name of the step.
<b>actualStatusCode</b>	String	REST steps only	The actual HTTP status code received from the REST API request. To obtain this value, map it to a workflow variable.
<b>assignees</b>	String	Calling steps and template steps	Step assignees; one or more user IDs that are assigned to the step. Multiple items are separated by commas.
<b>autoEnable</b>	Boolean	All step types	Indicates whether the step can be performed automatically when all prerequisite steps are completed, and no user inputs are required.
<b>calledInstanceURI</b>	String	Calling steps only	For a step that calls another workflow for execution, this property contains the URI path of the called workflow instance. You can use this value to retrieve information about the called workflow.  This property is null until the step is performed and either a new instance of the called workflow is created or an existing instance is found.
<b>calledWorkflowID</b>	String	Calling steps only	This property contains the workflow ID of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property <code>calledWorkflowMD5</code> is specified.
<b>calledWorkflowVersion</b>	String	Calling steps only	This property contains the workflow version of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed.  This property: <ul style="list-style-type: none"> <li>• Is null when the property <code>calledWorkflowMD5</code> is specified</li> <li>• Might be null when the property <code>calledWorkflowID</code> is specified.</li> </ul>
<b>calledWorkflowMD5</b>	String	Calling steps only	This property contains the 128-bit hash value of a workflow definition file; it is used to help locate an existing workflow instance when this step is performed. This property is null when the property <code>calledWorkflowID</code> is specified.
<b>calledWorkflowDescription</b>	String	Calling steps only	This property contains a description of the workflow to be called, from the point of view of the calling workflow.

Table 450. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>calledWorkflowDefinitionFile</b>	String	Calling steps only	This property contains the name of the workflow definition file that will be used to create a new workflow if an existing instance is not found when this step is performed. This property might be null.
<b>description</b>	String	All step types	Step description.
<b>expectedStatusCode</b>	String	REST steps only	The expected HTTP status code from the REST API request. If the expectedStatusCode value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a template step returns a return code that is greater than the allowed maximum return code.
<b>failedPattern</b>	Array of strings	Template steps only	Optional regular expression that can be returned for program execution failures. This property might be null.
<b>hasCalledWorkflow</b>	Boolean	Calling steps and template steps	Indicates whether this step calls another workflow (true or false). If true, this step is a "calling" step, that is, it calls another workflow for execution. If false, it is a template step.  This property is returned only when steps=null, which indicates a leaf step.
<b>hostname</b>	String	REST steps only	Indicates the hostname or IP address of the site to which the REST request is directed. For example: <code>www.ibm.com</code> .
<b>httpMethod</b>	String	REST steps only	Indicates the HTTP method that is used for issuing the REST API request. The possible values are: <ul style="list-style-type: none"> <li>• GET</li> <li>• PUT</li> <li>• POST</li> <li>• DELETE</li> </ul>
<b>instructions</b>	String	Template steps only	Detailed instructions on what the user must do to perform the step.
<b>instructionsSub</b>	Boolean	Template steps only	Indicates whether the step instructions contain variables (true or false).
<b>isConditionStep</b>	Boolean	Calling steps and template steps	Indicates whether this step is a conditional step (true or false).

Table 450. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>isRestStep</b>	Boolean	All step types	<p>Indicates whether this step is a REST API step (true or false).</p> <p>When set to true, the following properties contain details about the REST request. Otherwise, these properties are set to null.</p> <ul style="list-style-type: none"> <li>• actualStatusCode</li> <li>• expectedStatusCode</li> <li>• hostname</li> <li>• hostnameSub</li> <li>• httpMethod</li> <li>• port</li> <li>• portSub</li> <li>• queryParameters</li> <li>• queryParametersSub</li> <li>• requestBody</li> <li>• requestBodySub</li> <li>• schemeName</li> <li>• schemeNameSub</li> <li>• uriPath</li> <li>• uriPathSub</li> </ul> <p>The following step properties are not applicable for a REST step and thus, are omitted from the output:</p> <ul style="list-style-type: none"> <li>• template</li> <li>• templateSub</li> <li>• output</li> <li>• outputSub</li> <li>• saveAsDataset</li> <li>• saveAsDatasetSub</li> <li>• saveAsUnixFile</li> <li>• saveAsUnixFileSub</li> <li>• submitAs</li> <li>• maxLrecl</li> <li>• returnCode</li> </ul>
<b>maxLrecl</b>	Integer	Template steps only	<p>For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job. This value is an integer 80 - 1024. The default is 1024.</p>
<b>optional</b>	Boolean	All step types	<p>Indicates whether the step is optional (true or false).</p>

Table 450. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>output</b>	String	Template steps only	Indicates the name of the output file produced by the step (a data set or UNIX file). The output file can contain variables and values that are used by subsequent steps.
<b>outputSub</b>	Boolean	Template steps only	Indicates whether the output file name contains variable substitution (true or false).
<b>outputVariablesPrefix</b>	String	Template steps only	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.
<b>owner</b>	String	Calling steps and template steps	User ID of the step owner.
<b>port</b>	String	REST steps only	Port number that is associated with the REST request.
<b>portSub</b>	Boolean	REST steps only	Indicates whether the port number contains variable substitution (true or false).
<b>prereqStep</b>	Array of strings	All step types	Lists the names of the steps that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.
<b>procName</b>	String	Template steps only	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.
<b>queryParameters</b>	String	REST steps only	For a REST request that includes query parameters, this property contains the query parameters. Otherwise, this property is null.
<b>queryParametersSub</b>	Boolean	REST steps only	This property indicates whether the query parameters contain variable substitution (true or false). Otherwise, this property is null.
<b>regionSize</b>	String	Template steps only	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value was specified for the step, the default is 50000.
<b>requestBody</b>	String	REST steps only	For a REST request that includes a request body, this property contains the request body. Otherwise, this property is null.

Table 450. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>requestBodySub</b>	Boolean	REST steps only	This property indicates whether the request body variable substitution (true or false). Otherwise, this property is null.
<b>returnCode</b>	String	Template steps only	For a step that submits a job to run, this property indicates the return code that was returned when the job was submitted.
<b>saveAsDataset</b>	String	Template steps only	Data set name (fully qualified, no quotation marks) that contains the saved JCL.
<b>saveAsDatasetSub</b>	Boolean	Template steps only	Indicates whether the data set name contains variable substitution (true or false).
<b>saveAsUnixFile</b>	String	Template steps only	UNIX file name (absolute name) that contains the saved JCL.
<b>saveAsUnixFileSub</b>	Boolean	Template steps only	Indicates whether the UNIX file name contains variable substitution (true or false).
<b>schemeName</b>	String	REST steps only	The scheme name that is used for the REST request. For example: <code>http</code> .
<b>schemeNameSub</b>	Boolean	REST steps only	Indicates whether the scheme name contains variable substitution (true or false).
<b>scriptParameters</b>	Array of strings	Template steps only	For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.
<b>skills</b>	String	Calling steps and template steps	The type of skills that are required to perform the step.

Table 450. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>state</b>	String	All step types	<p>State of the step. One of the following status indicators is displayed:</p> <ul style="list-style-type: none"> <li>• <b>Unassigned.</b> The step is in the <i>Unassigned</i> state; no users or groups are assigned to the step.</li> <li>• <b>Assigned.</b> Users or groups are assigned to the step, but no user accepted ownership of the step.</li> <li>• <b>Not Ready.</b> A user accepted ownership of the step, however, a prerequisite step must be completed or a conditional dependency must be satisfied before the step can be performed.</li> <li>• <b>Ready.</b> The step is ready to be performed; all prerequisite steps and conditional dependencies are satisfied.</li> <li>• <b>In Progress.</b> The step is in progress. For a parent step, a state of <i>In Progress</i> means that at least one of the child steps is started, but is not yet complete, overridden, or skipped. For a leaf step, a state of <i>In Progress</i> means that the step is started, but is not yet complete, overridden, or skipped.</li> <li>• <b>Submitted.</b> The step included a job, which the step owner submitted.</li> <li>• <b>Complete.</b> The step was completed.</li> <li>• <b>Skipped.</b> The step was bypassed by the step assignee.</li> <li>• <b>Complete (Override).</b> The step was marked complete, but the work was performed outside of the Workflows task.</li> <li>• <b>Failed.</b> The step included a job that was submitted by the step owner. However, the job failed to complete successfully.</li> <li>• <b>Conflicts.</b> The step created an output file for use by a subsequent step. However, values in that file conflict with existing instance or global variables.</li> <li>• <b>Condition Not Satisfied.</b> The step is a conditional step, and the condition is not satisfied.</li> </ul>
<b>stepNumber</b>	String	All step types	<p>The step number. Steps are numbered to indicate the sequence in which steps are to be performed. For example, the first step in a workflow is 1.</p>
<b>steps</b>	Array of objects	All step types	<p>For a parent step, this is a nested array of step-info objects. For a leaf step, this property is null.</p> <p>Check this property first before you check the other, non-common step properties. A non-null value here means that the calling step properties are omitted, as are the template step properties and the REST step properties.</p>

Table 450. Get Archived Workflow Properties request: Format of the step-info object (continued)

Field name	Type	When returned	Description
<b>submitAs</b>	String	Template steps only	Indicates the type of executable program: JCL job, a REXX exec, or a UNIX shell script, which includes a REXX exec that is written for the UNIX shell environment. The possible values are the following: <ul style="list-style-type: none"> <li>• "JCL "</li> <li>• "TSO-REXX"</li> <li>• "shell-JCL "</li> <li>• "TSO-REXX-JCL "</li> <li>• "TSO-UNIX-REXX"</li> <li>• "TSO-UNIX-shell "</li> </ul>
<b>successPattern</b>	String	Template steps only	Regular expression that is returned for a successful program execution.
<b>template</b>	String	Template steps only	Indicates the template that is used to run a program or batch job (inline or external file).
<b>templateSub</b>	Boolean	Template steps only	Indicates whether template contains variable substitution (true or false). The default is false.
<b>timeout</b>	String	Template steps only	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it is ended by a timeout condition.
<b>title</b>	String	All step types	Step title.
<b>uriPath</b>	String	REST steps only	The URI path to use for the REST request.
<b>uriPathSub</b>	Boolean	REST steps only	Indicates whether the URI path contains variable substitution (true or false).
<b>userDefined</b>	Boolean	All step types	Indicates whether the step was added manually to the workflow (true or false). If true, the step was added by the workflow owner, using the <b>Update Workflow Steps</b> action in the Workflows table. If false, the step was defined in the workflow definition that was used to create the workflow.
<b>variable-references</b>	Array of objects	Template steps only	An array of variable-reference objects, the format of which is described in <a href="#">Table 451 on page 800</a> .
<b>weight</b>	Integer	Calling steps and template steps	The relative difficulty of the step compared to other steps within this workflow (an integer value 1 - 1000).

## Format of the variable-reference object

Table 451 on page 800 lists the fields in the variable-reference JSON object.

Table 451. Get Archived Workflow Properties request: Format of the variable-reference object		
Field name	Type	Description
<b>name</b>	String	Name of the variable.
<b>scope</b>	String	Variable scope, which is either instance or global.

### Format of the variable-info object

Table 452 on page 800 lists the fields in the variable-info JSON object.

Table 452. Get Archived Workflow Properties request: Format of the variable-info object		
Field name	Type	Description
<b>name</b>	String	Name of the variable.
<b>scope</b>	String	Variable scope, which is either instance or global.
<b>type</b>	String	Type of variable, which is one of the following values: <ul style="list-style-type: none"><li>• boolean</li><li>• string</li><li>• number</li><li>• date</li><li>• time</li><li>• array</li></ul>
<b>value</b>	String	Variable value.
<b>visibility</b>	String	Public or private.

### Example HTTP interaction

In the following example, the GET method is used to retrieve information about an archived workflow. The workflow is uniquely identified by the workflow key, which is represented by the following string value: 2535b19e-a8c3-4a52-9d77-e30bb920f912.

```
GET /zosmf/workflow/rest/1.0/archivedworkflows/2535b19e-a8c3-4a52-9d77-e30bb920f912
HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 398. Sample request to get archived workflow properties

An example of the response is shown in [Figure 399 on page 801](#).

```

HTTP/1.1 200 OK{
  "access": "Public",
  "productID": "ABC123",
  "jobStatement": null,
  "deleteCompletedJobs": false,
  "productName": "Product ABC",
  "globalVariableGroup": null,
  "productVersion": "Version 1",
  "jobsOutputDirectory": null,
  "vendor": "IBM",
  "archivedTime": "2020-03-04 03:18:36",
  "scope": "none",
  "statusName": "archived",
  "workflowID": "programExecutionSample",
  "owner": "zosmfad",
  "accountInfo": null,
  "isInstanceVariableWithoutPrefix": false,
  "workflowName": "testProgramExecutionSample",
  "automationStatus": null,
  "autoDeleteOnCompletion": false,
  "percentComplete": 0,
  "workflowDescription": "Sample that demonstrates how to run an executable program from a step.\n\t",
  "containsParallelSteps": false,
  "workflowDefinitionFileMD5Value": "5c5dd66eb3ca3cd1c578ccf323d57cc0",
  "isCallable": null,
  "system": "PLEX1.SY1",
  "workflowKey": "7a2263a7-7c91-40b4-8892-2a4342a222c3",
  "workflowVersion": "1.0",
  "category": "configuration"
}

```

Figure 399. Sample response from a get archived workflow properties request

## Delete an archived workflow

You can use this operation to remove an archived z/OSMF workflow from a z/OS system.

### HTTP method and URI path

```
DELETE /zosmf/workflow/rest/<version>/archivedworkflows/<workflowKey>
```

In this request, the URI path variables are described, as follows:

- **<version>** identifies the version of the z/OSMF workflow service. The following value is valid: 1.0.
- **<workflowKey>** identifies the archived workflow to be deleted.

### Query parameters

None.

### Description

This operation is used to delete an archived workflow from z/OSMF, including any notes that accompany the workflow and its steps, and the history log for the workflow.

### Request content

None.

### Authorization requirements

For a general workflow or configuration workflow, the ability to delete the workflow is limited to the current workflow owner and members of the z/OSMF workflow administrators group. For a provisioning

workflow, the domain administrator is also able to delete a workflow. A delete request from another user is rejected with the HTTP status code 403 (Forbidden) and an appropriate error message in the JSON response object.

For other authorization requirements, see [“Authorization requirements” on page 726](#).

## HTTP status codes

On successful completion, HTTP status code 204 (No content) is returned.

Otherwise, the following HTTP status codes are returned for the indicated errors. The response body is a standard error response body providing the reason code that is indicated and associated error message.

Table 453. HTTP error response codes for a delete archived workflow request	
HTTP error status code	Description
HTTP 400 Bad request	The request contained an incorrect parameter, such as an incorrect workflow key.
HTTP 403 Forbidden	The requestor user ID is not permitted to delete the workflow properties.
HTTP 404 Not found	The specified workflow key was not found; the workflow does not exist.

Other standard status codes can be returned, as described in [“HTTP status codes” on page 727](#).

## Response content

None.

## Example HTTP interaction

In the following example, the DELETE method is used to delete an archived workflow. The workflow is identified by the workflow key, which is the following string value: 7c4bac42-16a3-4af5-a5b9-263e60b280a4.

```
DELETE /zosmf/workflow/rest/1.0/archivedworkflows/7c4bac42-16a3-4af5-a5b9-263e60b280a4 HTTP/1.1
Host: zosmf1.yourco.com
Connection: close
Authorization: Basic em9zbWZhZDp6b3NtZmFk
```

Figure 400. Sample request to delete an archived workflow

For a successful request, the HTTP response 204 is returned.

```
HTTP/1.1 204 No Content
```

Figure 401. Sample response from a delete archived workflow request

---

## Chapter 2. Creating workflow definitions for z/OS

This information describes how to create workflow definitions that can be used with the Workflows task of z/OSMF. Included is an introduction to workflows concepts and a description of the basic elements of a workflow definition.

### What is a z/OSMF workflow?

Generally, a *workflow* guides you through the complete set of steps that are needed to accomplish a goal, and, when dependencies exist, controls the sequence for performing those steps. In this way, a workflow can help to ensure that the steps are performed in the correct order, and prerequisites and dependences are identified clearly along the way. Conceptually, a workflow encompasses both the work to be performed and its performers. By identifying the individual steps to be performed, a workflow allows for the steps to be divided among different areas of an organization, and different members of a team. Using a workflow, a project owner can delegate specific items to the team members best suited to carrying out particular tasks.

In z/OSMF, a workflow is a guided set of steps that help you perform an activity on z/OS, such as configuring a software product or component, managing a z/OS resource or structure, or simplifying some relatively complex operation. To support these activities, a workflow can be designed to perform a wide variety of operations, such as starting z/OS subsystems, submitting jobs and scripts, and invoking TSO/E functions in batch (assuming that the workflow user is properly authorized).

In short, a z/OSMF workflow:

- Is based on a structured set of steps that are designed by a *workflow author*.
- Is described to z/OSMF through a *workflow definition*. A z/OS organization can write its own workflow definitions or obtain definitions from a third-party source (a *workflow provider*). z/OSMF includes samples for the workflow authors to reference when they create workflow definitions.
- Is created when a user imports a workflow definition into the z/OSMF Workflows task.
- Identifies steps to be performed and allows for these steps to be divided among different areas of an organization, which helps to facilitate user activities on z/OS.
- Contains one or more steps that guide the user through some action to be performed. Steps might consist of manual instructions for performing the steps, or might include some form of guided assistance, such as submitting a batch job, running a REXX script or a shell script, or creating files, based on user inputs.

In z/OSMF, the Workflows task allows a z/OS installation to create and manage workflows for performing activities on the z/OS system. The user who is responsible for the workflow and ensuring that it gets completed is the *workflow owner*. The workflow owner assigns workflow steps to users, making them *assignees* of the step. The user who accepts ownership of a step becomes the *step owner*.

In the Workflows task:

- The **Workflows** page displays the existing workflows for an installation, and provides the control point for creating and managing workflows.
- The **Steps** page displays the steps in a workflow, and provides the control point for managing the steps. From this page, you can select actions for the steps, such as assigning steps, changing ownership of steps, and performing steps.

The following topics provide more details on workflows, workflow steps, and the process of creating workflow definition files for use with the Workflows task of z/OSMF:

- [“Terms you should know” on page 804](#)
- [“The Workflows task schema” on page 806](#)
- [“Creating the workflow definition file” on page 806](#)
- [“Defining steps for your workflow” on page 820](#)

- [“Defining variables for your workflow” on page 856](#)
- [“Workflow XML reference” on page 874.](#)

## Terms you should know

---

Workflow authors should be familiar with the following terms.

### Workflow

**1.** An activity that is associated with the z/OS system, such as configuring a component or product. **2.** The instantiation of a workflow in z/OSMF, based on a workflow definition. A workflow consists of one or more units of work to be performed on the z/OS system, as described by the workflow definition. A workflow is created when the Workflows task is used to create an instance of a workflow from a supplied workflow definition file.

### Workflows task

The task in the z/OSMF desktop that allows users to interact with workflows on z/OS.

### Workflow category

A classification of the activities to be performed in the workflow. In z/OSMF, a workflow that is used to configure system software is classified as a *Configuration* workflow. A workflow that is used to provision system software is classified as a *Provisioning* workflow. All other workflows are classified as *General* workflows.

### Workflow definition

The logical structure of a workflow, represented as a series of one or more steps. The workflow definition identifies the various system objects and actions that constitute activities on z/OS and the rules for performing those activities. The workflow definition includes all of the information that is specified in, or referenced by, the primary XML file (the *workflow definition file*) and possibly other files that are included by the workflow definition file. This content typically includes information about the workflow (such as name and version), step definitions, variable definitions, file templates, and bundle files.

### Workflow definition file

The primary XML file for a workflow definition. A workflow is stored in z/OSMF when its workflow definition file, and optionally, a workflow variable input file, is imported into the Workflows task.

### Workflow variable input file

An optional file that supplies default values for one or more of the input variables that are defined in the workflow definition file. The workflow variable input file is specified as input when the workflow definition is imported into the Workflows task. Typically, a workflow provider might supply a workflow variable input file to save users from having to manually enter inputs when they perform a workflow.

### Job output file

For a workflow that runs a batch job, the workflow author can specify that job spool files are to be saved in a user-specified location (a UNIX directory). The file that is used to save the contents of a job spool file is referred to as a *job output file*.

### Output properties file

A file that is created at the completion of a step. Typically, the output properties file holds the results of a batch job, shell script, or REXX exec program, as determined by the workflow author. The output properties file can be used by other steps or workflow instances. In practice, a step might submit a batch job to create some z/OS related parameters, which are then used by a subsequent step, thus saving the Workflows task user from having to enter the parameter values manually.

### Array variable

A variable definition that can be used to represent a list of values or name-value pairs.

### Global variable

A variable definition that is available to all workflow instances. The Workflows task saves global variables in a repository that is called the global variables pool.

**Note:** Global variables are deprecated, as of z/OS V2R3. IBM recommends that you use instance variables or system variables, instead. Global variables might not be supported in a future release.

**Instance variable**

A variable definition that is available only to instances of a particular workflow.

**Predefined variable**

A variable definition that can be used for string substitution in the current step only.

**System variable**

A variable definition that is created through the z/OSMF system variable services, which is a REST application programming interface. For more information, see [“z/OSMF system variable services” on page 709](#).

**Workflow author**

The person, typically a programmer, who creates the workflow definition by using the XML tagging language.

**Workflow owner**

The user who is given ownership of the workflow through the Workflows task. The workflow owner is responsible for delegating the work in the workflow to users to perform (the step assignees).

**Workflow provider**

The source of the workflow definition file, typically IBM, or a software vendor.

**Step**

A single, logical unit of work in a workflow. Consider each step to describe a specific activity to be performed on the system. A step is available to be performed when the workflow owner assigns the step to a user through the Workflows task, and the user accepts ownership of the step.

**Step owner**

The user who accepts ownership of a step and therefore responsibility for performing the step.

**Automation processing**

The processing of a workflow that contains one or more automated steps. A workflow that is comprised entirely of automated steps can complete with little or no user intervention. When automation processing is started on the workflow, the workflow runs to completion or until it is stopped by another condition, such as a user request or an error.

**Automated step**

A step can be designed to run automatically (without user interaction) when it is in Ready state. Such a step is referred to as an *automated step*. A workflow that is comprised entirely of automated steps can complete with little or no user intervention.

**Batch execution step**

A template step that runs an executable program as a batch job, such as a JCL job, a REXX exec, or a UNIX shell script. Contrast with an *immediate execution step*, which is a template step that runs a program in real time.

**Conditional step**

A step that can be performed when a logical condition is satisfied on the z/OS system or in the Workflows task. For example, a conditional step might become eligible to be performed if a job that is run by another step ends with a particular return code. A conditional step remains unavailable to be performed as long as the condition is not satisfied.

**Called workflow**

A workflow that is started by another workflow for execution. Conceptually, a called workflow is a step in the workflow that calls it (the calling workflow).

**Feedback step**

A step that includes a feedback form with questions for the step owner to answer at the conclusion of a step.

**Immediate execution step**

A template step that runs an executable program in real time, such as a REXX exec or UNIX shell script. Contrast with a *batch execution step*, which is a template step that runs a program as a batch job.

**REST step**

A step that issues a REST API request, such as a GET or PUT request.

### Template step

A step that runs an executable program, such as a JCL job, a REXX exec, or a UNIX shell script. On completion, the results can be made available to other steps, in the form of variables or an output properties file. Depending on how the program is processed, a template step is either of the following:

- *Immediate execution step*, which runs a program in real time
- *Batch execution step*, which runs a program as a batch job.

## The Workflows task schema

---

A valid workflow definition file is one that follows the XML syntax and also conforms to the rules of the Workflows task schema.

The Workflows task schema is supplied with z/OSMF in the following location:

```
/usr/lpp/zosmf/workflow/schemas/workflow_v1.xsd
```

The schema file is UTF-8 encoded.

If you are developing a workflow definition file, you require access to the schema, and therefore access to the z/OS system that is running z/OSMF.

## Creating the workflow definition file

---

This topic describes the elements that comprise the workflow definition file.

A workflow is defined through a workflow definition file, which consists of one or more XML files and other types of files. Depending on the workflow design, a workflow might consist of just a single workflow definition file, or it might have a primary XML file that references one or more subordinate XML files, XML fragments, and external files. This document uses the term *workflow definition file* to refer collectively to all of the files that define a given workflow.

As a workflow author, you can create a workflow definition file in XML, in accordance with the schema that is supplied with the Workflows task of z/OSMF. The schema defines the required and optional properties (XML elements and attributes) of a workflow and imposes constraints on the order in which the elements are specified, and on the values that can be specified for each element and attribute.

It is assumed that workflow authors are familiar with the XML specification and coding practices. The following references provide additional helpful information:

- The World Wide Web Consortium (W3C) XML Technology page: <http://www.w3.org/standards/xml/>
- XML Core Working Group Public Page: <http://www.w3.org/XML/>

Besides XML files, a workflow definition might include external files. That is, apart from XML fragments, the workflow definition can refer to translated text files and velocity template files. These files must be read-accessible by the user who is creating (importing) the workflow in the z/OSMF Workflows task.

You can provide the workflow definition file and any associated files in either a z/OS UNIX file or a z/OS data set. For a z/OS data set, use a sequential data set or a member of a partitioned data set (PDS).

For an example of how you can refer to an external file or fragment from a workflow definition file, see [“Defining entities for a workflow” on page 811](#).

### Workflow Editor task in z/OSMF

To help you with creating and editing a workflow definition, z/OSMF includes an editor for workflows. You can use the Workflow Editor task to view, create, and modify workflow definitions. The Workflow Editor provides a visual framework for working with the elements of a workflow definition—the steps, variables, and workflow metadata.

The Workflow Editor task:

- Presents the details of a workflow definition in a graphical user interface (GUI).

- Provides easy-to-use options for viewing, creating, and modifying a workflow definition.

Using the Workflow Editor task, you can:

- Select an existing workflow definition file for editing. Or, have the Workflow Editor create a starter workflow with which you can begin working.
- View details about the different sections of a workflow definition—the metadata, steps, and variables.
- Modify the workflow information, steps, and variables sections of the definition, including adding and deleting steps and variable definitions.
- Edit the workflow variable input file, if one is available for use with the workflow definition.
- Overwrite the workflow definition with your changes.

To get started with the Workflow Editor task, in the z/OSMF desktop, select **Workflow Editor**.

The workflow definition file must be valid XML, otherwise it cannot be opened in the editor.

If the workflow definition file resides in a z/OS UNIX file path, ensure that the file permissions are set as follows:

- A workflow definition file must allow read and write access for the "user" bit, but requires only read access for the "other" bit (that is, file permissions of at least 604).
- Any external files that are referenced by the workflow definition must have file permissions of at least 604.
- The directory location for these files requires file permissions of at least 505.

If the workflow definition file resides in a data set, ensure that your user ID has write access to the data set. Also, the z/OSMF server user ID, which is IZUSVR by default, requires read access to the data set. To verify that the server has access, contact your z/OSMF administrator.

More information about the Workflow Editor is provided in the online help.

## Structure of a workflow definition file

Structurally, a workflow definition file is comprised of several sections, as follows:

- **Document declaration statements** that are not directly related to workflow content. These statements are required at the beginning of every workflow XML file, and are described as follows:

### XML processing instruction

The primary XML file must start with a processing instruction (in column 1 of line 1) for the XML processor. This instruction defines the version of XML used and the encoding of the file. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

### Document type definition (DTD)

You can optionally use a DTD to define entities (variables) in the workflow. Using a DTD with workflows is optional because z/OSMF uses the default XML schema to validate the contents of a workflow file, rather than the DTD.

### Workflow root element

The workflow root element is the container for the main content of a workflow.

- **Workflow metadata**, which contains information about the workflow itself. For more information, see [“Specifying the workflow metadata” on page 813](#).
- Optionally, a **manifest** of external files that contain translated text for the various user interface elements (widgets) that are displayed for the workflow. The Workflows task of z/OSMF can display text for widgets in the language defined for the browser locale, if you supply the translated text in a properties-based resource bundle file. By default, Workflows task displays the text in whatever language is used in the workflow file. Thus, it is recommended that you use the default language (such as English) in the XML and use bundle files to include any other supported languages. For more information, see [“Including a manifest of translated text” on page 816](#)

- Optionally, one or more **variable definitions**, which you can use to have the Workflows task prompt the user for input values. A number of data types are supported for variables, including string, integer, decimal, boolean, time, and date. A declared variable can be referenced by one or more steps in a given workflow. For more information, see [“Defining variables for your workflow” on page 856](#).
- One or more **step elements** that describe the steps of the workflow. A workflow definition file must include at least one step, and should include all of the steps needed to complete an activity on z/OS (the workflow). For more information, see [“Defining steps for your workflow” on page 820](#).

The reference tables in [“Workflow XML reference” on page 874](#) summarize the basic elements of a workflow definition file, including the attribute values, descriptions, any default values, the XML attribute data types, and whether a particular attribute is required.

## Creating and viewing the workflow definition file

This topic is intended to give application programmers guidance on how to create a workflow definition file.

To be considered valid, a workflow definition file must follow normal XML syntax conventions and also conform to the rules of the Workflows task schema that is supplied with z/OSMF.

### Editing XML files on your workstation

It is recommended that you create and view the workflow XML files on a workstation, rather than on a z/OS system.

When you work with XML files, use a text editor that includes an XML validation function. Validation is the process of comparing your XML files with the Workflows task schema. Doing so ensures that the files use only those tags that are defined in the schema, and ensures that the files conform to the element rules specified in the schema.

To perform the XML validation, you need to transfer the schema file to the XML editor on your workstation. The Workflows task schema resides on the z/OS system according to the z/OSMF product directory path. By default, at this location:

```
/usr/lpp/zosmf/workflow/schemas/workflow_v1.xsd
```

For information about transport protocols, see [“Transferring the workstation files to z/OS” on page 809](#).

### Specifying the processing instruction in the primary XML file

As mentioned in [“Structure of a workflow definition file” on page 807](#), the primary XML file must begin with a processing instruction in column 1 of line 1. This instruction indicates to the XML processor the version of XML used and the file encoding format. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The following encoding formats are valid:

- UTF-8 (ASCII) or IBM-1047 (EBCDIC) for workflow definition files in UNIX files
- IBM-1047 for workflow definition files in z/OS data sets.

You must include a processing instruction in the primary XML file. However, you do not need to specify a processing instruction in any XML fragment files that you include with the primary XML file.

Some workstation XML editors might not recognize the IBM-1047 processing instruction and therefore do not display the file. Also, it is not possible to display a file in EBCDIC on your workstation. As a workaround, if you want a file to be IBM-1047, you can specify UTF-8 in the processing instruction while you are editing the file on your workstation. Then, after you transfer the file to z/OS in ASCII mode, you can edit the file on z/OS to change the processing instruction to IBM-1047. This action allows the Workflows task to process the file.

## Transferring the workstation files to z/OS

It is recommended that you use File Transfer Protocol (FTP) to transfer the XML files to a z/OS system. Doing so helps to ensure that the files are encoded properly for use on z/OS.

For XML files:

- If the processing instruction (the first line in the XML file) indicates that the file uses UTF-8 encoding, transfer the file to z/OS in binary mode, to preserve the file encoding.
- If the processing instruction indicates that the file used IBM-1047 encoding, transfer the file to z/OS in ASCII mode so that the file is converted to EBCDIC.

As supplied by IBM, the Workflows task schema file is encoded in UTF-8. Specify binary mode when you transfer this file to a z/OS system. After you transfer the files to the z/OS system, check the permissions of the transferred files to ensure that they can be opened by the z/OSMF Workflows task. For testing and workflow development purposes, consider setting the file permission of the transferred files to the octal value of 0777 (in "properties").

## Saving a workflow definition on z/OS

If you are saving to a data set, ensure that the data set logical record length (LRECL) is large enough to contain the XML file. Otherwise, the save request fails with an error message. For most workflow definition files, an LRECL of 1024 is large enough.

It is possible to save the workflow definition in a different file format, or in a different location. If you attempt to do so, check the workflow definition for any relative references to external file, including references that are represented by substitution variables. For such references, convert each reference to an absolute path (for a UNIX file) or a fully qualified data set name. This change ensures that the external files can be found after the workflow definition is saved in the new format or location.

As a suggested practice, use the Workflow Editor to create, edit, and save workflow definition files. The Workflow Editor can help to ensure that the workflow definition is saved correctly and that file references are converted properly. For more information, see the Workflow Editor online help.

## Sample XML files for your reference

To help demonstrate various capabilities of the workflow XML schema, z/OSMF includes a number of sample XML files. It is recommended that you load these samples and observe their behavior in the Workflows task as you read this information.

The samples are supplied with z/OSMF in the /samples subdirectory of the product file system, which is, by default: /usr/lpp/zosmf/samples.

Start with the following samples, which show a basic workflow definition, and demonstrate the use of language bundles and variables:

### **workflow\_sample\_basic.xml**

Shows the most basic workflow. It contains one step with only the required elements.

### **workflow\_sample\_variables.xml**

Shows the use of variables that require user input.

More advanced concepts are illustrated in the following samples:

### **workflow\_sample\_array\_variables.xml**

Shows the use of array variables in a workflow.

### **workflow\_sample\_array\_property.txt**

Shows a sample workflow variable input file, for use with workflow\_sample\_array\_variables.xml.

### **workflow\_sample\_automation.xml**

Shows the use of automated steps in a workflow.

### **workflow\_sample\_automation\_property.txt**

Shows a sample workflow variable input file, for use with workflow\_sample\_automation.xml.

**workflow\_sample\_calledwfBasic.xml**

Shows an example of a called workflow. This workflow is called by specifying its workflow ID.

**workflow\_sample\_calledwfMD5.xml**

Shows an example of a called workflow. This workflow is called by specifying its MD5 encrypted value (a 128-bit hash value).

**workflow\_sample\_calledwfVarMapping1.xml**

Shows an example of a called workflow. This example shows how variables can be mapped from the calling workflow to the called workflow.

**workflow\_sample\_calledwfVarMapping2.xml**

Shows an example of a called workflow. This example shows how variables can be mapped from the called workflow to the calling workflow.

**workflow\_sample\_condition.xml**

Shows the use of conditional steps in a workflow.

**workflow\_sample\_feedback.xml**

Shows an example of a feedback form that can be used to gather input on a step from the step owner.

**workflow\_sample\_file\_template0.xml**

Shows the use of a file creation template.

**workflow\_sample\_include\_external.xml**

Shows the use of a DTD to make references to the external files `workflow_sample_fragment0.xml` and `workflow_sample_fragment1.xml`. This sample also demonstrates other features of steps, and uses some HTML tags within a step description.

**workflow\_sample\_output.xml**

Shows an example of writing generated variables to an output file.

**workflow\_sample\_parallel\_steps.xml**

Shows the use of parallel steps in a workflow.

**workflow\_sample\_program\_execution.xml**

Shows an example of running an inline executable program (a UNIX shell script) from within a step.

**workflow\_sample\_predefinedVariable.xml**

Shows the use of predefined variables in workflow steps.

**workflow\_sample\_rexx\_template0.txt**

Shows how to invoke a REXX exec from a workflow.

**workflow\_sample\_substeps.xml**

Shows the use of substeps and the use of step prerequisites to establish dependency chains.

**workflow\_sample\_translation.xml**

Shows a basic workflow that refers to a language bundle file. This workflow is used with `workflow_sample_bundle0.txt`.

**workflow\_sample\_wf2wf.xml**

Is a workflow that calls another workflow for processing.

**workflow\_sample\_wizards.xml**

Shows the use of instructions and wizards that use input variables.

**workflow\_sample\_wizards\_upgrade.xml**

Shows the use of the workflow upgrade function. It upgrades the workflow that is created in the sample file `workflow_sample_wizards.xml`.

## References to external files

When you refer to an external file from the primary XML file, observe the following considerations:

- If the external file resides in a z/OS data set, specify the fully qualified data set name, which is preceded by a double forward slash (//). Do not enclose the data set name in single quotation marks.

For example:

```
//SYS1.PRODUCTX.TESTFLOW  
//SYS1.PRODUCTX(TESTFLOW)
```

- If the external file resides in a UNIX file, you can specify an absolute (fully qualified) path name, or a relative path name (that is, relative to the primary XML file).

When the workflow definition file is imported into z/OSMF, the Workflows task verifies that each referenced file exists and that the user has READ access to the files. The Workflows task then makes copies of the files, and later refers only to the copies.

## Using variable substitution in the workflow definition file path

If you refer to an external file in your workflow definition, you can use variables in the file path name. Doing so allows users of the workflow to customize the file path for their environment. Thus, a file path that uses variables can add flexibility to your workflow definition.

To enable a workflow definition for file path substitution, on the <fileTemplate> element, set the attribute `filePathSubstitution` to `true`. Doing so means that the workflow user is responsible for ensuring that any variables that are used in the file template path must be replaced with valid values.

To supply valid values, the workflow user must edit the workflow input property file and replace the substitution variables with installation-specific values. The user must do this substitution before creating the workflow in the UI. The values that are supplied for the variables in file path are used only during workflow creation time, and cannot be changed during the workflow.

The default value for "filePathSubstitution" is false.

For example, assume that your workflow definition is defined, as follows:

```
<fileTemplate substitution="true" filePathSubstitution="true">  
/u/${instance-filepath}MyTemplate.txt  
</fileTemplate>
```

Here, the workflow user must provide a value for the instance variable "filePath" in the input property file, such as: `filePath=testDir`.

When the user proceeds through the Create Workflow dialog, the Workflows task performs the variable substitution to derive the actual file path for the step: `/u/testDir/MyTemplate.txt`

If your workflow definition is a UNIX file, the <fileTemplate> file path must be a UNIX path. If your workflow definition is a PDS member, the <fileTemplate> file path must be a data set name.

The Workflows task performs validation checking of the file path. A valid file path is one of the following:

- An absolute UNIX path name
- A fully qualified data set name (sequential or PDS) path name (a fully qualified name starting with "///")
- A relative path name, which is relative to the main XML file container. This structure can be a UNIX directory or a PDS. For a PDS, a relative path is the name of a member within the PDS. You cannot specify a relative path when the container is a sequential data set.

### Notes:

1. If you do not specify `filePathSubstitution="true"`, the file template path is treated as a UNIX path even if it contains variables. Remember, UNIX systems support most special characters in directory names, such as "\$", "{", "}".
2. On creation of the workflow, the substituted file path is saved as a property to its corresponding workflow step. It cannot be changed during the workflow, regardless of whether the variable is changed later.

## Defining entities for a workflow

You can use the document type definition (DTD) of XML to define entities in the workflow definition file. Entities are external files, fragments, or variables that can be referenced within the workflow definition

file. The XML processor replaces the references with the values that are specified in the DTD. You might use an entity to define a value that is subject to change as you develop the workflow, and thus can be changed in one place to affect all references.

You can define entities either inline, in one of the workflow XML files, or as system entities, in which case the XML processor obtains the replacement text from an external file.

Observe the following coding considerations:

- An inline entity can be used like a macro instruction. That is, you can define text in one place that is frequently referenced throughout the workflow definition file.
- A system entity is useful if you want to split the workflow definition file into smaller chunks for manageability, or reuse portions across multiple workflow definition files. For example, for a set of steps that is shared across different workflows. A system entity file can reside in the UNIX file system or in a z/OS data set, and the path name format is as described in [“References to external files” on page 810](#).

The path name must be expressed as relative (not absolute). A further restriction for DTD entities is that the referenced entity must reside in the same "container" as the main workflow XML file.

- For a UNIX file, the referenced entity must reside in the same directory or a subdirectory of the primary XML file.
- For a PDS, the referenced entity must reside in the same PDS. Here, a *relative path name* is simply the name of the member within the PDS.

Do not use a sequential data set to store an entity file. Also, be aware that a workflow definition that is contained in a sequential data set cannot refer to external entity files.

An entity file in a PDS member must start with the XML processing instruction, as described in [“Creating the workflow definition file” on page 806](#), with IBM-1047 specified as the encoding format. In fact, any entity file that uses IBM-1047, whether it comes from a data set or a UNIX file, must start with this processing instruction. For UTF-8 files, the instruction is optional.

The following example shows how entities can be defined in the DTD.

```
<!DOCTYPE workflow [<!ENTITY copyright "Copyright IBM Corp., 2013">
    <!ENTITY step1 SYSTEM "step1.xml" >
    <!ENTITY step2 SYSTEM "step2.xml" >
    <!ENTITY step3 SYSTEM "step3.xml" >
]>
```

An entity can be referred to in the document by using the following notation:

```
&copyright;
```

**Tip:** For another example, see the sample `workflow_sample_include_external.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system. This sample shows the use of a DTD to make references to the external files `workflow_sample_fragment0.xml` and `workflow_sample_fragment1.xml`, which are also supplied in the `/samples` subdirectory. The `workflow_sample_include_external.xml` sample also demonstrates other features of steps, and uses some HTML tags within a step description.

## Specifying the workflow root element

Use the workflow root element (`<workflow>`) to specify the XMLSchema-instance namespace, and optionally, a schema location.

For example:

```
<workflow xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="workflow_v1.xsd">
```

The workflow element must end with a closing element: `</workflow>`.

Between the starting and ending element is the body of the workflow definition.

The schema file location specified here is ignored by the Workflows task when it validates an imported workflow definition file. You might find it helpful, however, to use a language-sensitive text editor when creating your workflow definition file. The path you specify indicates the location (relative to the file being edited if expressed as a relative path name) of the schema. Such editors can provide immediate feedback if you violate a rule of the schema, and might provide type-ahead support as you enter elements and attributes.

Though you might develop your workflow primarily on your workstation, and even though your editor is not flagging any errors, you do not know for sure that you have created a valid workflow until after you have imported it into the Workflows task of z/OSMF. Observe the following coding considerations:

- Differences exist among the various implementations of XML schema validators. Thus, a workflow definition file that validates in your text editor might not validate when it is imported into the Workflows task.
- The Workflows task performs additional validation that is not enforced by the schema, for example, validating that any message or template files that are referenced in the workflow actually exist on the z/OS system.

## Specifying the workflow metadata

Use the workflow information element (<workflowInfo>) to specify metadata for the workflow, such as the workflow identifier, description, version, and vendor, and possibly other details. The Workflows task displays the metadata to users in the Workflows table and the **Properties for workflow** page. Users can use the metadata values for filtering or configuring the workflows view.

The workflow information element is required.

You can specify the following sub-elements for the workflow information element:

### parallelSteps

For a workflow with automated steps, this property indicates whether the automated steps can be run in parallel (concurrently), thus possibly completing more quickly. For a parallel-steps workflow, set this property to `true`. Otherwise, if this property is omitted or set to `false`, automated steps are run one-by-one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.

### workflowID

Short, arbitrary value that identifies the workflow. This element is required.

This element can include the optional attributes *scope* and *isCallable*. For more information, see [“Setting the workflow scope” on page 815](#) and [“Callable workflows” on page 815](#).

### workflowDefaultName

Default name for the workflow. The default name is shown in the **Workflow name** field of the Workflows task when a user creates the workflow. The user can accept this name, or specify a different name.

The workflow name:

- Must be unique in the Workflows task.
- Can contain up to 100 characters. Leading and trailing white space is ignored.
- Must not contain the characters for ampersand ('&'), forward slash ('/'), logical or ('|'), greater than ('>'), or less than ('<').

The workflow name is not case-sensitive; for example: MyWorkflow and MYWORKFLOW are the same workflow.

This element is optional. If you omit it, the Workflows task assigns a name to the workflow, based on the following convention:

```
<workflow-description> - Workflow_<number>
```

Where:

- *workflow-description* is the description from the workflow definition file.
- *number* is the first available number, beginning at 0. If you later delete this workflow, its number can be reused by the Workflows task.

### **workflowDescription**

Short description of the workflow. This element is required.

When a workflow is imported into z/OSMF, the Workflows task uses the workflow description (specified in the <workflowDescription> tag) to create a default workflow name, which the user can change.

### **workflowVersion**

Version of the workflow definition. This element is required.

The Workflows task caches only the latest version of any imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you change any portion of the workflow definition. This includes changes to the primary XML file or any subfiles or referenced files. For this reason, when you author a workflow definition, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.

### **vendor**

Name of the workflow provider. This element is required.

### **jobsOutputDirectory**

Indicates a location to be used for automatically storing saved job output files from the workflow. Use this option if you want to retain job output files, perhaps as a record of the work that is done by the workflow. Specify a valid UNIX file path and directory on the user's system, beginning with a single forward slash ('/'). For example: /u/IBMUUSER/jobFiles. In the Workflows task user interface (UI), the workflow owner can overwrite this value with a different location, as needed.

If this option is enabled, a value must be specified. The value must be an existing directory on the user's system. Otherwise, the Workflows task cannot create the workflow; an error message is displayed in the Workflows task UI.

The job output files are saved in the IBM-1047 encoding format. This format is viewable in a z/OS Console by using the **cat** command. As an alternative, the user can use an FTP tool to view or download the files with the transfer type ASCII.

Observe the following authorization requirements:

- Workflow owner user ID requires write access to the directory.
- For the steps that create job output, the step owner user IDs require write access to the directory. Otherwise, the steps cannot be performed.

If the Job Output Location option is not enabled, the workflow does not save its job output files.

Table 465 on page 876 describes the elements that make up the workflow metadata.

Figure 402 on page 814 shows an example of the metadata XML tags for a workflow.

```
<workflowInfo>
  <parallelSteps>true</parallelSteps>
  <workflowID scope="system" isCallable="system">SampleWorkflow</workflowID>
  <workflowDefaultName>WORKFLOW#001</workflowDefaultName>
  <workflowDescription>A simple workflow</workflowDescription>
  <workflowVersion>1.0</workflowVersion>
  <vendor>XYZ, Inc.</vendor>
  <General/> <!-- This element is empty, and completely optional. -->
  <jobsOutputDirectory>/u/IBMUUSER/jobFiles</jobsOutputDirectory>
</workflowInfo>
```

*Figure 402. General metadata for a workflow*

## Setting the workflow scope

It is possible to have multiple instances of a workflow run at the same time, on the same system or sysplex. However, in some cases you might need to restrict a workflow to one instance only, which can be useful for coordinating actions across a system or sysplex. In z/OSMF workflows, this concept is referred to as the *singleton scope* for the workflow.

To set the singleton scope for a workflow, specify the optional attribute *scope* on the sub-element (<workflowID>). Specify the scope attribute with one of the following values:

### **system**

A maximum of one instance of this workflow can exist on any one system in the sysplex.

### **sysplex**

A maximum of one instance of this workflow can exist in the sysplex.

### **none**

No limit exists for the number of instances of this workflow. For a callable workflow, this setting means that a new instance is always created on the calling system.

Workflow scope has ramifications for callable workflows. You might, for example, determine that a new instance of the workflow should always be created in response to a calling workflow. Or, you might prefer to have an existing instance of a workflow be used to handle the request. For considerations, see [“Coordinating workflow-to-workflow actions” on page 841](#).

## Callable workflows

In z/OSMF, a workflow can invoke another workflow for processing. A workflow that invokes another workflow is the *calling workflow*. A workflow that is invoked by another workflow is the *called workflow*.

To indicate whether a workflow is eligible to be called by another workflow, specify the optional attribute *isCallable* on the workflow sub-element (<workflowID>). On the *isCallable* attribute, also indicate the callable *range* for the workflow, that is, whether the workflow can be called only within the same system, or whether it can be called from any system in the same sysplex. Specify the *isCallable* attribute with one of the following values:

### **system**

This workflow can be called only by another workflow that is running on the same system.

### **sysplex**

This workflow can be called by any workflow that is running in the same sysplex.

If you omit the *isCallable* attribute, the workflow cannot be called by another workflow.

More information about callable workflows is provided in [“Calling steps” on page 840](#).

## Workflow category

In z/OSMF, a workflow category is a classification of the activities that are to be performed in a particular workflow. Workflows that are used to configure system software such as a product or component are *Configuration* workflows. Workflows that are used to provision system software, such as Db2 or IMS, are *Provisioning* workflows. All other workflows are *General* workflows.

The category is specified by the workflow author in the workflow definition file. The workflow category is optional. If no category is specified for the workflow, the workflow belongs to the General category. [Figure 402 on page 814](#) shows an example of how the workflow category is specified: <General/>.

For a Configuration workflow or a Provisioning workflow, the workflow metadata specifies product-specific details, such as the product name, product version, and product ID. The metadata for a Provisioning workflow also specifies the type of software to be provisioned (software type). The Workflows task presents this information to users in the **Category Specific** tab on the **Properties for workflow** page.

The following example shows the metadata for a Configuration workflow:

```
<workflowInfo>
  <workflowID>Productx-01</workflowID>
  <workflowDefaultName>WORKFLOW#001</workflowDefaultName>
  <workflowDescription>Initial configuration of Product X</workflowDescription>
  <workflowVersion>1.0</workflowVersion>
  <vendor>IBM</vendor>
  <Configuration>
    <productID>abc123</productID>
    <productName>Product X</productName>
    <productVersion>V1R99</productVersion>
  </Configuration>
  <jobsOutputDirectory>/u/IBMUUSER/jobFiles</jobsOutputDirectory>
</workflowInfo>
```

In this example, the metadata provides product-specific fields in the <workflowInfo> element, as follows:

**productID**

Identifier of the product or component that is being configured through the workflow, such as the product identifier (PID) or function modification identifier (FMID).

**productName**

Name of the product or component that is being configured through the workflow.

**productVersion**

Version and release of the product or component that is configured through the workflow.

## Including a manifest of translated text

Use the translated text files element (<translatedTextFiles>) to define the *message manifest* for a workflow. The message manifest contains one or more bundle definitions, each of which contains one or more language file definitions. A *bundle* is a logical grouping for a set of text elements used in the workflow.

To refer to these files, other text elements in the workflow definition can specify a bundle name and an identifier (or key) within the language files that contain the replacement text for that language. The Workflows task maps the bundle name and language into the file defined for that language and displays the text in the language that is in effect for the browser, if a language file is provided.

The Workflows task uses only the browser language to locate the translation file; the country is ignored. For example, en-US and en-UK are two versions of English: American, and British, respectively. The Workflows task uses only the value "en" to locate the language file.

The following example shows a message manifest:

```
<translatedTextFiles>
  <bundle name="StepMessages">
    <language name="en" path="steps/english.txt"/>
    <language name="fr" path="steps/french.txt"/>
  </bundle>
  <bundle name="VariableLabels">
    <language name="en" path="vars/english.txt"/>
    <language name="fr" path="vars/french.txt"/>
  </bundle>
</translatedTextFiles>
```

For an example of referencing a bundle in a translatable element, see the sample file **workflow\_sample\_translation.xml**, which is supplied with z/OSMF in the /samples subdirectory of the product file system.

[Table 469 on page 886](#) describes the elements that make up the message manifest.

## Enabling a workflow definition file for future upgrades

Over the course of time, you might want to provide users with a revised version of your workflow, presumably with enhanced functions. This topic describes the tags that you can use to make your workflow definition upgradeable.

In z/OSMF, to *upgrade* a workflow means to create a new instance of an existing workflow, based on a new definition file. As the workflow author, you can design your workflow definition to be upgradeable to future versions of the definition.

During an upgrade operation, the Workflows task user upgrades an existing workflow to a new level of the workflow by using the action **Create New Based on Existing**, which is provided in the Workflows table. By upgrading the workflow, the user creates a new instance of the workflow, while retaining data from the existing workflow.

### Creating an upgradeable workflow definition

Using a set of tags in the Workflows schema, you can add an upgrade capability to your workflow definition. The tags specify which previous versions of the workflow are supported for upgrades. Also, the rules for data mapping, which variables, steps and attributes can be copied forward, and so.

To provide upgrade option in a workflow definition, you must provide a workflow definition with same workflowID as the current (old) version of the workflow definition.

To add an upgrade capability to your workflow definition, you define an upgrade element (<preserveOptions>) with the attributes described in this topic.

Start by defining the <preserveOptions> element after all of the defined steps in the new workflow definition. On the <preserveOptions> element, you can specify the following elements and attributes:

- **Version.** This element specifies information about the workflow to be upgraded. It contains the following attributes:
  - **Value.** Version of the workflow definition that is supported for upgrading. You must specify at least one prior version of the workflow definition.
  - **Type.** Specify either patch or release, as follows:
    - **Patch** indicates that the upgrade to is intended to fix a defect in the prior version of the workflow. Based on this idea, the new workflow replaces, rather than coexists, with the prior version. Thus, the prior version is canceled when the upgrade is performed.
    - **Release** indicates that the upgrade creates a new release of the workflow. The prior version is retained, based on the assumption that some installations with multiple releases might choose to use more than one version of the workflow.
- **Variable set (<variableSet>).** This element specifies which variable values are to be copied from the existing workflow. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default) causes the option **Copy attribute values based on upgrade definition** in the Workflows task wizard **Create New Based on Existing** to be selected by default. If so, all of the variable values are copied to the upgraded workflow.
- **Step set (<stepSet>).** This element specifies the step values to be copied from the existing workflow. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default) causes the option **Copy step attributes based on upgrade definition** in the Workflows task wizard **Create New Based on Existing** to be selected by default. If so, all of the current workflow step attributes are copied to the upgraded workflow. The attributes include the step assignees, step owners, step states, notes, submitted job status, and the record of how variable conflicts, if any, were resolved.
- **Workflow history (<workflowHistory>).** This element specifies whether the workflow history is copied from the existing workflow to the new instance. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default), causes the option **Copy workflow history** in the Workflows task wizard **Create New Based on Existing** to be selected by default. Otherwise, the option is not selected.

- Workflow notes (<workflowNotes>). This element specifies whether workflow notes are copied from the existing workflow to the new instance. On this element, you can optionally specify the attribute (<defaultChecked>). Setting it to true (the default), causes the option **Copy workflow notes** in the Workflows task wizard **Create New Based on Existing** to be selected by default. Otherwise, the option is not selected.
- Include (<include>). Use this element to specify which steps or variables in the prior workflow definition file are copied to the new workflow. You can specify this element multiple times. Specify either a regular expression or a variable name.
- Exclude (<exclude>). Use this element to specify which steps or variables are excluded from the set that is generated by the <include> element.
- Upgrade notes (<upgradeNotes>). Use this element to provide the user with information about the upgraded workflow definition. For example, you might use this element to provide the user with details about using the workflow.

For descriptions and data types of these elements and attributes, see [“Workflow upgrade elements summary” on page 883](#).

The <include> element in the variable set specifies which variables are to be copied, based on either a regular expression (regExp) or the variable name. The following example specifies that all of the existing steps that match the specified regular expression are to be copied to the same steps in the new workflow definition:

```
<include regexp="\w★">
```

To copy the value from the variable that is named *setting3* in an existing workflow to the same variable in the new workflow definition, you might code the following:

```
<include name="setting3"/>
```

To copy the value from the variable that is named *setting1* in an existing workflow to the variable named *setting2* in the new workflow definition, you might code the following:

```
<include name="setting1" mapTo="setting2"/>
```

If you specify the <include> element multiple times in a sequence, and a variable is included for multiple times after calculation by name and regular expression, the last <include> specification is the one that is used for the copy operation.

## Collecting user feedback

It is possible to collect feedback from the users of a workflow. A workflow author can optionally include a feedback form on one or more steps with customized questions for the step owner to answer at the conclusion of a step. Such feedback can be useful for determining the effectiveness of a workflow design, or collecting user requirements for future enhancements to a workflow. Inclusion of a feedback form is optional; answering the questions in a feedback form can be optional or required, as determined by the workflow author.

The workflows XML schema includes elements to help workflow authors create questions for users. The Workflows task includes functions to allow the workflow owner to prompt users for feedback, and collect the responses into a consolidated document. When all of the required feedback is provided, the workflow owner can send the feedback to the workflow author for evaluation.

### How feedback is collected

Collecting feedback from the users of a workflow involves the participation of the following roles:

#### Workflow author

Workflow author defines feedback questions in the workflow definition file, and designates the questions as optional or required.

## Workflow owner

For a workflow that includes feedback questions, the workflow owner is responsible for collecting the feedback. From the Workflows table, the workflow owner can select Feedback to launch actions that are related to feedback. The workflow owner can display pages to see which steps require feedback, which steps have incomplete feedback, and options for notifying the step owners who need to complete feedback.

## Step owner

For a step that includes a feedback form, the step owner is responsible for providing feedback by answering questions about the step. To answer feedback for a step, the step owner selects a step and selects the table action Feedback. Only the step owner can display the feedback page for a step. The Feedback action is disabled for steps that do not contain feedback questions.

When all of the required feedback is provided by step owners, the workflow owner can save the accumulated feedback into a feedback file. On the Generate Feedback Summary page, the workflow owner can create a report of the feedback, which can be sent to the workflow vendor for evaluation.

## Schema elements

As the workflow author, you define feedback questions and answers in the workflow definition file.

### Example

Figure 403 on page 819 shows how to define each of the question types in a workflow definition file.

In the example:

- `itemOne` defines a multiple choice question
- `itemTwo` defines an either or choice
- `itemThree` defines a question that accepts a write-in response.

```
<!-- ===== feedback definitions ===== -->
<feedbackItem name="itemOne">
  <question>How difficult was this step?</question>
  <!--The user must select one answer from the available choices-->
  <answers>
    <singleSelect hasOtherAnswer="true">
      <label value="difficult">Very difficult</label>
      <label value="moderate">Somewhat difficult</label>
      <label value="moderate">Neutral</label>
      <label value="moderate">Somewhat easy</label>
      <label value="easy">Very easy</label>
    </singleSelect>
  </answers>
</feedbackItem>

<feedbackItem name="itemTwo">
  <question>What did you like about this step?</question>
  <!--The user can select more than one of the available choices-->
  <answers>
    <multipleSelect hasOtherAnswer="true">
      <label value="simple">Ease of use</label>
      <label value="info">Instructions were helpful</label>
      <label value="useful">Performed a useful function</label>
      <label value="quick">Ran quickly</label>
    </multipleSelect>
  </answers>
</feedbackItem>

<feedbackItem name="itemThree">
  <question>How would you describe your experience with this step?</question>
  <!--The user supplies a text response (up to 500 characters) -->
  <answers>
    <text/>
  </answers>
</feedbackItem>
```

Figure 403. You can define various questions for step owners to answer.

The questions that are defined in Figure 403 on page 819 can be referenced by the steps in your workflow. Figure 404 on page 820 shows how the questions defined earlier can be included in the step

definitions. Notice the attribute `required` is included for questions that require an answer from the step owner.

```
<!-- ===== step with feedback questions ===== -->
<step name="StepOne" >
  <title>A step with feedback</title>
  <description>A step with feedback.</description>
  <!--On the feedback tag, the attribute "name" identifies the
        feedback question. The attribute "required" indicates whether
        a user response is required or optional. -->
  <feedback name="itemOne" required="true"/>
  <feedback name="itemTwo" required="true"/>
  <feedback name="itemThree" required="true"/>
  <instructions>This step has three questions for you to answer.
    All of the questions require a response.</instructions>
  <weight>1</weight>
</step>
:
<!-- ===== Another step with feedback questions ===== -->
<step name="StepTwo" >
  <title>Another step with feedback</title>
  <description>Another step with feedback.</description>
  <!--On the feedback tag, the attribute "name" identifies the
        feedback question. To indicate that a user response is required,
        the attribute "required" is included and set to true. To indicate
        that a response is optional, you can set required to false or
        omit this attribute. -->
  <feedback name="itemOne" required="true"/>
  <feedback name="itemThree"/>
  <instructions>This step has two feedback items. itemOne is required,
    and itemThree is optional.</instructions>
  <weight>1</weight>
</step>
```

Figure 404. How feedback questions can be included in the steps in your workflow.

## Defining steps for your workflow

A workflow is composed of one or more units of work called *steps*. A workflow definition file must contain at least one step; each step can contain substeps. In the Workflows task, a wizard guides the user through a step, which can be either manual or automated. For a manual step, the wizard only displays the instructions that are required for the user to perform the step. For an automated step, the wizard uses a template to create a file or run a REXX exec, UNIX shell script, or JCL job. Optionally, the wizard can perform substitution by referencing variable values.

Every step has a name and contains a title and description. Steps can contain substeps, which can also contain substeps, up to five levels of nesting. For example, Step 1 can contain Step 1.1, which can contain Step 1.1.1, which can contain Step 1.1.1.1, which can contain Step 1.1.1.1.1. The total aggregation of steps and their substeps across the entire workflow cannot exceed 500.

The step name must be unique across the entire workflow. The step name is not displayed in the Workflows task, but it is used within the workflow to reference prerequisite steps. The title should be brief. It is displayed in the step table when the workflow is opened. Step titles are indented for substeps. The step description can be more detailed. It is displayed in the "General" tab under step properties to provide a bit more context, if necessary.

A step can be designated as optional. This designation has effects on how its weight is used in the calculation of the percentage-complete for a workflow. Weights are described in ["Parent steps and leaf steps"](#) on page 821.

Any step can optionally contain a set of references to prerequisite steps (by step name) which must be completed before this step can be performed. Each prerequisite is identified in its own `<prereqStep>` element. The Workflows task displays the prerequisite chain for a step in the "Details" tab for the step properties. The Workflows task also keeps track of dependencies with regard to the states of the steps.

Any prerequisite step must have been defined previously in the workflow (though not an ancestor of the referencing step). If the reference is to a step with substeps, the substeps are treated as prerequisites (they need not be explicitly specified). If only certain substeps under a parent step are dependencies, they can be listed explicitly.

A step can contain multiple prerequisites steps, but these need not be cumulative. That is, if Step3 depends on Step2, and Step2 depends on Step1, then Step2 would identify Step1 as a dependency, and Step3 would identify Step2 as a dependency. If Step3 identified both Step2 and Step1, no harm would occur, but it makes the workflow more complex than necessary, and possibly more difficult to maintain over time.

A step can be defined as *conditional*. Such a step is available to be performed based on whether a logical condition is satisfied on the z/OS system. For example, a conditional step might become eligible to be performed if a job run by another step ends with a particular return code. A conditional step, which depends on a logical condition, is different than a dependent step, which depends on a particular step being completed. For more information, see [“Making a step conditional” on page 850](#).

What is described here applies to any step. For the distinctions between a parent step and a leaf step, see [“Parent steps and leaf steps” on page 821](#).

[Table 470 on page 888](#) describes the elements that make up a step.

## Parent steps and leaf steps

A *parent step* contains a set of nested step elements (at least one). A step with no substeps is called a *leaf step*.

A leaf step actually performs, or tells the user how to perform, the actions required to complete the step. Leaf steps contain, at a minimum, instructions and a weight. Optionally, a leaf step can also contain a skills category, a template (for file creation or job submission), and references to variables which can be substituted into the instructions or template, or both.

The skills element (<skills>) specifies a suggested skills category for performing the step, such as "Security administration" or "Network administration." The Workflows task displays this value in the step table for a workflow. This value is free-form; specify it at your discretion.

The weight element (<weight>) specifies the relative difficulty of the step (a positive integer value from 1 to 1000). The Workflows task uses this value in the calculation of the percentage-complete value that is displayed as the workflow is performed. The scale is arbitrary; specify it at your discretion. Specify a lesser value for a step in which the user performs a simple action, such as cutting and pasting some a command text that you provide. For a more complicated task, such as deploying a digital certificate on z/OS, specify a greater value.

The instructions element (<instructions>) defines the content of the *Review Instructions* tab of the Step Perform wizard in the Workflows task. In this element, you provide the detailed instructions on what the user must do to perform the step to completion. The instructions can contain certain HTML tags for formatting, and hyperlinks to refer to additional information. The instructions can also use variable substitution. The <instructions> element is required, and must contain some content. Otherwise the step cannot be performed (the Workflows task Perform tab is omitted for the step).

In the Workflows task, the *Review Instructions* tab is displayed after variable-prompting. Thus, instructions can contain substituted values. However, instructions cannot be used to guide the user through the variable gathering stage. The variable attributes (title, abstract, and description) must be sufficient to guide the user, though you can also provide some guidance or context in the step description.

*Review Instructions* is the final tab that the user sees when performing a manual step. The user is expected to follow the instructions and then press **Finish** to complete the step.

Usually, a step must be marked complete before the workflow can continue. Depending on your design, however, you might want to allow the user to mark a step as *Failed* manually. This option might be useful if a step cannot be performed manually (outside of the workflow). To enable this option, include the can-mark-as-failed element (<canMarkAsFailed>) for the step and set it to true. Doing so causes the *Review Instructions* tab to prompt the user to confirm whether the step could be completed manually. If appropriate, the user can mark the step as *Failed* and continue with the workflow. By default, this option is disabled (the can-mark-as-failed element is set to false).

Leaf steps can contain optional references to variables that were defined earlier in the workflow definition. Based on your requirements, you might need to allow the step owner to modify the value of a referenced variable. To do so, identify the variable reference with the variable value element (<variableValue>). If so, the Workflows task displays a wizard to guide the user through entering values for the variable when performing the step, as described in [“Defining variables for your workflow” on page 856](#). If a step only references a variable for read and not for modification, you do not need to specify the <variableValue> element.

**Note:** Array variables, which are used to map multiple values, are not shown in the Workflows task wizard.

You can code symbols using the Velocity syntax in both the instructions and the template that can be replaced by the value of the variable as entered by the user. More information is provided in [“Using Velocity templates for variable substitution and other functions” on page 856](#).

You can determine how and when your workflow should prompt the user for variable values. For instance, you could have a step that is used only to prompt for the variables, which are then referenced by subsequent steps. Or, you can prompt for the variable directly within the step that uses it.

When coding the variable reference within a step, you can specify whether the variable is required. If a variable is required, and the user does not enter a value for it, the Workflows task prevents the user from proceeding until a value is specified. If a variable is optional, and the user does not enter a value for it, the Workflows task allows the user to advance to the edit screen, and the Velocity symbol in the instructions or template is displayed as-is without substitution. You can, however, use conditional Velocity statements to generate different text based on whether a variable has a value. If the symbol is left unresolved, a user might not understand these references, depending on how you name the variable. An unresolved variable in a template will likely result in an incorrect file or executable. However, the user can optionally edit the files before saving them or running them.

The variable reference also specifies whether to allow the user to change the value if the variable has one already. As a workflow author, you might know that the variable has already been used for an important purpose, such as allocating a file, and that, if the user changes the value, the overall procedure defined in the workflow would be corrupted. Here, you should not allow the user the opportunity to change the value. Instead, the Workflows task displays the value in read-only mode. The user can still override the value, after responding to a confirmation prompt that includes a list of any other steps that reference the same variable. For any variable that is referenced by additional steps, the Workflows task displays a list of those steps, along with the variable description, when the user presses the information icon. Even if you make a variable read-only when it already has a value, the user can override the value by checking a box in the description to make the value editable, thus accepting the consequences of doing so.

## Template steps

A step that runs a program, such as a JCL job, a REXX exec, or a UNIX shell script, is referred to as a *template step*. The program that is run by the step is referred to as a *template*. This topic describes how to write a template step so that you can run programs and batch jobs in your workflow.

### Program design considerations

For a template step, you must determine:

- Whether to include the executable code inline (within the step XML tagging), or in a separate, external file that is referenced from the step.
- How the executable code (the template) is processed — in real time or submitted as a batch job. You can:
  - Run a program in real time for immediate results. The program can be a REXX exec or a UNIX shell script.
  - Submit a job for batch processing. The batch job contains JCL and might also imbed an executable program that runs under batch, such as a REXX exec or UNIX shell script.

For more information, see [“Running a program in real time” on page 828](#) and [“Submitting a JCL job for batch processing” on page 831](#).

- Whether the program uses variables (substitution values). You can use variables to prompt for input from the Workflows task user. In the Workflows task user interface (UI), the variable value is presented to the user, who can override it. For more information about variables, see [“Using variables in a template step”](#) on page 826.
- Whether the program contents can be modified by the end user, and if so, where the modified template is saved. For more information, see [“Saving the contents of the template”](#) on page 827.
- Whether the program creates a properties file, which contains variables for subsequent steps to reference. For more information, see [“Creating a properties file”](#) on page 827.

## How the user interacts with a template step

When a step includes a template element, the Workflows task user interface (UI) enables the **Next** button on the wizard instructions page. When the user presses **Next**, the wizard guides the user through the activity, such as running a program or job, or creating and saving a properties file. The behavior of the wizard is further controlled through the elements that you define within the template element.

## How automation processes a template step

For an automated workflow, the Workflows task performs the step under the user's identity. As a result, the user can view the results of the step processing, such as a job status or a properties file.

The user's identity is either the step owner user ID or a runAsUser user ID, if the element runAsUser (<runAsUser>) is included on the step element. When a runAsUser is not specified for a step, the step is performed under the step owner user ID. For more information about the runAsUser element, see [“runAsUser identity for a step”](#) on page 852.

## Elements of a template step

In a workflow definition, a template step is defined with the element <template> and its related elements and attributes in the Workflows XML schema. The elements are used to indicate where to find the program that is run, how the program is processed (submitted in batch or executed in real time), and other details. [Table 454 on page 823](#) provides a summary of the template step elements.

How the program is processed — immediately or in batch — determines which <template> elements are applicable for use with the <template> element. Indicate your choice of program processing by using the element <submitAs> with the appropriate setting. In [Table 454 on page 823](#), the applicable elements for each processing type are indicated in the columns **Used in an immediate execution step** and **Used in a batch execution step**.

<i>Table 454. Summary of template step elements</i>			
Element Name	Description	Used in an immediate execution step	Used in a batch execution step
<b>template</b>	Identifies the step as a template step, which is a step that runs an executable program, such as a JCL job, a REXX exec, or a UNIX shell script.	Yes	Yes

Table 454. Summary of template step elements (continued)

Element Name	Description	Used in an immediate execution step	Used in a batch execution step
<b>inlineTemplate</b>	<p>Indicates that the program is included inline, that is, within the step XML structure. This element is mutually exclusive with the &lt;fileTemplate&gt; element.</p> <p>In the following example, a REXX program is provided inline within the &lt;inlineTemplate&gt; element.</p> <pre>&lt;template&gt;   &lt;inlineTemplate substitution="true"&gt;/* rexx */   parse arg arg1   SAY "this is a sample to submit TSO REXX script     to run immediately"   SAY "the first parameter is :" arg1   SAY \${instance-st_user}   SAY "outvar:st_group =" SYS123   SAY "outvar:st_user =" USERS   SAY "This execution will meets timeout."   &lt;/inlineTemplate&gt; &lt;/template&gt;</pre>	Yes	Yes
<b>fileTemplate</b>	<p>Indicates that the program is contained in an external file, and specifies the location of the file. This element is mutually exclusive with the &lt;inlineTemplate&gt; element.</p> <p>Specify the location as either the full path name of the file, beginning with the forward slash (/) and including the file name, or a relative path. If the file resides in the same directory as the workflow definition, specify the file name only.</p> <p>In the following example, the program to be executed (workflow_sample_rexx_template0.txt) resides in the same UNIX directory as the workflow definition file:</p> <pre>&lt;template&gt;   &lt;fileTemplate substitution="true"&gt;     workflow_sample_rexx_template0.txt   &lt;/fileTemplate&gt;   &lt;submitAs maxRc="0"&gt;TSO-REXX-JCL&lt;/submitAs&gt; &lt;/template&gt;</pre>	Yes	Yes
<b>submitAs</b>	<p>Indicates the type of program processing to be performed, as described in “Running a program in real time” on page 828 and “Submitting a JCL job for batch processing” on page 831.</p>	Yes	Yes

Table 454. Summary of template step elements (continued)

Element Name	Description	Used in an immediate execution step	Used in a batch execution step
<b>maxLrecl</b>	<p>Specifies the maximum record length, in bytes, for the input data for the job. This value is used when the step is performed automatically (autoEnable=true).</p> <p>If the step is performed manually, the user can optionally specify the maximum record length on the <b>Edit JCL</b> page in the Workflows task UI.</p> <p>For more information, see <a href="#">“Maximum record length” on page 833</a>.</p>	No	Yes
<b>saveAsUnixFile</b>	<p>Specifies the path name for saving the executable code as a UNIX file after the user edits the template in the Workflows task. Must be specified as a full path name of the file, beginning with the forward slash (/) and including the file name.</p> <p>The presence of this element results in the <b>save as UNIX file</b> option being presented to the user, primed with the element value, if specified.</p> <p>For more information, see <a href="#">“Saving the contents of the template” on page 827</a>.</p>	Yes	Yes
<b>saveAsDataset</b>	<p>Specifies the data set name for saving the executable code after the user edits the template in the Workflows task. Must be specified as a fully qualified data set name, without quotations.</p> <p>The presence of this element results in the <b>save as data set</b> option being presented to the user, primed with this value, if specified.</p> <p>For more information, see <a href="#">“Saving the contents of the template” on page 827</a>.</p>	Yes	Yes
<b>successPattern</b>	<p>Specifies a regular expression that is returned for a successful program execution. This element is required. You must specify one (and only one) regular expression for a successful program completion. For more information, see <a href="#">“Completion messages” on page 830</a>.</p>	Yes	No
<b>failedPattern</b>	<p>Optionally specifies a regular expression that is returned for program execution failures. You can omit this element or specify up to 100 different specifications for &lt;failedPattern&gt;. For more information, see <a href="#">“Completion messages” on page 830</a>.</p>	Yes	No
<b>outputVariablesPrefix</b>	<p>Specifies a prefix for variables that are created by the step. For more information, see <a href="#">“Writing variables to a properties file” on page 829</a>.</p>	Yes	Yes

Table 454. Summary of template step elements (continued)

Element Name	Description	Used in an immediate execution step	Used in a batch execution step
<b>scriptParameters</b>	Contains the input parameters that can be set by the step owner. This element is optional. For more information, see <a href="#">“Input parameters” on page 829</a> .	Yes	No
<b>procName</b>	Specifies the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC. For more information, see <a href="#">“TSO/E address space for program execution” on page 829</a> .	Yes	No
<b>regionSize</b>	Specifies the region size for the TSO/E address space. If no value is specified for the step, the default is 50000. For more information, see <a href="#">“TSO/E address space for program execution” on page 829</a> .	Yes	No
<b>timeout</b>	Specifies the maximum amount of time that the program can run before it is ended by a timeout condition. For more information, see <a href="#">“TSO/E address space for program execution” on page 829</a> .	Yes	No
<b>output</b>	Specifies the default name and location of the properties file that is created by the step. The properties file can contain variables and values that are used by subsequent steps. The Workflows task UI allows the user to modify the file name and location, as needed. For more information, see <a href="#">“Creating a properties file” on page 827</a> .	Yes	Yes

#### Notes:

1. Avoid creating an inline template with characters that interfere with XML. If this problem occurs, use a file template instead. Also, when you use Velocity comparison operators in the instructions, do not use the less than (<) or greater than (>) characters, as they interfere with XML. Instead, use the alternative notation: lt, le, gt, and ge.
2. No white space manipulation occurs for an inline template. Use care when you type the template into the XML, based on the context of the template. To write JCL, for example, start the first line of the template immediately after the beginning <inlineTemplate> element. Subsequent lines must start on column 1, rather than being indented for readability.

### Using variables in a template step

The "substitution=" attribute on the template element indicates whether the template contains variables. The default is false. If you specify "substitution=true," you must also specify at least one variable value element (<variableValue>) in the step. Otherwise, the workflow fails with a validation error when the user attempts to import the workflow definition into the Workflows task.

Observe the following considerations:

- If you include more variables than you reference, no errors result.
- If you refer to variables that are not included, but you do include at least one variable, no validation errors result. However, the Workflows task cannot resolve the variables during the substitution process.

## Saving the contents of the template

To specify the location of an external file for saving the template, include the `<saveAsDataset>` element or the `<saveAsUnixFile>` element for the step. The location can be a sequential data set, partitioned data set (PDS), or z/OS UNIX path and file name. The Workflows task UI presents the user with the option to save the template in this location.

By including the `<saveAsDataset>` element or the `<saveAsUnixFile>` element, you control whether the template is saved to a data set or a z/OS UNIX file, respectively. To allow the user to save to either data format, include both elements in the step definition.

The template contents are always saved in EBCDIC encoding (code page IBM-1047).

Providing neither element has the same result as defining both elements without a value. Consider it a shorthand way of indicating that you can tolerate the file in either a data set or a UNIX file, and you have no suggestion as to what to name the output. For example, if you do not care about which data set the user selects for saving the program, you can provide an empty element like this: `<saveAsDataset/>`.

You can set these elements to a default location. If so, the location is initialized in the widget that is presented to the user, who can override it. If you omit a default value, the user must specify a valid location.

Table 455 on page 827 provides more details about these elements.

Table 455. Elements for saving a generated job	
Element Name	Description
<b>saveAsDataset</b>	<p>Specifies the data set name for saving the executable code after the user edits the template in the Workflows task. Must be specified as a fully qualified data set name, without quotations.</p> <p>The presence of this element results in the <b>save as data set</b> option being presented to the user, primed with this value, if specified.</p>
<b>saveAsUnixFile</b>	<p>Specifies the path name for saving the executable code as a UNIX file after the user edits the template in the Workflows task. Must be specified as a full path name of the file, beginning with the forward slash (/) and including the file name.</p> <p>The presence of this element results in the <b>save as UNIX file</b> option being presented to the user, primed with the element value, if specified.</p>

The `<saveAsDataset>` and `<saveAsUnixFile>` elements accept variable substitution. Suppose, for example, that you want to provide a default data set name for saving a job, but also allow the user to specify the data set high-level qualifier. Here, you could specify the `<saveAsDataset/>` element like this:

```
<saveAsDataSet substitution="true">${instance-hlq}.MYPROD(CONFIG)</saveAsDataset>
```

If you plan to save to a data set, be aware that the data set must already exist (be allocated) when the workflow runs. Otherwise, an error results. The Workflows task does not allocate the data set that you specify on the `<saveAsDataset>` element. Similarly, the path and file that you specify on `<saveAsUnixFile>` element must exist when the workflow runs. Otherwise, an error results.

## Creating a properties file

You can design a step to run a job to create variables, and save them to a *properties file*. When a step creates a properties file, the Workflows task reads in the contents of the file and saves its values for use by other steps in the same workflow.

To specify the default name and location of the properties file that is produced by the step, include the element `<output>` on the step element. The properties file can be a data set, a UNIX file, or a JES spool file, as follows:

#### Data set

Specify a fully qualified data set name, without quotations.

Omit the `sysoutDD=` attribute from the output element.

#### UNIX file

Specify a full path name of the file, beginning with the forward slash (/) and including the file name.

Omit the `sysoutDD=` attribute from the output element.

#### JES spool file

Include the `sysoutDD=` attribute on the output element and set this attribute to `true`.

Specify the spool file by using the following syntax:

```
[step.]ddname
```

Where:

- *step* is the name of the job step that creates the spool file. This value is optional. If no job step name is specified, z/OSMF attempts to find a matching job step by checking the results of the executed job. z/OSMF reads from the most recently created DD name to the earliest to find a match. To avoid ambiguity, it is recommended that you always specify the step name.
- *ddname* is the DD name for the step. This value is required.

When a step includes an output element, the Workflows task UI enables the **Edit Output File Path** option in the step perform wizard. The Workflows user can modify the file name and location, or choose a different file, as needed. If the `sysoutDD=` attribute is set to `true`, the Workflows task UI enables the **Edit Output DD Name** page in the step perform wizard. The Workflows user can modify the DD name, as needed.

When a properties file is processed, the Workflows task scans the file to determine whether the file contains any workflow input variables (properties that are written as name-value pairs). If so, the Workflows task attempts to add the variables to the variables pool. The input variables then become available for use by subsequent steps in the workflow.

#### Notes:

1. The identity of the user who performs the step is used to read the resulting properties file. If the step is performed under a `runAsUser` user ID, the `runAsUser` ID is used to read the file.
2. As a suggested practice, your workflow can clean up the properties file when it is no longer needed. If the file is a spool file, this action is not necessary because JES purges the file from the spool as part of normal processing.
3. If a spool file is used, and the file is purged before it is read by the Workflows task, the step is marked *Failed*.

## Running a program in real time

A template step that runs a program in real time is called an *immediate execution step*.

To run an executable program (a REXX exec or UNIX shell script) in real time, include one of the following attributes on the `submitAs` element (`<submitAs>`):

#### TSO-REXX

Run a REXX exec program in real time.

#### TSO-UNIX-REXX

Run a REXX exec program for the UNIX environment in real time.

#### TSO-UNIX-shell

Run a UNIX shell script in real time.

The program results are immediately available to the step owner.

## Input parameters

To enable an executable program to receive input parameters from the step owner, include the element `<scriptParameters>` element with the following attributes:

### **<description>**

Text description of the parameter, such as its intended use or recommended value.

### **<value>**

Default value of the parameter.

## Writing variables to a properties file

In your program, you might choose to create variables, and save them to a properties file. When the file is saved, the properties are available for use with the same workflow. For more information, see [“Creating a properties file” on page 827](#).

Another workflow can access the properties file when the file is:

- Specified in the **Output File** field in the Workflows task UI.
- Read in as a workflow variable input file on creation of the workflow.

For a program that creates variables in a properties file, you can optionally include a prefix on the output variables. Here, you can specify a meaningful prefix that identifies a string as an output variable. To do so, add the element `<outputVariablesPrefix>` to the `<submitAs>` element.

In the following example, the string `outvar:` is applied to the names of any variables that are created by the step:

```
<outputVariablesPrefix needResolveConflicts="true">outvar:</outputVariablesPrefix>
```

To manage potential variable name conflicts, you can specify a default behavior by adding the attribute `<needResolveConflicts>` to the element `<outputVariablesPrefix>`. By default, the step owner is prompted to choose the appropriate variable value in the Workflows task.

For an array variable, variable name conflicts are handled differently. The step owner is not prompted to choose the appropriate variable value. Instead, variable name conflicts are handled by the behavior you specify on the `(loadOutputFileArrayValue)` attribute. Add this attribute to the element `<outputVariablesPrefix>` and set it to true or false, as follows:

- If set to true (the default), the workflow uses the array variable values from the output file, rather than from the Workflows task.
- If set to false, the workflow uses the existing values from the Workflows task.

## TSO/E address space for program execution

The executable program runs in a TSO/E address space on the z/OS host system. You can control how the TSO/E address space is started by including the following elements:

### **<regionSize>**

Region size (in kilobytes) to be used for the address space. The valid range of values is 50000 - 2096128 (kilobytes). The Workflows task UI allows the step owner to specify a different region size, or use the supplied value. If no value is specified, the region size is 50000 kilobytes, by default.

### **<procName>**

TSO logon procedure to be used for the address space. If no value is specified, the default is to use IZUFPROC, which is supplied by IBM as a cataloged procedure in proclib.

### **<timeout>**

Time limit for an executable program. The valid range of values is 60 - 3600 (seconds). If no value is specified, the timeout value is 1800 seconds (30 minutes), by default.

## Completion messages

On completion of program execution, standard TSO/E service messages are written to the script log. As the workflow author, you can supplement the TSO/E messages with your own customized messages to indicate successful completion or an error condition. To do so, add the following sub-elements to the `<submitAs>` element:

### **<successPattern>**

Successful completion message. This element is required. You must specify one (and only one) regular expression for a successful program completion.

The format is:

```
<successPattern>success regular expression</successPattern>
```

### **<failedPattern>**

Error message. This element is optional. You can omit this element or specify up to 100 different expressions for potential program errors.

The format is:

```
<failedPattern>failure regular expression</failedPattern>
```

Messages are limited in size, based on the program type, as follows:

#### **TSO-REXX**

The maximum length is 10000 lines.

#### **TSO-UNIX-REXX**

The maximum length is 255 characters; extra characters are truncated.

#### **TSO-UNIX-shell**

The maximum length is 255 characters; extra characters are truncated.

## Temporary locations

Before running the program, z/OSMF saves the program to a temporary file. On completion of the program execution, z/OSMF deletes the temporary file. The location of the temporary file depends on the program type, as follows:

#### **TSO-REXX**

userID.IZUWFTSO.W, plus a randomly generated 7-digit number.

#### **TSO-UNIX-REXX**

/tmp/fileName-scriptRexxFile.rexx, where *fileName* is a randomly generated 32-digit number.

#### **TSO-UNIX-shell**

/tmp/fileName-scriptShellFile.sh, where *fileName* is a randomly generated 32-digit number.

Other related output files are saved to the following temporary locations:

- Variable output file is saved at /data/app/IZUWorkflows-workflowKey/outputFiles/stepName-outputfile
- Message output file is saved at /data/app/IZUWorkflows-workflowKey/scriptFile/stepName-scriptOutputMessageFile

## Examples

In [Figure 405 on page 831](#), a template definition contains inline REXX commands for immediate processing. For more examples of running programs from a step, see file `workflow_sample_program_execution.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

```

<step name="TSO-TSO-REXX_Execution">
  <title>A step that runs a REXX exec program.</title>
  <description>In this step, an inline REXX exec is processed immediately on the host system.
    The processing is ended by a time-out condition.
  </description>
  <variableValue name="st_group" required="true"/>
  <variableValue name="st_user" required="true"/>
  <variableValue name="procNameVariable" required="true"/>
  <instructions substitution="false">This step outputs some variables and prints a few words.
  </instructions>
  <weight>1</weight>
  <skills>System Programmer</skills>
  <template>
    <inlineTemplate substitution="true">/* rexx */
    parse arg arg1
    SAY "this is a sample to submit TSO REXX script to run immediately"
    SAY "the first parameter is :" arg1
    SAY ${instance-st_user}
    SAY "prefix:st_group =" SYS123
    SAY "prefix:st_user =" USERS
    SAY "This execution will meets timeout."
    </inlineTemplate>
    <submitAs>TSO-REXX</submitAs>
    <successPattern>success.*</successPattern>
    <failedPattern>failed.*</failedPattern>
    <outputVariablesPrefix needResolveConflicts="true">prefix:</outputVariablesPrefix>
    <scriptParameters>
      <description>A simple parameter</description>
      <value>para1</value>
    </scriptParameters>
    <procName substitution="true">${instance-procNameVariable}</procName>
    <regionSize>50000</regionSize>
    <timeout>60</timeout>
    <saveAsUnixFile substitution="true">/u/${instance-st_user}/savedStuff/myScript.sh</saveAsUnixFile>
  </template>
</step>

```

Figure 405. This sample step submits a REXX exec for immediate processing.

## Submitting a JCL job for batch processing

A template step that runs a program as a batch job is called a *batch execution step*. The submit as element (<submitAs>) indicates the type of program to be run.

To run an executable program as a batch job, such as a JCL job, a REXX exec, or a UNIX shell script, you include one of the following attributes on the submit as element (<submitAs>):

### JCL

Submit a JCL job for batch processing on z/OS. The results are indicated in the job log.

### TSO-REXX-JCL

Submit a JCL job that contains a REXX program. The program runs as a batch job on z/OS; the results are indicated in the job log.

### shell-JCL

Submit a JCL job that contains a UNIX shell script. The program runs as a batch job on z/OS; the results are indicated in the job log.

Each program type is executed as a batch job by the Workflows task, which creates the necessary JCL and JOB statement, and displays the job output in the Workflows task. The size of any program to be run (JCL, REXX, or shell) is limited to 10000 lines of code.

In the following example, a template definition contains inline JCL that runs a TSO command:

```

<template>
  <inlineTemplate>//STEP1          EXEC  PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD    SYSOUT=A
//SYSTSIN  DD    *
LISTUSER IBMUSER
/*
  </inlineTemplate>
  <submitAs>JCL</submitAs>
</template>

```

Suppose that the first line of the JCL were to be placed on the next line after the beginning <inlineTemplate> tag. Doing so would inject a space into the JCL stream and cause a JCL error.

Similarly, if any of the lines of JCL were to be indented, a JCL error would occur. Therefore, unless the file or program is small, use `<fileTemplate>` instead, which identifies the path name of an external file that contains the template. For path name examples, see [“References to external files” on page 810](#).

In the following example, the file that is named `jcljob.txt` contains the JCL and exists in the same UNIX directory as the workflow definition file:

```
<template>
  <fileTemplate>
    jcljob.txt
  </fileTemplate>
  <submitAs>JCL</submitAs>
</template>
```

The Workflows task routes the job to another system in the sysplex, as determined by the system name that is chosen by the user when importing the workflow. Both JES2 and JES3 are supported. The JOB statement is entered by the Workflows task user separately, and applied to the job stream before it is submitted. Do not include a job card in your JCL template.

Observe the following considerations:

- A JCL template is submitted by the Workflows task after it includes the user-specified JOB statement and the appropriate JES routing statement, if the job is to be run on a different system in the sysplex.
- A REXX template is run by an IKJEFT01 job. This job creates a temporary data set to contain the REXX exec, and then executes the exec from that data set.
- A shell script is run by the BPXBATCH program. A temporary directory and file is created to contain the script. The script is run from this location and then the temporary directory and file are deleted.

## Writing variables to a properties file

In your program, you might choose to create variables, and save them to a properties file. When the file is saved, the properties are available for use with the same workflow. For more information, see [“Creating a properties file” on page 827](#).

For a program that creates variables in a properties file, you can optionally include a prefix on the output variables. Here, you can specify a meaningful prefix that identifies a string as an output variable. To do so, add the element `<outputVariablesPrefix>` to the `<submitAs>` element.

In the following example, the string `outvar:` is applied to the names of any variables that are created by the step:

```
<outputVariablesPrefix needResolveConflicts="true">outvar:</outputVariablesPrefix>
```

To manage potential variable name conflicts, you can specify a default behavior by adding the attribute `<needResolveConflicts>` to the element `<outputVariablesPrefix>`. By default, the step owner is prompted to choose the appropriate variable value in the Workflows task.

For an array variable, variable name conflicts are handled differently. The step owner is not prompted to choose the appropriate variable value. Instead, variable name conflicts are handled by the behavior you specify on the `(loadOutputFileArrayValue)` attribute. Add this attribute to the element `<output>` and set it to true or false, as follows:

- If set to true (the default), the workflow uses the array variable values from the output file, rather than from the Workflows task.
- If set to false, the workflow uses the existing values from the Workflows task.

## Maximum return code

On the `<submitAs>` element, you can optionally indicate the maximum acceptable return code from the program on the `maxRc=` attribute.

When the user presses the **Close** button from the Status tab of the **Perform** page (which displays the job output), the Workflows task checks the job return code. If the return code is less than or equal to the

maxRc value, the Workflows task UI marks the step as complete. Otherwise, the Workflows task UI marks the step as failed.

JCL limits the return code value to the range 0 — 4095. If you do not specify a return code, the Workflows task uses a value of zero by default. BPXBATCH, which runs a shell script, further limits the return code to the range 0 — 15. If you are familiar with BPXBATCH, you might be aware that this service multiplies the script return code by 256 to derive a final return code value. The Workflows task divides the result by 256. Thus, you can code your workflow to be consistent with the value that is returned by the script.

## Maximum record length

For a step that submits a job, you can use the `<maxLrec1>` element to specify the maximum record length, in bytes, for the input data for the job. This value is used when the step is performed automatically (`<autoEnable>` is set to true; see [“Automated steps” on page 846](#)). If the step is performed manually, the user can optionally specify the maximum record length on the **Edit JCL** page in the Workflows task. If you omit the `<maxLrec1>` element, the default is 1024. For a job that uses the IEBUPDTE program to create or modify data sets, set this value to the minimum value of 80 to ensure that fixed-length records of 80 bytes or less can be processed.

## Examples

For examples of submitting batch jobs from a step, see file `workflow_sample_wizards.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

## Using variables in template steps

You can add customization capabilities to template steps through the use of step-specific variables and other techniques for designing steps, as described in this topic.

As the workflow author, you might want to:

- Specify certain variables as being able to be used in a JOB statement.
- Use the same JCL repeatedly over a number of steps, but with a different set of values substituted for each job submission.

This topic describes techniques that you can use to add flexibility to template steps, in the following topics:

- [“Variable substitution in the JOB statement” on page 833](#)
- [“Predefined variables in a step” on page 834](#)

## Variable substitution in the JOB statement

As the workflow author, you might want to indicate that certain variables in the workflow can be used in a JOB statement. To make a variable available for user selection, include the `exposeToUser` (`<exposeToUser>`) element on the variable element tag. This element indicates that the variable is to be included among the user-selectable variables in the **List variables for substitution** window of the Workflows task.

With the `exposeToUser` element, you can identify a certain set of variables that can be used (through substitution) in the JOB statement when the step owner is performing the step. To use a variable, the step owner would have to modify the JOB statement to add the variable reference. The step owner can then view the substitution to see whether the JOB statement is as desired. The step owner can optionally designate the edited JOB statement be used for all steps in the workflow instance, but must understand the impact of doing so.

Suppose, for example, that you want to allow the user to select a notification user ID for the job. The system will send a message to this user ID when the job completes.

You can use the following technique to present the user with a selectable user ID FRED that is intended to be used with the NOTIFY= keyword parameter in the JOB statement. On the variable definition, include the exposeToUser element:

```
<variable name="notify-user">
<label>Notify user ID</label>
<abstract>This user ID can be selected for job notifications</abstract>
<description>Simple string variable.</description>
<exposeToUser>
<usage>Use this variable with the NOTIFY= parameter of the JOB statement
so that its value will designate who is notified when the job completes.</usage>
</exposeToUser>
<category>exposeToUser variables</category>
<string>
<default>FRED</default>
</string>
</variable>
```

In the workflow step, when the user displays the **Create JOB statement** page, the default JOB statement for the workflow is shown. For example, the following JOB statement is supplied with z/OSMF:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
MSGLEVEL=(1,1),REGION=0M,NOTIFY=IBMUSER
```

On the **Create JOB statement** page is the option **List variables for substitution**. If the user clicks this option, the Workflows task displays the variables in the workflow that include the exposeToUser element. The user can make JOB statement substitutions by manually copying these variables into the JOB statement.

In this example, the notify-user variable would be displayed in the **List variables for substitution** window. To use the notify-user variable for the notification user ID, the user can copy it into the JOB statement, as follows:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
MSGLEVEL=(1,1),REGION=0M,NOTIFY=${instance-notify-user}
```

On selecting the **View JOB Statement Substitutions** option, the user can display the substituted values to verify that the JOB statement is correct. In this example, the notification user ID is resolved to the value FRED:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
MSGLEVEL=(1,1),REGION=0M,NOTIFY=FRED
```

## Predefined variables in a step

Suppose that your workflow contains a number of steps that run the same job repeatedly, except for some values in the body of the job that must change each time that the job is submitted. As an alternative to maintaining a slightly different copy of the JCL for each step, you can use the same job and call it from multiple steps. This technique requires that each step supply its own substitution values or *predefined variables* for each job submission. Doing so allows you as the workflow author to adjust the contents of the job dynamically to suit each step, based on the predefined variables.

To specify a predefined variable for a step, include the predefined variable (<predefinedVariable>) element on the step element tag, with the following element attribute:

### name=

Name of the variable (a string).

A predefined variable is treated as a string substitution for the current step only. You can specify multiple predefined variables per step. To avoid overriding the variables that are defined for the workflow, use a unique name for the predefined variable.

# REST steps

You can design a step to invoke a Representational State Transfer (REST) service. A step that calls a REST service is referred to as a *REST step*. This topic describes the use of REST steps and the Workflows schema elements that you can use to create them.

A typical use for a REST step would be to obtain values for your workflow and assign the values to variables defined in the workflow. Suppose, for example, that your workflow requires an IP address and a port number from a REST endpoint. A REST step could be used to invoke this REST endpoint and map the values from the REST endpoint response body to your IP address and port number variables.

An example of a REST step that obtains an IP address and port number is shown in [Figure 406 on page 839](#).

## Notes:

1. The REST service to be called must support the use of JSON as the content-type for the request body and the response body.
2. It is assumed that you understand the layout of the returned data and know where to obtain specific values from the JSON string for mapping to variables.
3. The returned values can be mapped to instance variables only. You cannot modify the values of global variables.

## How the user interacts with a REST step

From the user's perspective, when a step definition includes the `<rest>` element, the Workflows task enables the **Perform** button on the step instructions page. When the user presses **Finish**, the perform wizard issues the REST request. The behavior of the wizard is further controlled through the sub-elements that you define within the `<rest>` element.

## Elements of a REST step

In a workflow definition, a REST step is defined with the `<rest>` element and a number of sub-elements and attributes that provide details about the REST request. The details include the HTTP method, scheme name, URI path, query parameters, request body, expected status code, and actual status code. The `<rest>` element also defines any mapping properties that you can use to save the JSON properties (in the response body) to workflow variables.

[Table 456 on page 835](#) provides a summary of the REST step elements.

Table 456. Summary of REST step elements				
Element Name	Description	Required or optional	Supports substitution	Default value
httpMethod	Indicates the HTTP method that is used for issuing the REST request. The following values are valid: <ul style="list-style-type: none"><li>• GET</li><li>• PUT</li><li>• POST</li><li>• DELETE.</li></ul>	Required	No	None

Table 456. Summary of REST step elements (continued)

Element Name	Description	Required or optional	Supports substitution	Default value
<b>schemeName</b>	<p>Hypertext transfer protocol (HTTP) header for the request.</p> <p>The following values are valid::</p> <ul style="list-style-type: none"> <li>• If you specify HTTP, you must also specify a host name for the receiving system. Optionally, you can specify a port number for the host name. The z/OSMF default port for HTTP requests is 80.</li> <li>• If you specify HTTPS, you must also specify the host name for the receiving system, and the user name and password that can be used to log in to the receiving system. Optionally, you can specify a port number for the host name. The z/OSMF default port for HTTPS requests is 443.</li> <li>• If you omit the scheme name and host name, the REST request is sent to the local z/OSMF server.</li> </ul> <p>For more considerations, see the Table Notes.</p>	Optional	No	None.
<b>hostname</b>	<p>Host name or IP address of the system to which the REST request is directed. For example <code>www.ibm.com</code>.</p> <p>For more considerations, see the Table Notes.</p>	Optional	Yes	Host name that is defined for z/OSMF, which is "*" by default.
<b>port</b>	<p>Port number to use for the REST request.</p> <p>For more considerations, see the Table Notes.</p>	Optional	Yes	Port number that was defined for z/OSMF, which is "80" for HTTP requests and "443" for HTTPS requests.

Table 456. Summary of REST step elements (continued)

Element Name	Description	Required or optional	Supports substitution	Default value
<b>username</b>	<p>z/OS user ID that allows the user to access the specified URI. This value is a user ID that is defined to your installation's z/OS security management facility (for example, RACF).</p> <ul style="list-style-type: none"> <li>For an HTTPS request, the user name and password are required. These values are used to create a secure request to the specified URI.</li> <li>For an HTTP request, the user name and password are optional. These values are used for basic authentication only.</li> </ul>	Optional	Yes	None
<b>password</b>	<p>Password or pass phrase that is associated with the user name. For an HTTPS request, the user name and password are required.</p> <p>The Workflows task applies Base64 decoding to this value. Therefore:</p> <ul style="list-style-type: none"> <li>To specify a password value on this element, encode the value by using Base64 encoding and specify the result here. For example, the string password would be specified as cGFzc3dvcmQ==.</li> <li>To use variable substitution for the password, ensure that the password variable is included in the workflow input variable file for the workflow, and is specified in Base64 encoded form.</li> </ul>	Optional	Yes	None
<b>certificates</b>		Optional	Yes	None
<b>uriPath</b>	URI path to use for the REST request.	Required	Yes	None
<b>queryParameters</b>	For a GET or POST request, this element contains the query parameters.	Optional	Yes	None
<b>requestHeaders</b>	This element contains the request header.	Optional	Yes	None
<b>requestBody</b>	For a PUT or POST request, this element contains the request body.	Optional	Yes	None
<b>expectedStatusCode</b>	The expected HTTP status code from the REST API request. If this value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a job template step returns a return code that is greater than the allowed maximum return code.	Required	No	None

Table 456. Summary of REST step elements (continued)

Element Name	Description	Required or optional	Supports substitution	Default value
<b>actualStatusCode</b>	The actual HTTP status code that is received from the REST request. To obtain this value, map it to a workflow variable.	Optional	No	None
<b>propertyMapping</b>	The property from the REST response body that is mapped to a workflow variable. You can specify multiple propertyMapping elements in a REST step.	Optional	No	None

**Table notes:**

1. Scheme name and host name, and the optional port number, are used together. You must specify both scheme name and host name, or neither of them. Otherwise, the workflow fails validation when the user attempts to import the workflow definition into the Workflows task.
2. You must include the "mapTo" attribute on the <actualStatusCode> element. Set it to the name of the workflow variable that is to receive the actual status code value from the REST request. For example:

```
<actualStatusCode mapTo="status_code" />
```

**Mapping the returned data to variables**

You can assign the values that are returned from a GET request to variables that are defined in your workflow. To do so, include the <propertyMapping> element on the <rest> tag. This element identifies the required values in the JSON response body and saves the values to workflow variables. On the <propertyMapping> element, specify the value to be assigned on the "mapTo" attribute. You can specify multiple <propertyMapping> elements in a REST step.

In the following example, the workflow owner is defined in a property mapping.

```
<propertyMapping mapTo="workflow_owner">["name"]</propertyMapping>
```

If the response body is:

```
{ "name": "John" }
```

the workflow variable "workflow\_owner" is assigned the value John.

**Specifying substitution variables for REST step elements**

You can specify substitution variables as values for the elements <hostname>, <port>, <uriPath>, <queryParameters>, and <requestBody>. Doing so allows you to set the values for these elements dynamically from the workflow variables.

For an element that contains variable substitution, do the following:

- Include the optional attribute "substitution" and set it to *true*. If not specified, the default is *false*. A value of *true* must be specified for the Workflows task to allow the variable substitution.
- Set the element value to *\${instance-WORKFLOW\_VARIABLE\_NAME}*, where *WORKFLOW\_VARIABLE\_NAME* is the workflow variable that is defined in the workflow.

A sample REST step is shown in [Figure 406 on page 839](#). In this example, variable substitution is used in the <requestBody> element to allow dynamic substitution for the stackId and stackname properties.

```

<step name="get_ip_and_port_data">
  <title>Get values for workflow variables ip_addr and port from REST API call.</title>
  <description>Invoke RESTful API to get IP address and port number.</description>
  <prereqStep name="get_stack_data_for_rest_request"/>
  <instructions substitution="false"> Execute step to retrieve the IP address and port number
  from z/OS Communications Server.</instructions>
  <weight>10</weight>
  <skills>REST</skills>
  <rest>
    <httpMethod>POST</httpMethod>
    <uriPath>/zosmf/workflow/WorkflowManager/cloud/ipaddr/</uriPath>
    <requestBody substitution="true">
      {
        "name" : "GGAIPA",
        "description" : "IP address created by GGA",
        "ipaddr" : "1.1.1.2",
        "usageType" : "Internal",
        "workloadDeploymentGroup" : "/wdg/12345",
        "deploymentId" : "deployerA",
        "recoveryMethod" : "MANUAL_DISRUPTIVE",
        "hostName" : "mvstst1",
        "stack" :
          {
            "stackId" : "${instance-stack_id}",
            "stackname" : "${instance-stack_name}"
          },
        "boundTcpPorts" : [
          "4",
          "81"
        ]
      }
    </requestBody>
    <expectedStatusCode>201</expectedStatusCode>
    <actualStatusCode mapTo="status_code" />
    <propertyMapping mapTo="ip_address">["ipaddr"]</propertyMapping>
    <propertyMapping mapTo="port0">["boundPorts"][0]</propertyMapping>
    <propertyMapping mapTo="port1">["boundPorts"][1]</propertyMapping>
  </rest>
</step>

```

*Figure 406. Sample REST step definition with substitution variables and property mapping variables*

In this example, several returned values are mapped to workflow variables, as follows:

- `actualStatusCode` is mapped to the `"status_code"` variable.
- `"ip_address"` property is mapped to the `"ipaddr"` variable from the JSON object in the response body.
- `"port0"` property is mapped to the first element in an array of ports in the JSON object in the response body.
- `"port1"` property is mapped to the second element in an array of ports in the JSON object in the response body.

## Example of a REST step

```
<step name="get_system_data_for_rest_request" optional="false">
  <title>Get System values now.</title>
  <description>Invoke REST API to get an unnamed array of variables.</description>
  <variableValue name="query_variant" />
  <variableValue name="body_sub" />
  <variableValue name="path_sub" />
  <instructions substitution="false">Execute this step to retrieve system values for this
workflow.
</instructions>
<weight>10</weight>
<skills>REST</skills>
<autoEnable>true</autoEnable>
<canMarkAsFailed>false</canMarkAsFailed>
<rest>
  <httpMethod>GET</httpMethod>
  <schemeName substitution="false">https</schemeName>
  <hostname substitution="false">www.mycompany.com</hostname>
  <uriPath substitution="true">/zosmf/${instance-path_sub}</uriPath>
  <queryParameters substitution="true">${instance-query_variant}</queryParameters>
  <requestBody substitution="true">${instance-body_sub}</requestBody>
  <expectedStatusCode>200</expectedStatusCode>
  <actualStatusCode mapTo="status_code"/>
  <propertyMapping mapTo="array2element3">["items"][0]</propertyMapping>
  <propertyMapping mapTo="array3element2">["items"][1]</propertyMapping>
  <username substitution="false">ibmuser</username>
  <password substitution="false">c3lzMQ==</password>
</rest>
</step>
```

## Calling steps


Suppose that your workflow needs a function that is already performed by another workflow on your system. Rather than attempt to replicate the same function in your workflow, you can add a step to start the other workflow as needed. A workflow that starts another workflow is the *calling workflow*. A workflow that is started by another workflow is the *called workflow*. On completion, the called workflow returns control to the calling workflow.

From the point of view of the calling workflow, the called workflow is simply another "step" to be performed. You define the called workflow in the workflow definition file for the calling workflow, on the step element (<step>), through a set of elements and attributes.

With the ability to call one workflow from another, you can follow a modular approach to designing workflows. You can include a common, frequently performed, action in one workflow and allow other workflows to call that workflow, as needed. By including one or more called workflows in your design, you can maximize reuse of your code by using "common workflows," and effectively, integrate several workflows together.

To make a workflow eligible to be called by another workflow, include the *isCallable* attribute on the workflow metadata element <workflowInfo>, as described in [“Callable workflows” on page 815](#).

## How the user interacts with a calling step

In the Workflows task user interface, a step that starts a called workflow is indicated to the user with the following icon in the **Workflow Steps** page: 

In the Workflows task, when the user reaches a step that calls another workflow, z/OSMF determines whether an instance of the called workflow is already active on the system. If so, the Workflows task redirects focus to the called workflow so that the step owner can complete it. If the called workflow is already completed, the Workflows task displays the Workflows Steps table, showing the step that started the called workflow as marked complete.

Otherwise, if the called workflow is not already created, the Workflows task opens to the **Create Workflow** page so that the user can create the workflow. The user can supply a name for the workflow on the **Create Workflow** page, or accept the default workflow name.

## Coordinating workflow-to-workflow actions

The Workflows schema metadata attributes *scope* and *isCallable* can be useful for coordinating workflow-to-workflow actions across a system or sysplex. By setting these attributes in various combinations, you can achieve the following effects:

- Restrict the number of instances of a workflow to one instance (a *singleton*).
- Always attempt to use an existing instance of a workflow in response to a calling workflow. Or, always cause a new instance of a workflow to be created, even if an instance exists already.
- Limit the use of a callable workflow to the same system or sysplex.

The *scope* and *isCallable* attributes are specified on the workflow metadata element (<workflowInfo>), as described in [“Callable workflows” on page 815](#).

Nine combinations of *scope* and *isCallable* are possible, as shown in [Table 457 on page 841](#).

Table 457. Use <i>scope</i> and <i>isCallable</i> to coordinate workflow-to-workflow actions		
scope value	isCallable value	Effect on the called workflow
system	system	Workflow is limited to one instance per system, and can be called on that system only. If an instance does not exist, an instance is created.
sysplex	system	Workflow is limited to one instance per sysplex, and can be called from the same system as the calling workflow.
none	system	A new instance is always created. The instance can be called from the same system as the calling workflow.
system	sysplex	Workflow is limited to one instance per system, and can be called from any system in the sysplex. If an instance does not exist, an instance is created.  For an automated workflow, if the calling step is performed automatically, the called workflow is searched for only on the calling system. If an instance is found, it is used; otherwise a new instance is created on the calling system.
sysplex	sysplex	Workflow is limited to one instance per sysplex, and can be called from any system in the sysplex. If an instance does not exist, an instance is created.
none	sysplex	A new instance is always created. The instance can be called from any system in the sysplex.
system	none (omitted)	Workflow is not callable.
sysplex	none (omitted)	Workflow is not callable.
none	none (omitted)	Workflow is not callable.

## How workflow access type is handled

When a workflow calls another workflow for processing, z/OSMF changes the access type for the called workflow to match the calling workflow. This processing ensures that the requested access type is applied consistently to both of the workflows in a calling relationship.

However, this processing is not performed when the called workflow is limited to one active instance in the system or sysplex.

## Designing a step to call another workflow

To have a step call another workflow, specify the step element (<step>) with the elements for defining a called workflow. The major step elements for defining a called workflow in your workflow definition file are described in [Table 458 on page 842](#).

Table 458. Major step elements for defining a called workflow in your workflow definition file		
Element name	Description	Required or optional
<b>variableMapping</b>	Used to transfer instance variable values between the called workflow and calling workflow, and specify options for sharing variables.  More information about this element and its sub-elements is provided in <a href="#">“Sharing variables between the calling workflow and called workflow” on page 843</a> .	Optional.
<b>callingStepWeight</b>	Specifies the relative difficulty of the step compared to other steps within this workflow (a positive integer value 1 – 1000). The Workflows task uses this value in the calculation of the percentage-complete value that is displayed for the calling workflow. The scale is arbitrary; specify it at your discretion. Consider the difficulty of the called workflow as single step among the other steps in the calling workflow.	Required.
<b>callingStepSkills</b>	Specifies a suggested skills category for performing the step, such as "Security administration" or "Network administration." The Workflows task displays this value in the step table for a workflow. This value is free-form; specify it at your discretion.	Optional.
<b>callingStepAutoEnable</b>	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required. If <i>callingStepAutoEnable</i> is not specified, the default is <i>false</i> .  More information about designing steps to run automatically is provided in <a href="#">“Automated steps” on page 846</a>	Optional.
<b>canCallingStepMarkAsFailed</b>	Indicates whether the step can be marked as <i>Failed</i> manually by the step owner. When set to <i>true</i> , the <b>Review Instructions</b> page in the Step Perform wizard includes the option to allow the step owner to mark a step as <i>Failed</i> manually. When <i>false</i> , this option is not displayed to the user.  If <i>canCallingStepMarkAsFailed</i> is not specified, the default is <i>false</i> .	Optional.
<b>calledWorkflowDefinitionFile</b>	Specifies the external file that contains the workflow definition for the called workflow. Provide the absolute (fully qualified) path name, or a relative path name (that is, relative to the location of the calling workflow).  This element is optional; it is used only if z/OSMF must create a new instance of the called workflow on the system.  For a relative path, the path must begin with <i>./</i> or <i>../</i> . After this notation, all subsequent instances of <i>./</i> or <i>../</i> in the path will be resolved.  An example of using a relative path is shown in <a href="#">Figure 407 on page 846</a> .	Optional.
<b>calledWorkflowDescription</b>	Specifies a short description of the called workflow. The Workflows task displays this text on the <b>Create Workflow</b> page, when it prompts the user for the workflow definition file.	Required.
<b>calledWorkflowID</b>	The name of the workflow to be called. The combination of <i>calledWorkflowID</i> and <i>calledWorkflowVersion</i> must be unique within the Workflows task.	A selection is required: Either <i>calledWorkflowID</i> or <i>calledWorkflowMD5</i> .

Table 458. Major step elements for defining a called workflow in your workflow definition file (continued)

Element name	Description	Required or optional
<b>calledWorkflowVersion</b>	The version of the definition file that is used to create the called workflow. The combination of calledWorkflowID and calledWorkflowVersion must be unique within the Workflows task.  The Workflows task caches only the latest version of an imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify any portion of the workflow definition file, including changes to any sub-files or referenced files. For this reason, when you create a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.	Optional.
<b>calledWorkflowMD5</b>	An MD5 encrypted value (a 128-bit hash value) that you can use to identify the called workflow.	A selection is required: Either calledWorkflowID or calledWorkflowMD5.

## Identifying the called workflow

To start a called workflow, the calling step must identify which workflow is to be called. The z/OSMF schema provides two methods for you to reference the called workflow. Use either of the following approaches:

- Specify the workflow ID of the called workflow on the workflow ID element (<calledWorkflowID>). You can further qualify this specification by optionally including the version of the workflow definition of the called workflow on the element (<calledWorkflowVersion>). The version is typically updated by the workflow author whenever any portion of the workflow definition file is changed.

The Workflows task caches only the latest version of an imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify the workflow definition. For this reason, when you create a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.

- Specify the called workflow MD5 element (<calledWorkflowMD5>). This element specifies a 128-bit hash value that can be used to identify the called workflow. You can specify this element in place of the workflow ID and version elements.

**Note:** No more than one level of nesting of called workflows is permitted in a workflow-to-workflow relationship. Thus, the specified workflow definition cannot contain a step that calls another workflow.

## Sharing variables between the calling workflow and called workflow

It is possible to share variables between the calling workflow and the called workflow. Any variables that are defined to either workflow can be shared by using the element <variableMapping>.

This element consists of two sub-elements and their associated attributes, as follows:

- Use the element <fromCallingToCalled> to describe the variable values that are to be transferred from the calling workflow to the called workflow.
- Use the element <fromCalledToCalling> to describe the variable values that are to be transferred from the called workflow to the calling workflow. To handle variable conflicts, you can optionally include the attribute `override=` to specify whether the called workflow variables take precedence over the calling workflow variables. The default is `override=false`.

On each element, you can optionally specify the following sub-elements and attributes:

### regExpression

Regular expression. Use this attribute to filter on variable names with one or more wildcard characters. For example, to select all variables prefixed with "setting," you can specify:

```
<regExpression>^setting.*$/regExpression>
```

**variableName**

Name of the variable. Use this element to identify the variable that is to be shared with the target workflow. The variable is also saved in the Workflows task variable pool.

To map this variable to a specific variable in the target workflow, include the attribute `mapTo=` on the element `variableName` and set it to the name of the target variable. The behavior of the attribute `mapTo=` on the element `variableName` depends on which element is used to pass variables, as follows:

- When specified on the element `<fromCallingToCalled>`, the variables are mapped to the target variables only when a new instance of the called workflow is created in response to the calling step.
- When specified on the element `<fromCalledToCalling>`, the variables are mapped to the target variables on completion of the called workflow.

In some cases, you might not know which of your workflow variables are needed by the called workflow. Suppose, for example, that the called workflow is supplied by a different workflow provider. Creating variable mappings in such cases is not possible. As an alternative to creating variable mappings, you can share your instance variables implicitly with a called workflow by making the variables publicly visible. z/OSMF makes all of the calling workflow's publicly visible instance variables available to the called workflow. These variables are referred to as *caller scope* variables. They are shared only with the called workflow. For more information, see [“Caller scope variables” on page 870](#).

**Sharing the account information and JOB statement with the called workflow**

If the calling workflow contains account information and a JOB statement, this information is propagated to the called workflow. On creation, the called workflow can use the passed information if it submits a job to run. To receive the JCL information, the called workflow must have a workflow scope of NONE.

The passed information is:

**accountInfo**

Account information to use in the JCL JOB statement.

**jobStatement**

JOB statement JCL that is used in the job.

If this JCL information is present in the calling workflow, z/OSMF stores it when the called workflow is created. If the called workflow submits a job to run, z/OSMF applies the account information and JOB statement from the calling workflow to the job that is submitted by the called workflow.

If no account information and JOB statement can be obtained from the calling workflow, the called workflow uses the default JOB statement that is supplied with z/OSMF:

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,  
MSGLEVEL=(1,1),REGION=0M,NOTIFY=IBMUSER
```

**Example of how a called workflow is defined in a step**

As an example, assume that your workflow includes a step ("Define User"), which is used to define a user ID and security group to the system security product. Usually, to verify that this setup is done correctly, users would run another workflow. In this example, you add a step to start the other workflow directly as a called workflow. When the step owner selects this step to be performed, the Workflows task displays the called workflow so that the step owner can perform it.

Further assume that a number of variables are shared between the workflows by using the element `<variableMapping>`. The step that calls the workflow (the calling step) passes a number of variables on the element `<fromCallingToCalled>`. Similarly, the called workflow passes a number of variables to the called workflow, on the element `<fromCalledToCalling>`.

The definition for a called workflow might be coded as shown in [Figure 407 on page 846](#).

For illustrative purposes, the example in [Figure 407 on page 846](#) shows various methods for sharing variables between workflows, as follows:

- On the element `<fromCallingToCalled>`, the step that calls the workflow (the calling step) passes variables in the following ways:
  - To pass all variables with "setting" as the variable name prefix, the calling step specifies the element `<regExpression>`, as follows:

```
<regExpression>^setting.*$/regExpression>
```

- To pass the value of the variable that is called `st_user` to the called workflow, the calling step specifies the variable, as follows:

```
<variableName>st_user</variableName>
```

- If an instance of the called workflow is not already created, z/OSMF creates one in response to the called workflow. If so, the element `<fromCallingToCalled>` ensures that the new called workflow inherits the variables from the calling workflow. In this example, the value of the variable `st_uid` is passed to the calling workflow, and overlays the existing value of the variable that is named `st_group` because the attribute `mapTo=` is included on the element `<variableName>`, as follows:

```
<variableName mapTo="st_group">st_uid</variableName>
```

- On the element `<fromCalledToCalling>`, the called workflow shares a number of variables with the calling workflow. Any variables to be copied back to the calling workflow are performed on completion of the called workflow. Here, the override attribute is included so that the called workflow's variables override those of the calling workflow:

- To pass all variables with "set" as the variable name prefix, the element `<regExpression>` is specified, as follows:

```
<regExpression>^set.*$/regExpression>
```

- To pass the value of variable that is called `st_uid` to the calling workflow, and overlay its existing value for the variable that is named `st_gid`, the `mapTo=` attribute is included on the `variableName` element, as follows:

```
<variableName mapTo = "st_gid">st_uid</variableName>
```

- To pass the variable that is called `st_user` to the calling workflow, the variable is specified, as follows:

```
<variableName>st_user</variableName>
```

In [Figure 407 on page 846](#), a relative path is provided for the location of the called workflow definition file on the element `<calledWorkflowDefinitionFile>`. By using a relative path instead of an absolute path, you provide the location of the file in relation to the calling workflow. To be found during processing, the file must reside in the same file system as the calling workflow.

In the example, the location of the called workflow is specified as:

```
<calledWorkflowDefinitionFile>..\samples\workflow_sample_wizards.xml</calledWorkflowDefinitionFile>
```

If the calling workflow is located in the `\usr\lpp\zosmf` directory, the called workflow location resolves to `\usr\lpp\zosmf\samples\workflow_sample_wizards.xml`.

```

<step name="Define User">
  <title>Ensure that the user ID and group are created.</title>
  <description>
    This step verifies that the required user ID and security
    group are created. This step invokes another workflow (a called workflow),
    which is identified here based on the workflow ID and version.
    Alternatively, we could have identified the called workflow using its
    MD5 hash value.
  </description>

  <variableMapping>
    <!-- Variables to share with the called workflow. -->
    <fromCallingToCalled>
      <!-- Use a regular expression to filter the variables. -->
      <regExpression>^settin.*$/regExpression>

      <!-- The following line copies the value of st_uid to the variable st_group. -->
      <variableName mapTo = "st_group">st_uid</variableName>

      <!-- The following line copies the value of st_user to the called workflow,
      if no st_user variable definition already exists in the called
      workflow.
      This value also will be saved in the Workflows task variable pool.-->
      <variableName>st_user</variableName>
    </fromCallingToCalled>

    <!-- Variables to share with the calling workflow. Here, the override attribute
    is set to true, so that the called workflow's variable values will override
    those of the calling workflow. -->
    <fromCalledToCalling override= "true">
      <regExpression>^set.*$/regExpression>
      <variableName mapTo = "st_gid">st_uid</variableName>
      <variableName>st_user</variableName>
    </fromCalledToCalling>
  </variableMapping>

  <callingStepWeight>10</callingStepWeight>
  <callingStepSkills>System Programmer</callingStepSkills>

  <calledWorkflowDefinitionFile>.\..\samples\workflow_sample_wizards.xml
  </calledWorkflowDefinitionFile>

  <calledWorkflowDescription>This called workflow is used to help verify that the user
  and group are created successfully.</calledWorkflowDescription>

  <calledWorkflowID>workflow.sample.wizards</calledWorkflowID>

  <calledWorkflowVersion>1.0</calledWorkflowVersion>
</step>

```

Figure 407. Example: Defining a called workflow on the step element tag.

## Automated steps

A workflow might have steps that can be performed without the need for user interaction, such as a job that can be submitted without user input. If so, you can designate the step as an *automated step* in the workflow definition file. Doing so instructs the Workflows task to run the step automatically, as soon as any prerequisite steps in the workflow are completed. By including one or more automated steps in a workflow, you help to simplify the user experience.

In the Workflows task user interface, automated steps are indicated to users in the following ways:

- In the *Workflow Steps* table, the column Automated indicates whether a step is automated, based on how the step is defined in the workflow definition.
- In the Workflows task main page, when an automated step is performed, the workflow status is indicted as *Automation in Progress*.

A workflow can consist of both automated steps and non-automated (manually performed) steps.

## How the user interacts with an automated step

To perform an automated step, the user selects the **Perform** action in the *Workflow Steps* table, as is done for manual steps. For an automated step, the Workflows task presents the user with a dialog window to confirm whether the step and any subsequent automated steps are performed automatically. Alternatively, the user can choose to perform the step manually, by selecting an option in the dialog window that is called **Manually perform the selected step**.

When started, an automated step—or series of automated steps—can run to completion or until stopped by another condition, such as a user stop request or a step error. Specifically, a workflow with automated steps can run until one of the following conditions occurs:

- Completion of all subsequent steps.
- Processing reaches an automated step for which one or more required variables are not satisfied.
- Processing reaches a non-automated step in the sequence of steps.
- Processing reaches an automated step that is not currently eligible for automatic processing. That is, the step is *Unassigned*, *Assigned*, *Not Ready*, or *Submitted*.
- Processing is stopped through a user request.
- An error is encountered.

A workflow that is comprised entirely of automated steps can run to completion without user intervention.

## Tracking the progress of automated steps

For workflows that contain automated steps, z/OSMF creates notifications and history entries to inform step owners of the automation progress.

At the completion of an automated step or a sequence of automated steps, z/OSMF creates a notification to inform the step owner of the step status. If processing reaches a manual step that requires user interaction before the workflow can continue, z/OSMF creates a notification for the step owner to prompt for action. Similarly, if an automated step is stopped or fails for any reason, z/OSMF sends a notification to the step owner. In z/OSMF, users can access notifications through the Notifications task.

During the processing of an automated step, z/OSMF updates the workflow history to indicate the key checkpoints in the workflow progress, such as:

- Completion of the automated step
- Completion of all automated steps in the workflow
- Automation is started or stopped through user request (and by whom)
- An error is encountered during the processing of an automated step.

Automation progress is not displayed (in terms of step completion check marks) until the user refreshes the display.

Users of the Workflows task can view the details of the step status in the Workflow History table.

## Design considerations for automated steps

Consider a step to be eligible for automation if it requires no user input at all. Or, if all of the required inputs can be supplied to the workflow at creation time, in the form of a workflow variable input file.

When you code the step element, you can specify whether the step is automated (that is, can be performed automatically by the Workflows) by including the `autoEnable` attribute on the step element (`<step>`). Set this attribute to `true` or `false`, as needed. By default, the `autoEnable` attribute is `false`. [Figure 408 on page 848](#) shows an example.

```

<step name="Sample_Automated_Step" optional="true">
<title>This is a very simple JCL job</title>
<description>This optional step submits an empty job using IEFBR14.</description>
<instructions>This step is performed automatically.</instructions>
<weight>1</weight>
<skills>Submit a job to run on z/OS</skills>
<autoEnable>true</autoEnable>
<template>
  <inlineTemplate>//STEP1      EXEC  PGM=IEFBR14
//SYSEXEC  DD  DUMMY
//* PRINT DD SYSOUT=A
/*
    </inlineTemplate>
    <submitAs>JCL</submitAs>
  </template>
</step>

```

*Figure 408. You can designate a step as automated by adding the `autoEnable` element to the `<step>` element tag.*

For any automated steps that you provide, it is recommended that you use the description tag to provide the user with enough information to understand the implications of allowing the step to run.

When automated steps are ordered consecutively in a workflow, a request to run the first automated step begins a process in which each subsequent automated step can run to completion, or until one of the steps encounters a condition that stops the processing of steps. For this reason, it is recommended that you group automated steps in the workflow definition file together to take full advantage of this cascading behavior.

If you do not rely on automated steps to complete in a certain sequence, consider using parallel processing for a potentially faster completion time for your workflow. For details, see [“Enabling automated steps for parallel processing”](#) on page 849.

## Using the suspend element to control automation

As the workflow author, you can force automation to stop before a particular step is run. You might do so, for example, to pause automation so that someone can perform an action outside of the workflow before the workflow resumes automated processing.

To force a stop in the sequence of automated steps, include the suspend element (`<suspend>`) in the step definition. When automation processing reaches a step that includes the suspend element, automation stops at the step. Here, z/OSMF can send an email notification to one or more recipients that you specify, such as a person who should be prompted to take action. You might, for example, need to direct someone to configure a resource, or collect some information that the workflow owner needs to proceed with the workflow. Or, you might simply need to notify someone that a created object is available for use. The properties of the email are described in [Table 459 on page 849](#).

To resume the workflow, a user can choose to manually run the suspended step, or specify that automation is to resume from the step that contains the suspend element. In the latter case, automation ignores the suspend element and begins automation from the selected step.

On the suspend element, you can specify the settings to create the notification email. [Table 459 on page 849](#) describes these sub-elements of the suspend element.

Table 459. Information to specify for notifying a user about a suspended step.

Element	Description
<b>toRecipients</b>	<p>Specify the email addresses of the persons to be notified of the suspended step. To specify more than one recipient, enter each address, separated by commas or spaces.</p> <p>It is possible to use variable substitution in this field.</p> <p>In addition to any users that you specify on the &lt;toRecipients&gt; element:</p> <ul style="list-style-type: none"> <li>• For a configuration or general workflow, z/OSMF always sends a notification to the workflow owner.</li> <li>• For a provisioning workflow, z/OSMF sends a notification to the domain administrator, if this user identity is defined. Otherwise, z/OSMF notifies the workflow owner.</li> </ul>
<b>subject</b>	<p>Specify a brief, meaningful subject for the notification email. If you omit this value, the email subject is set to no subject by default.</p> <p>It is possible to use variable substitution in this field.</p>
<b>content</b>	<p>Specify the text of the message that you want to send to the recipient. If you omit this value, the email content is set to no content by default.</p> <p>It is possible to use variable substitution in this field.</p>

Figure 409 on page 849 shows an example of how the suspend element can be coded within a step definition.

```
<description>Submit an empty JCL job to JES using IEFBR14.</description>
<suspend>
<toRecipients>neelesh@my.company.com, conner@my.company.com</toRecipients>
<subject>Automation suspended</subject>
<content>These are the instructions to resume automation and contact the
domain administrators</content>
</suspend>
<instructions substitution="false">This is a very simple JCL job submission that is
performed
by using IEFBR14.</instructions>
```

Figure 409. Example of a suspend step

In Figure 409 on page 849, the email notification:

- Is sent to the recipients neelesh@my.company.com and conner@my.company.com
- Has the subject "Automation suspended"
- Contains the following text: "These are the instructions to resume automation and contact the domain administrators."

**Note:** For a suspended step, the *Step Properties* page in the Workflows task includes a tabbed area called **Suspend Information**. This area displays the email settings that are specified on the sub-elements of the suspend element.

## Enabling automated steps for parallel processing

As the workflow author, you can request that automated steps be run in parallel (concurrently), rather than sequentially. A workflow with steps that can be run in parallel is called a *parallel-steps workflow*.

With parallel processing, a workflow can take less time to complete. However, the steps might complete in an unexpected sequence. Consider using parallel processing when you do not rely on automated steps to complete in a certain sequence.

To use parallel processing, include the attribute `parallelSteps=true` in the workflow metadata. Otherwise, if this attribute is set to `false` or omitted, automated steps are run one by one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.

When a parallel-steps workflow is started, the Workflows task locates the automation ready steps and attempts to run them concurrently.

A step is considered to be *automation ready* when it is:

- Enabled for automation. In the workflow definition file, the attribute `autoEnable=true` is specified on the step element (`<step>`).
- In an eligible state: *ready*, *in-progress*, or *failed*. For a failed step, the Workflows task performs the step again.

In a parallel-steps workflow, the failure of an automated step does not stop automation processing for the other automated steps. Processing continues until all of the automated steps are completed or failed, or a condition occurs that stops automation processing, such as a user stopping automation by using the **Stop Automation** action in the Workflows task.

**Note:** The ability to suspend step processing is mutually exclusive with the ability to run steps in parallel. Therefore, if a workflow includes the suspend element (`<suspend>`) in the step definition, it is not eligible for parallel processing. The Workflows task enforces this restriction. An attempt to start a workflow that contains both the suspend element (`<suspend>`) and the attribute `parallelSteps=true` results in an error.

## Making a step conditional

A *conditional step* is available to be performed when a logical condition is satisfied on the z/OS system or in the Workflows task. A conditional step might become *Ready* (eligible to be performed), for example, if a job run by another step ends with a particular return code. A conditional step remains *Not Ready* (unavailable to be performed) as long as the condition is not satisfied.

Understand that a conditional step, which depends on a logical condition, is different than a *dependent* step, which depends on a particular step being completed, to satisfy a prerequisite.

In the Workflows task user interface, conditional steps:

- Are indicated to users in the **Details** tab on the *Step Properties* page.
- Are shown in the *Not Ready* state until the condition is true (satisfied) — even when the prerequisite steps, if any, are complete.

A conditional step becomes ready for performing only when a specific condition is satisfied in the current step or a preceding step. Thus, the expression being tested and a text description are required sub-elements of the condition element.

## Target states

Optionally, you can specify a desired *target state* for a conditional step. The target state specifies the state the step is to assume when the condition is true. Typically, the target state is *Ready*, which is the default value, if you choose to omit this sub-element.

The following target states are valid:

- Ready
- Skipped
- Complete.

A conditional step remains unavailable to be performed as long as the condition is not satisfied. This rule applies even when `prereqStep` or `prereqTargetStateSet` are also defined for the step. Also, a `targetStateSet` with a condition is evaluated only when the condition is true.

## Types of conditional expressions

The following types of conditional expressions are supported:

- Expressions using logical operators AND (&) and OR (|). For example:

```
${step1.returnValue} == "0000" || (${step2.returnValue} == "0000" && ${step2.stepOwner} == "IBMUSER")
```

- Expressions based on ternary operators. For example:

```
condition ? value_if_true : value_if_false
```

- Mathematical functions. For example:

```
Math.max(${step1.returnValue} , ${step2.returnValue} ) > 0
```

## Design considerations for conditional steps

Observe the following design considerations for conditional steps:

- A conditional step must be a leaf step (a step with no substeps). A parent step cannot be a conditional step.
- When coding the step element, specify whether the step is conditional by including the `condition` attribute on the step element (`<step>`). Also, specify both the expression being tested for (typically a mathematical or logical expression) and a text description of the condition. Both the expression and its description are displayed to the end user in the **Details** tab on the *Step Properties* page.
- A conditional expression can refer only to preceding steps in the workflow.
- You can include workflow input variables in conditional expressions. Doing so allows conditional steps to resolve to true or false, based on installation-specific conditions.
- You can use the following step attributes in conditional expressions: `<stepState>` and `<returnValue>`.
- `<returnValue>` is a string type attribute; you cannot use it in a mathematical comparison. To compare a return code with a second return code or another numerical value, such as zero (0), you can write the condition expression like this: `${step2.returnValue} ) > "0000"`. Represent the return code string with four characters, for example "0000" or "0008".

### Example

As an example, assume that Step 3 should not be performed unless Step 1 and Step 2 complete with a return code zero. Here, the XML for Step 3 could be coded as follows:

```

<Step name=Step3">
  <title>A conditional step based on return code</title>
  <description>This conditional step is not ready until
                                the two preceding steps complete with RC 0
  </description>
  <instructions>Run this job.</instructions>
  <condition>
    <expression>${step1.returnCode} == "0000" || (${step2.returnCode} == "0000"
    </expression>
    <description>This step requires that Steps 1 and 2 have
                                completed successfully.
  </description>
</condition>
<targetState>Ready</targetState>

```

Figure 410. You can designate a step as conditional by adding the condition element to the <step> element tag.

The previous example can be expanded to include a condition, based on a variable value. In [Figure 411 on page 852](#), Step 3 is not performed unless Step 1 and Step 2 complete with a return code zero and the variable `${instance-st_user}` is IBMUSER.

```

<Step name=Step3">
  <title>A conditional step based on return code and user ID</title>
  <description>This conditional step is not ready unless
                                the two preceding steps complete with RC 0
                                variable st_user value is IBMUSER
  </description>
  <instructions>Run this job.</instructions>
  <condition>
    <expression><![CDATA[${step1.returnCode} == "0000" &&
    ${step2.returnCode} == "0000" &&
    ${instance-st_user} == "IBMUSER" ]]>
    </expression>
    <description>This conditional step is not ready unless the two
                                preceding steps complete with RC 0 and the variable
                                st_user value is IBMUSER.</description>
  </condition>
  <targetState>Ready</targetState>
  :
</step>

```

Figure 411. You can use variable values in the condition to be satisfied.

Note that a variable reference can contain an underscore, for example: `${instance_st_user} == "IBMUSER"` or a hyphen, for example: `${instance-st_user} == "IBMUSER"`.

## runAsUser identity for a step

For workflow automation, you can specify the user ID under which a particular step is to be performed by including the element `runAsUser` (<runAsUser>) on the step element. The value that you specify on the `runAsUser` element is considered to be the *runAsUser ID* for the step.

When a `runAsUser` is not specified for a step, the step is performed under the step owner user ID.

The `runAsUser` ID element is intended for use with an automated workflow that originates from z/OS Cloud Provisioning. The `runAsUser` ID element is not applicable in other cases; its use can result in an error that prevents the step from being performed. For the intended use cases, see [“How a runAsUser ID is used in workflow automation” on page 853](#).

If you specify the `runAsUser` element, you must ensure that the user ID that you specify is (or will resolve to) a valid z/OS user ID. The user ID can be lower case, upper case, or mixed case.

## How a runAsUser ID is used in workflow automation

During workflow automation, a step is switched to the runAsUser ID for the operations that are shown in Table 460 on page 853.

Table 460. How a runAsUser ID is used in workflow processing		
Step type	Automated operation	How the runAsUser ID is used
Template step	Step submits a JCL job (a batch execution step).	Batch job runs under the runAsUser ID.
Template step	Step runs a program in real time (an immediate execution step).	Program runs under the runAsUser ID.
Template step	Step copies an output file to its own storage.	Read action is performed under the runAsUser ID.
Template step that includes the saveAsDataset element.	Step writes the template contents to a data set.	Save as a data set action is performed under the runAsUser ID.
Template step that includes the saveAsUnixFile element.	Step writes the template contents to a UNIX file.	Save as a UNIX file action is performed under the runAsUser ID.
REST step	Step calls a REST service.	REST service is invoked under the runAsUser ID.

For other step types and operations, the runAsUser ID is not applicable. Specifically, the runAsUser ID is not used for the following step types:

- Instructions steps
- Calling Steps

## Specifying approvers for a step

In z/OS Cloud Provisioning, the runAsUser element is typically used with the approver element, as follows:

- The runAsUser element (<runAsUser>) specifies the user identity under which the step is to be performed.
- The approver element (<approver>) specifies who must grant approval before the step can be performed. Up to 12 approvers can be specified for a step. The use of an approver element requires that the runAsUser element be specified.

Various combinations of approvals are supported for the runAsUser element. In the examples that follow, assume that a workflow includes a step that is performed under an authorized administrator identity, which is represented by the runAsUser ID ADMIN-USERID.

- In the following example, either HIREN or NICK must approve the step:

```
<runAsUser>ADMIN-USERID</runAsUser>
<approver>HIREN NICK</approver>
```

- In the following example, both HIREN and NICK must approve:

```
<runAsUser>ADMIN-USERID</runAsUser>
<approver>HIREN</approver>
<approver>NICK</approver>
```

- In the following example, CONNER and either HIREN or NICK must approve:

```
<runAsUser>ADMIN-USERID</runAsUser>  
<approver>CONNER</approver>  
<approver>HIREN NICK</approver>
```

## Using variable substitution

You can use variables to represent the runAsUser ID and approver ID. To indicate that substitution is used, include the attribute "substitution=true" on the element, and specify the substitution string on the element.

In the following example, the variable ADMIN-USERID is used to represent the runAsUser ID for a step:

```
<runAsUser substitution="true">${instance-ADMIN-USERID}</runAsUser>
```

If you use variable substitution, understand that the variable must be an instance variable; it cannot be a global variable.

In a called workflow, the caller scope variables are eligible for use with user ID substitution. For more information about caller scope variables, see [“Caller scope variables” on page 870](#).

## Static runAsUser IDs

If the runAsUser ID is resolved at workflow creation time, it is considered to be a *static* runAsUser ID. Here, the value is determined during substitution, by using values from the input properties file. Or, it might be set to a fixed value in the workflow definition.

At workflow creation time, z/OSMF checks the resulting value of substitution to ensure that it is a valid z/OS user ID and is permitted to z/OSMF (that is, permitted to the z/OSMF SAF profile prefix profile in the APPL class). If these checks fail, the workflow cannot be created.

As a recommended practice for a static runAsUser, include the variable in the workflow input properties file. Also, avoid using a variable that can be prompted for at workflow creation. For a description of the prompt= attribute for instance variables, see [“Using the element atCreate to qualify a variable definition ” on page 861](#).

## Dynamic runAsUser IDs

If the runAsUser ID is assigned during workflow processing, it is considered to be a *dynamic* runAsUser ID. Suppose, for example, that Step A creates a user ID, which is then used by Step B for performing some action, such as issuing a command.

If a step is associated with a dynamic runAsUser ID, z/OSMF does not validate the runAsUser value at workflow creation time. Instead, the runAsUser value is checked when the step is run. When the step is being performed, the Workflows task processes the variable substitution to derive the actual user IDs for the step.

As a recommended practice for a dynamic runAsUser, use a variable that is not referenced in the workflow input properties file.

## How the static or dynamic determination is made

The determination as to which runAsUser values are dynamic and which are static is made at workflow creation time, as follows:

- If a runAsUser value is defined with substitution=true, it is considered to be dynamic if the initial substitution results in no change to the value.
- If a runAsUser value is defined with substitution=false, or the initial substitution results in a change to the value, the runAsUser value is considered to be static.

Assume that a runAsUser value is represented by the variable reference `${instance-rau}`. Table 461 on page 855 shows how the static or dynamic determination is resolved for `${rau}`, depending on a number of factors, such as whether the variable can be prompted or is pre-specified in the input properties file.

Table 461. Basic usage scenarios: How the static or dynamic determination is made for the runAsUser value						
<runAsUser>	<atCreate prompt=>	<runAsUser substitution=>	Input properties file?	Result of initial substitution	Dynamic or static?	Notes
<code>\${instance-rau}</code>	<i>false</i>	<i>true</i>	Not specified.	<code>\${instance-rau}</code>	Dynamic	Dynamic because there is no change after substitution.
<code>\${instance-rau}</code>	<i>true</i>	<i>true</i>	Not specified.	<code>\${instance-rau}</code>	Dynamic	Dynamic because there is no change after substitution.
<code>\${instance-rau}</code>	<i>false</i>	<i>true</i>	<code>rau=IBMUSER</code>	IBMUSER	Static	Static because the value changed after substitution.
<code>\${instance-rau}</code>	<i>true</i> <sup>1</sup>	<i>true</i>	<code>rau=IBMUSER</code>	<code>\${instance-rau}</code>	Dynamic	Dynamic because there is no change after substitution.
1. Workflows processing ignores the prompt=true setting for a runAsUser when it resolves the substitution string. Here, the runAsUser value is always determined to be dynamic.						

## Using translatable strings

The tables in “Workflow XML reference” on page 874 indicate which elements have translatable values with a type of `nlsString` or `nlsRichString`.

A translatable string takes two optional attributes, `bundle=` and `bundleKey=`. If one attribute is specified, the other must also be specified.

Table 462 on page 855 describes shows the `bundle=` and `bundleKey=` attributes.

Table 462. Translatable strings			
Attribute name	Description	Type	Requirements and restrictions
<b>bundle=</b>	The name of a bundle defined in the message manifest	A single-token string	The referenced bundle must exist.
<b>bundleKey=</b>	The key within a language file containing the replacement text for the text element.	<code>nonNullString</code>	The referenced file should contain a key of this name, but this is not validated by the Workflows task.

## Using rich translatable strings

Within the schema, any string defined with type `nlsRichString` or `nlsRichVelocityString` is a translatable string that can contain HTML tags. Not all HTML tags (and their attributes) are supported, though tags for headings, tables, lists, hyperlinks, and text formatting are available.

The allowable tags are: `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `ol`, `ul`, `dl`, `dt`, `dd`, `li`, `br`, `p`, `hr`, `table`, `th`, `td` (with the `frame`, `rules`, and `width` attributes), `tr`, `caption`, `colgroup`, `col`, `thead`, `tbody`, `tfoot`, `i`, `b`, `u`, `em`, `strong`, `cite`, `code`, `samp`, `kbd`, `pre`, `tt`, `sub`, `sup`, `big`, `small`

To specify a hyperlink, use the anchor (`<a>`) tag. When clicked, the hyperlink opens a new tab or window, based on the user's browser settings. You can specify the `href` attribute only. To include an ampersand character (`&`) in the URL, enclose the symbol in quotes: `"&"`. Also, include the protocol with the URL. For example, a value of `"http://www.ibm.com"` is correct, but `"www.ibm.com"` is not.

## Defining variables for your workflow

This topic describes the elements and types that make up a variable definition. Variables can be referenced by workflow steps for substitution in step instructions and templates, and for calls to REST interfaces. A workflow can contain up to 1500 variable definitions.

This topic includes the following information:

- [“Using Velocity templates for variable substitution and other functions” on page 856](#)
- [“Specifying the variable element and its attributes” on page 858](#)
- [“Sub-elements of the variable element” on page 859](#)
- [“Using the element `atCreate` to qualify a variable definition ” on page 861](#)
- [“How to refer to a variable” on page 863](#)
- [“Workflow internal variables ” on page 864](#)
- [“System variables” on page 866](#)
- [“Array variables” on page 869](#)
- [“Caller scope variables” on page 870](#)
- [“Providing a workflow variable input file” on page 871](#)

The elements and attributes that are used to define variables are listed in [Table 475 on page 914](#) and [Table 476 on page 917](#).

## Using Velocity templates for variable substitution and other functions

By using variables, you can add significant function to your workflow. With variables, you can design steps to prompt the user for input before performing particular functions, such as running a job. Further, you can use variables in conditional expressions, for added flexibility in your design.

In z/OSMF, the open source Apache Velocity Engine is used for performing variable substitution and conditional directives. You can use variables for simple string replacement, and also for creating conditional directives that allow you to generate different strings, based on the presence or value of any variable that is referenced by the step. The type of the variable, as defined in the XML, is passed in to the Velocity Engine so that the expected behavior is preserved, except for time and date, which are passed in as strings.

You define a variable by coding `$instance-variable-name`. The prefix *instance* means that the scope of the variable is within the current instance of the workflow. Another prefix, *global*, is used to define a variable that can be used by all workflows in the system.

The following example shows a workflow step that does not use variables.

```
<step name="Step1" >
  <title>
    Define the started task user ID to SAF.
  </title>
```

```

    <description>
        Define the started task user ID to SAF.
    </description>
    <instructions>
        You must define the user ID to your security product.
        For example, for RACF:<br/>
            ADDUSER STASK OMVS(UID(18136) HOME(/u/stask))
        <br/><br/>
        After you have entered this command from the TSO command line,
        press <strong>Finish</strong> to complete the step.
    </instructions>
    <weight>2</weight>
    <skills>Security administration</skills>
</step>

```

Suppose that you want to prompt the user for input, which might then be substituted in the command that is contained in the instructions. To do so, you can modify this step to include a variable. In the example that follows, the reference to UID 18136 in the previous example is replaced with a variable that is used to prompt the user for a UID.

```

<step name="Step1" >
    <title>
        Define the started task user ID to SAF.
    </title>
    <description>
        Define the started task user ID to SAF. You will be
        prompted for the UNIX UID to assign to the user.
    </description>
    <variableValue name="uid" required="true"/>
    <instructions substitution="true">
        You must define the user ID to your security product.
        For example, for RACF:<br/>
            ADDUSER STASK OMVS(UID($instance-uid) HOME(/u/stask))
        <br/><br/>
        After you have entered this command from the TSO command line,
        press <strong>Finish</strong> to complete the step.
    </instructions>
    <weight>2</weight>
    <skills>Security administration</skills>
</step>

```

In the example, observe the following considerations:

- User input is defined by using the `variableValue` element. In this example, the variable is named `uid`.
- Substitution is performed by using a variable reference, which is `$instance-uid` in this example.

A variable reference follows this format:

- Dollar sign (\$)
- Scope, which is either `instance` or `global`
- Hyphen (-)
- Variable name, for example, `uid`.

For more examples of how to code symbolic variable references within instructions and templates, see file `workflow_sample_wizards.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

#### Note:

- When you are using Velocity comparison operators in the instructions, do not use the less than ("`<`") and greater than ("`>`") characters, as they interfere with XML. Instead, use the alternative notation: `lt`, `le`, `gt`, and `ge`.
- When you are doing string substitution, understand that the number sign ('`#`') is a reserved character for the Velocity Engine. Avoid using the number sign in string substitution, as it can lead to unexpected results when the variable is resolved.

If you need to use the number sign character in string substitution, use this technique:

1. Create a variable such as `numberSign` and assign it the value `"#"`

2. Replace all of the "#" characters with the variable reference "\${instance-numberSign}".
- White space (newlines, tabs, spaces) is collapsed before text is displayed in the Workflows task. The indenting that is shown in this example is included for readability only. To obtain the required spacing for your workflow in the Workflows task, you must provide the appropriate HTML formatting tags. You might need to experiment with the spacing somewhat.

For more information about the Velocity Engine, see the following website: <http://velocity.apache.org>.

## Specifying the variable element and its attributes

A workflow variable is defined on the variable (<variable>) element.

The elements and attributes that are used to define variables are listed in [Table 475 on page 914](#) and [Table 476 on page 917](#).

The following attributes are supported for the variable (<variable>) element:

### name

Name of the variable. The variable name is required, and must be a string that consists of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. The variable name must begin with a letter.

The combination of variable name and variable scope must be unique within the workflow.

When you are choosing a variable name, be aware that the following string values are reserved; they cannot be used as the variable name:

- workflowKey
- metaAppVersion
- releaseVersion
- String value that ends with --lastSetStep
- String value that ends with --lastSetTime.

### scope

Scope of the variable, as follows:

#### instance

Variable is used only within the workflow in which it is defined. If multiple workflows are created from the same workflow definition file, each has its own set of instance scoped variables.

#### global

Variable can be referenced by any workflow that is imported into the Workflows task. Global variables are shared across all workflows, even workflows that are created from different workflow definitions. As an example, you might use a global variable to refer to a product-specific constant across a number of workflows that are associated with the product.

The scope is required. The default is *instance*.

The combination of name and scope must be unique within the workflow.

You cannot use the same name for both an instance variable and a global variable in the same workflow definition.

Use global variables with caution to avoid possible naming conflicts across unrelated workflows. Consider your naming conventions carefully and avoid using unspecific variable names. Similarly, consider qualifying your variables, for example, with the 3-character prefix associated with your software product, or a unique identifier.

Be aware that variables are case-sensitive. For example, "Variable1" is not the same as "variable1."

### Notes:

- Global variables are deprecated, as of z/OS V2R3. IBM recommends that you use instance variables or system variables, instead. Global variables might not be supported in a future release. For information about creating system variables, see “z/OSMF system variable services” on page 709.
- z/OSMF includes a number of “built-in” workflow variables, which might provide the function that you require; see “Workflow internal variables ” on page 864.

### visibility

Specifies whether the variable is intended for public or private use. This attribute is intended for the workflow author’s use. The visibility setting does not affect how the variable is processed by the Workflows task. This attribute is optional; the default is *private*.

## Example of a variable definition

In the example in [Figure 412 on page 859](#), the variable `variable_test` is defined.

```
<variable name="variable_test" scope="instance" visibility="private">
  <label>Variable 4</label>
  <abstract>Abstract for Variable 4.</abstract>
  <description>Description for Variable 4.</description>
  <category>variables</category>
  <string/>
</variable>
```

*Figure 412. Specifying attributes on the variable element*

## Sub-elements of the variable element

This topic describes the sub-elements and types that make up a variable definition.

The `<variable>` element can contain the following sub-elements:

- label (required)
- abstract (required)
- description (required)
- exposeToUser (optional)
- category (required)
- datastore (optional)

Variables require a label (`<label>`) and an abstract (`<abstract>`). These values are displayed in the Workflows task when it prompts for input. In addition, the Workflows task displays a description (`<description>`) for the variable if the user clicks the information icon for the abstract.

You can specify a category (`<category>`) for a variable to assign it to a logical group of related variables. For a given step, all variables with the same category are displayed on the same web page. When viewed through the Workflows task, the workflows Step Perform wizard proceeds through each of the categories that you define for the step. In this way, you can logically organize many variables to provide context and an easier user experience for users who enter variable values.

## Variable definition type-specific elements

A variable definition includes a type-specific element that contains elements and attributes specific to that type. The supported types and corresponding element names are described as follows:

### boolean

The Workflows task displays a simple check box to prompt the user for this variable. Specify a default value of `true` or `false` to indicate whether the check box is initially displayed to the user as checked. If you do not provide a value, `true` is used by default.

## string

The Workflows task displays the variable in a text box, initially primed with an optional default value, if you specify one. Use the "multiLine=" attribute (Boolean) of the string element (<string>) to specify whether the text box is small or large.

You can specify a number of choices for the variable. If so, the Workflows task displays the text box with a menu from which the user can select a value for the variable. The "valueMustBeChoice=" attribute (Boolean) of the string element specifies whether the user must choose from the predefined values or can enter a custom value.

You can specify more validation criteria for the variable in one or more of the following ways:

- Minimum length (<minLength>) or maximum length (<maxLength>), or both, of the string value
- Predefined validation type (<validationType>) to be provided by the Workflows task. You can request validation for common constructs, such as data set names, data set qualifiers, z/OS user IDs. For a list of available validation types, see [“Variable definition elements and types summary” on page 913](#).
- Regular expression (<regularExpression>) that you provide when neither of the other mechanisms meet your requirements. The regular expression must adhere to the JavaScript standard; see the document posted at <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>.

The criteria that you specify are enforced by the Workflows task in real time as the user types the input value. If the user specifies incorrect syntax, for example, the Workflows task displays the variable in red, along with a default error message, which you can override by providing the error message element (<errorMessage>). Any default or choice value that you specify in the variable definition is also subject to these criteria.

The Workflows task cannot load an XML file that violates the criteria. Similarly, you cannot define the "valueMustBeChoice=" attribute with a value of true without defining any choices.

## integer

The Workflows task displays the variable in a text box, initially set to a value that you can optionally specify. The Workflows task restricts the user's input value to a signed 31-bit value (in the range of -2147483648 to 2147483647). You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the integer. The Workflows task validates the default value against the minimum and maximum when the workflow definition file is imported into z/OSMF.

## decimal

The Workflows task displays the variable in a text box, initially set to a value that you can optionally specify. A decimal is an integer with a "decimalPlaces=" attribute on the <decimal> element. A decimal value allows the same whole number value as an integer, plus up to six decimal places (that is, a value in the range of -2147483648.999999 to 2147483647.999999). The "decimalPlaces=" attribute has a default value of 1. You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the decimal. The Workflows task validates the default value against the minimum and maximum when the workflow definition file is imported into z/OSMF.

## time

The Workflows task displays the variable in a timebox. By default, the time is displayed in 15-minute increments, based on your specified default, or the current time, if you do not specify a default. The user can type in a value, also. You can optionally specify a minimum value (<minValue>) and maximum value (<maxValue>) for the time. The Workflows task restricts the user input to the range you specify.

### Note:

- You specify this variable in hours-minutes-seconds format (hh:mm:ss), but the timebox displays the time in a slightly different format.
- The schema allows slight variations of this format, but the Workflows task does not. Using a time format other than hh:mm:ss can have an unpredictable result.

## date

The Workflows task displays the variable in a calendar, for which the date is based on your specified value, or the current date, if you do not specify a value. The user can type in a value, also. You can optionally specify a minimum value (`<minValue>`) and maximum value (`<maxValue>`) for the date. The Workflows task restricts the user input to the range you specify.

### Note:

- You specify this variable in year-month-day format (yyyy-mm-dd), but the calendar displays the date in a slightly different format.
- The schema allows slight variations of this format, but the Workflows task does not. Using a date format other than yyyy-mm-dd might have an unpredictable result.

## password

By defining a password variable, you can add a password prompt to your workflow. If you do so, the user is prompted to provide a password on the **Input Variables** tab of the Workflows task. In the user interface, the password variable is displayed as an input field. The input field replaces the user's typed characters with masking characters, such as asterisks ('\*\*\*\*\*'). The password is not shown as it is typed.

You must specify either of the following types of validation checking for a password variable:

- Minimum length (`<minLength>`) or maximum length (`<maxLength>`), or both, of the password value.
- Match with a regular expression (`<regularExpression>`). The expression must adhere to the JavaScript standard; see the document posted at <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>.

The criteria that you specify are enforced by the Workflows task in real time as the user types the input value. If the user specifies an incorrect syntax, for example, the Workflows task displays the variable in red, along with a default error message, which you can override by providing the error message element (`<errorMessage>`).

The password variable type has no default value.

## array

When you need to map a list of values or name-value pairs, use an array variable. The format of an array variable can be either a list of individual values (an *array list*) or a set of one or more name-value pairs (a *JSON array*).

Unlike other types of variables, an array variable cannot be set manually by the user from the Workflows task user interface (UI). Instead, an array variable must be set by using a workflow variable input file, or by using an output file in a workflow step (an inline template step or file template step).

An array variable has no default value.

For more information, see [“Array variables” on page 869](#).

## Using the element `atCreate` to qualify a variable definition

This topic describes the element `atCreate` (`<atCreate>`). For users of the Create Workflow REST service, the `atCreate` element provides additional options for working with variables.

The Create Workflow REST service is described in [“Create a workflow” on page 728](#).

The following attributes can be specified on the `atCreate` element:

### name

Specifies the variable for which the variable attributes are being set. The name is required. For example, to set a variable named `var1`, define the `atCreate` element with the name `var1`.

### scope

Specifies the scope of the variable. This value is set to *instance* (the only valid value) or is omitted; the default is *instance*.

## required

This attribute has a specialized purpose. For a workflow that is created through the Create Workflow REST service, this attribute indicates whether the variable must be set to a value at the time of workflow creation. A variable can be set in any of the following ways:

- Defining a default value for the variable in the workflow definition
- Setting a value for the variable in the workflow variable input file
- Specifying a value for the variable on a Create Workflow service.

This attribute is optional; the default is *false*. If a variable is marked as "required," but the variable is not given a value, an attempt to create the workflow through the Create Workflow REST service will fail with an error.

For a workflow that is created through the Workflows task user interface, this option is ignored. That is, the workflow is created, regardless of the setting of the required attribute.

**Note:** This setting is returned as the "requiredAtCreate" property of a variable by the Retrieve Workflow Definition service; See [“Retrieve a workflow definition” on page 770](#).

## prompt

For users of the Create Workflow REST service, this attribute identifies a variable that *should* be prompted for by the program that issues the REST service. By itself, the prompt attribute does not enforce any behavior for the workflow creation. However, by setting prompt to *true*, you can indicate that prompting is recommended for the variable. The user of the Create Workflow REST service can query the value of the prompt attribute for any variables in the workflow to determine whether any variables should be prompted for.

This attribute is optional; the default is *false*.

**Note:** This setting is returned as the "promptAtCreate" property of a variable by the Retrieve Workflow Definition service; See [“Retrieve a workflow definition” on page 770](#).

You can specify the atCreate element for any instance variables that are used in a workflow definition. The atCreate element is not valid for global variables.

## Example of using the element atCreate

Suppose that you have a variable that would be useful to include in a number of different workflows. If so, you can define the variable in an XML file (an XML fragment) and include the same fragment in the appropriate workflow definitions. The variable is now shared between these workflows.

In Figure 413 on page 862, the variable `variable_test` is defined in an XML file. The variable is used by more than one workflow, so the variable definition is coded in an XML file that can be included with multiple workflows.

```
<variable name="variable_test" scope="instance">
  <label>Variable 4</label>
  <abstract>Abstract for variable 4.</abstract>
  <description>Description for variable 4.</description>
  <category>variables</category>
  <string/>
</variable>
```

Figure 413. Variable definition in this example

Now suppose that the variable's attributes for *required* and *prompt* need to be set differently for different workflows. The `atCreate` (`<atCreate>`) element, which is used with the variable element, allows you to specify different *prompt* and *required* settings for the same variable in different workflow definitions. To do so, have each workflow definition include the XML file that defines the variable. Then, in each workflow definition, specify the `atCreate` element to further clarify the properties of the variable.

In Figure 414 on page 863, the `atCreate` element is used to specify the attributes required and prompt for the variable `variable_test`, which was defined in Figure 413 on page 862. The `atCreate` element can be defined differently in any workflow definition that refers to the variable `variable_test`.

```
<atCreate name="variable_test" required="true" prompt="true"/>
```

Figure 414. How the `atCreate` element is used to specify variable attributes for required and prompt

By including the `atCreate` element in each workflow definition, you can set different values for the *prompt* and *required* attributes for the same variable in different workflows.

## How to refer to a variable

Using braces around variable references is optional, but recommended as a good programming practice. The braces help to ensure that variables are clearly identified in the workflow. Further, the braces prevent ambiguity when it comes time for variable substitution, such as in conditional expressions, and jobs and scripts. For example, the variables `$st_userFRED` and `${st_user}FRED` are evaluated differently by the Workflows task. In the former case, the Workflows task searches for a variable called `st_userFRED`. In the latter case, it is clear that the variable is `st_user`.

A variable reference can contain an underscore or a hyphen. For example, both of the following references are valid: `${instance_st_user} == "IBMUSER"` and `${instance-st_user} == "IBMUSER"`.

Examples of how variables are referenced and used are provided in “Defining steps for your workflow” on page 820. Also, see the file `workflow_sample_variables.xml`, which is supplied with z/OSMF in the `/samples` subdirectory of the product file system.

## Simplified instance variable format in substitution and conditions

If you need to define many instance variables in a workflow definition file, you can save some typing by using the simplified variable format. Here, you can omit the prefix `instance-` from the names of instance variables. To use the simplified variable format, you must enable it by including the optional element `<workflowSettingInfo>`, which is a subelement of the `<workflow>` element.

The `<workflowSettingInfo>` element specifies variables settings for the workflow. It contains two subelements:

### **<variablesSetting>**

To omit the prefix `instance-` from the names of instance variables, include the attribute `isInstanceVariableWithoutPrefix` set to `"true"`. By default, this attribute is `"false"`, which means that instance variables require the prefix.

If you set `isInstanceVariableWithoutPrefix` to `"true"`, you must also ensure that none of the instance variables in the workflow definition are prefixed by `instance-`, either in variable definitions or in conditional expressions.

### **<globalVariableGroup>**

Specifies a global variable group name (on the `name=` attribute) for global variables in the workflow.

Figure 415 on page 863 shows how to specify the use of the simplified variable format.

```
<workflow>
<workflowSettingInfo>
  <variablesSetting isInstanceVariableWithoutPrefix="true"/>
  <globalVariableGroup name="GlobalVarGroup1" />
</workflowSettingInfo>
<workflowInfo>
...
```

Figure 415. Specifying the use of the simplified instance variable format in a workflow definition

Use care with the simplified variable format when you specify variables in substitutions to ensure that the variables are specified consistently. In the following example, the simplified format is used for instance variable references. In the example, Step 3 is not performed unless Step 1 and Step 2 complete with a return code zero and the instance variable `${st_user}` is IBMUSER.

```
<Step name=Step3">
  <title>A conditional step based on return code</title>
  <description>This conditional step is not ready unless
                the two preceding steps complete with RC 0
                variable st_user value is IBMUSER
  </description>
  <instructions>Run this job.</instructions>
  <condition>
    <expression><![CDATA[${step1.returnCode} == "0000" &&
                        ${step2.returnCode} == "0000" &&
                        ${st_user} == "IBMUSER" ]]>
    </expression>
    <description>This conditional step is not ready unless the two
                  preceding steps complete with RC 0 and the variable
                  st_user value is IBMUSER.</description>
  </condition>
  <targetState>Ready</targetState>
  :
  <template>
  <inlineTemplate substitution="true">
  //STEP3 EXEC PGM=IKJEFT01,DYNAMNBR=20
  //SYSTSPRT DD SYSOUT=A
  //SYSTSIN DD *
  ADDGROUP ${st_group} OMVS(GID(${st_gid}))
  /*
  </inlineTemplate>
  <submitAs>JCL</submitAs>
  </template>
</step>
```

Figure 416. You can use variable values in the condition to be satisfied.

**Note:** Though the simplified variable format is supported, the suggested practice is that you use the standard format, for example: "`${instance-varName}`" or "`${global-varName}`".

## Workflow internal variables

z/OSMF includes a number of "built-in" variables, which can be used by workflow authors. These variables are called *workflow internal variables*. You can reference them in your workflow definition without the need for you to define them. This topic lists the workflow internal variables that are available for your use.

Workflow internal variables are separated into two different scopes, as follows:

### Step scope

Internal variables that refer to information about the specific step in which they are referenced. For example, the step title. To refer to these variables in your workflow, use the following syntax: `$_step-variable-name`

### Workflow scope

Internal variables that refer to information about the entire workflow. For example, the workflow name. To refer to these variables in your workflow, use the following syntax: `$_workflow-variable-name`

A reference to a workflow internal variable must include the scope prefix: `_step-` or `_workflow-`. Otherwise, the variable is treated as an undeclared local variable, which results in an error.

Table 463 on page 865 lists the variables that are provided with z/OSMF. Each variable resolves to a string, which is described in the **Description** column. Some internal variables are designed specifically for use with provisioning workflows. Others are applicable to any workflow type.

Table 463. Internal variables: Variables that are provided with z/OSMF.

Variable reference syntax	Variable scope	For use with (workflow type)	Description
<code>\${_workflow-<b>actionID</b>}</code>	Workflow	Provisioning workflows	For an actions workflow, this variable resolves to the action ID for the action object.
<code>\${_workflow-<b>clusterInstanceName</b>}</code>	Workflow	Provisioning workflows	Resolves to the created cluster instance name.
<code>\${_workflow-<b>domainID</b>}</code>	Workflow	Provisioning workflows	Resolves to the ID of the domain that is associated with the template.
<code>\${_workflow-<b>parentRegistryID</b>}</code>	Workflow	Provisioning workflows	Resolves to the ID of the software instance parent registry entry.
<code>\${_workflow-<b>registryID</b>}</code>	Workflow	Provisioning workflows	Resolves to the ID of the software services registry.
<code>\${_workflow-<b>softwareServiceInstanceName</b>}</code>	Workflow	Provisioning workflows	Resolves to the created software service instance name.
<code>\${_workflow-<b>sysplexName</b>}</code>	Workflow	All workflow types	Resolves to the name of the sysplex on which the workflow is running.
<code>\${_workflow-<b>systemName</b>}</code>	Workflow	All workflow types	Resolves to the name of the system on which the workflow is running.
<code>\${_workflow-<b>tenantID</b>}</code>	Workflow	Provisioning workflows	Resolves to the ID of the tenant that is associated with the resource pool.
<code>\${_workflow-<b>templateID</b>}</code>	Workflow	Provisioning workflows	Resolves to the unique identifier for the template.
<code>\${_workflow-<b>templateName</b>}</code>	Workflow	Provisioning workflows	Resolves to the name of the template that is associated with the resource pool.
<code>\${_workflow-<b>workflowName</b>}</code>	Workflow	All workflow types	Resolves to the descriptive name for the workflow.
<code>\${_workflow-<b>workflowKey</b>}</code>	Workflow	All workflow types	Resolves to the workflow key, which is a string value, generated by z/OSMF to uniquely identify the workflow instance.
<code>\${_workflow-<b>workflowOwner</b>}</code>	Workflow	All workflow types	Resolves to the user ID of the workflow owner.

Table 463. Internal variables: Variables that are provided with z/OSMF. (continued)

Variable reference syntax	Variable scope	For use with (workflow type)	Description
<code>\${_workflow-workflowOwnerUpper}</code>	Workflow	All workflow types	Resolves to the user ID of the workflow owner (in uppercase letters).
<code>\${_workflow-workflowSystem}</code>	Workflow	All workflow types	Resolves to the unique name that is assigned to the system definition (the system nick name).
<code>\${_step-runAsUser}</code>	Step	All workflow types	Resolves to the user ID under which the workflow step is to be performed (a runAsUser ID).
<code>\${_step-runAsUserUpper}</code>	Step	All workflow types	Resolves to the runAsUser ID under which the workflow step is to be performed (in uppercase letters).
<code>\${_step-stepName}</code>	Step	All workflow types	Resolves to the step name for the current step.
<code>\${_step-stepNumber}</code>	Step	All workflow types	Resolves to the step number for the current step. Steps are numbered to indicate the sequence in which steps are to be performed. For example, the first step in a workflow is 1.
<code>\${_step-stepOwner}</code>	Step	All workflow types	Resolves to the user ID of the user to whom the step is assigned.
<code>\${_step-stepOwnerUpper}</code>	Step	All workflow types	Resolves to the user ID of the user to whom the step is assigned (in uppercase letters).
<code>\${_step-stepTitle}</code>	Step	All workflow types	Resolves to the step title for the current step.

## System variables

It is possible to create variables that you can use with all workflow instances. These variables are called *system variables*. Unlike instance variables, system variables are system-wide in scope. You can reference them in your workflow without the need for you to define them in the workflow definition. This topic explains how to create and use system variables.

To create a system variable, you use z/OSMF system variable services, or the z/OSMF Systems task GUI. With these functions, you can create, update, retrieve, and delete z/OSMF system variables. The system variables that you create are stored by z/OSMF in the z/OS system variable pool. Each z/OS system in your enterprise can have a unique set of system variables in its system variables pool.

You can create any number of system variables on a z/OS system. No practical limit exists.

A system variable cannot be used as part of an array variable. For information about array variables, see [“Array variables” on page 869](#).

## How to include system variables in your workflow

The following topics describe ways of including system variables in your workflow:

- “Using direct variable reference” on page 867 shows the use of the `setVariable` function to map a system variable to a workflow variable.
- “Using z/OSMF system variable services” on page 867 shows the use of REST services to drive the creation of a workflow.

### Using direct variable reference

The following steps describe a technique for including a system variable in a workflow.

1. **Create the system variable.** If a desired system variable is not already defined on your system, you can create it. Suppose, for example, that you want to define a variable for JES job class so that you can specify the job class for workflow-initiated jobs. In z/OSMF, open the Systems task. In the **Systems** table, select your system. Then, select **Actions > System Variables** to display a window that you can use to create the job class system variable or any other system variables that you require.
2. **Map the system variable to a variable in your workflow.** In your workflow, include a step that assigns the value of the system variable to a workflow variable. To do so, you can use the `setVariable` function.

In the following example, the `setVariable` function is used to assign the value of the system variable you defined (JOBCLASS) to the workflow variable wf\_JOBCLASS.

```
<step name="AssignSysVariable" optional="false">
<title>AssignSystemVariable</title>
<description>Assign system variable to workflow variable</description>
<instructions substitution="false">Generated instruction text for this step.
Update this field with your own text</instructions>
<weight>1</weight>
<autoEnable>true</autoEnable>
<canMarkAsFailed>false</canMarkAsFailed>
<setVariable name="wf_JOBCLASS" scope="instance" substitution="true">${_sys-JOBCLASS}</setVariable>
</step>
```

Figure 417. Example of how a system variable can be used in a workflow step.

Notice that the format for referring to a system variable is `${_sys-variableName}`. This notation is used to extract the system variable value from the system variable pool. You can include the system variable directly in the workflow by using the `${_sys-variableName}` format, which is similar to how you use `${instance-varName}` to refer to an instance variable defined in the workflow. Depending on the system where workflow runs, the system variable that is associated with that system is used.

### Using z/OSMF system variable services

In contrast to the previous technique, the following method makes use of z/OSMF REST services to accomplish the same goal.

1. **Create system variable.** From a program that you create, issue the "Create or update system variables" REST service to define the system variable to your system. For more information, see “Create or update system variables” on page 710.
2. **Get the system variable.** From your program, issue the "Get system variables" service to obtain the system variables for the desired system. This operation retrieves the system variables from the system variable pool and returns them in a JSON array. Your program should include logic to examine the array for the system variables of interest. For more information, see “Get system variables” on page 712.

In the example in Figure 418 on page 868, the returned array includes a system variable for the JES job class.

```
{ "system-variable-list":
  [
    { "description": "Class",
      "name": "JOBCLASS",
      "value": "X" }
  ]
}
```

Figure 418. Example of the returned object from the Get system variables service.

3. **Map the system variable to a variable in your workflow.** Your program can create the workflow by issuing the "Create workflow" service. In the request object for this service, assign the value of the system variable to the variable array that is passed to the workflow on creation. For more information, see ["Create a workflow"](#) on page 728.

## Example use for a system variable

Figure 419 on page 868 shows how a system variable can be used in the z/OSMF user interface. Here, the system variable JOBCLASS is substituted for CLASS in the customized JOB statement for z/OSMF.

Workflows > Customize JOB Statements

### Customize JOB Statements

Selected JOB statement:

Default JOB Statement

\* JOB statement JCL:

```
//IZUWFJB JOB (ACCTINFO),CLASS=${_sys-JOBCLASS},MSGCLASS=0,
//          MSGLEVEL=(1,1),REGION=0M,NOTIFY=HIREN
```

Undo Changes

Figure 419. Variable substitution for a system variable in the Workflows task customize job statements window.

The resulting JCL substitutes the system variable with CLASS=A when the workflow runs on one system and CLASS=X when it runs on a different system. See [Figure 420 on page 868](#).

```
//IZUWFJB JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
//          MSGLEVEL=(1,1),REGION=0M,NOTIFY=HIREN
/*JOBPARM SYSAFF=Z2
//ALLOC1 EXEC PGM=IEFBR14
//DD1 DD DSN=HIREN.SLC.DEMO,
//      DISP=(NEW,CATLG),
//      UNIT=3390,SPACE=(TRK,(10,10,10)),
//      DCB=(DSORG=PO,RECFM=FB,LRECL=1024)
```

```
//IZUWFJB JOB (ACCTINFO),CLASS=X,MSGCLASS=0,
//          MSGLEVEL=(1,1),REGION=0M,NOTIFY=HIREN
/*JOBPARM SYSAFF=Z3
//ALLOC1 EXEC PGM=IEFBR14
//DD1 DD DSN=HIREN.SLC.DEMO,
//      DISP=(NEW,CATLG),
//      UNIT=3390,SPACE=(TRK,(10,10,10)),
//      DCB=(DSORG=PO,RECFM=FB,LRECL=1024)
```

Figure 420. Results of variable substitution for a system variable in the Workflows task.

## Array variables

When you need to map a list of values or name-value pairs, use an array variable. Array variables provide an alternative to defining multiple variables to represent multiple values.

The format of an array variable can be either a list of individual values (an *array list*) or a set of one or more name-value pairs (a *JSON array*).

Unlike other types of variables, an array variable cannot be modified manually by the user from the Workflows task user interface (UI). Instead, an array variable can be set by using an output file in a workflow step (an inline template step or file template step). Therefore, the Workflows task does not display array variables to the user.

### Notes:

1. An array variable has no default value.
2. A system variable cannot be used as part of an array variable. For information about system variables, see [“System variables” on page 866](#).
3. An excessively large properties file with many thousands of array variables can degrade the performance of the Workflows task.

## Examples of using array variables

Examples:

- This array variable is formatted in a list of individual items:

```
["Z0SV24T", "DB211T"]
```

- This array variable is formatted as a JSON array (name-value pairs):

```
[{"property1": "tt1", "dsName": "TEST.DSNAME.TT1"},  
 {"property1": "tt2", "dsName": "TEST.DSNAME.TT2"},  
 {"property1": "tt3", "dsName": "TEST.DSNAME.TT2"}]
```

- This array variable combines both list and JSON array formats:

```
["Z0SV24T", "DB211T",  
 {"property1": "tt1", "dsName": "TEST.DSNAME.TT1"}]
```

## Performing substitution with array variables

As with other types of variables, substitution with array variables follows the conventions of the velocity template. For more information, see [“Using Velocity templates for variable substitution and other functions” on page 856](#).

In the following example, the array variable `testJsonArrayVariable` is used in a script that is run by a workflow step.

```
#foreach($test in $instance-testJsonArrayVariable)  
#set($test.dsn=$test.dsn + 500)  
$test.dsn  
#end
```

*Figure 421. Example of how an array variable might be used in a workflow step.*

The array variable is treated as a string during substitution in the workflow, as shown in [Figure 422 on page 870](#).

```

<step name="complexCondition"
<title>Install Product ABC</title>
<description>In this step, both prereqTargetStateSet and normal targetStateSet are used together.</
description>.
<prereqStep name="basicStep1"/>
<prereqDescription>If Basic Step 1 is completed, this step becomes ready to perform.</prereqDescription>
<prereqTargetStateSet>
<condition>
  <expression><![CDATA[${instance_testJsonArrayVariable} ==
'[{ "dsn": "TEST.SYS1.LINKLIB", "dstype": "PDS" }]']>
  </expression>
<description>Example of using an array variable</description>
<!-- Target step state (skipped) is specified in this conditional step -->
<targetState>Skipped</targetState>
</condition>

```

Figure 422. Example of how an array variable might be used in a workflow step.

In the example in [Figure 422 on page 870](#), if the value of the variable `$instance-testJsonArrayVariable` is `[{"dsn": "TEST.SYS1.LINKLIB", "dstype": "PDS"}]`, the step is skipped.

## Caller scope variables

When a workflow calls another workflow, the calling workflow's instance variables are implicitly shared with the called workflow. Such variables are known as *caller scope* variables.

A called workflow can reference its caller scope variables by using the following syntax:

```
${_caller-VAR1}
```

This method provides a simple alternative to explicitly mapping the caller's variables in the called workflow, as described in [“Sharing variables between the calling workflow and called workflow” on page 843](#).

To be shared with a called workflow, the instance variables must be visible for public use. On the variable element (`<variable>`), the visibility attribute specifies whether the variable is intended for public or private use. This attribute is optional; the default is `private`. To allow an instance variable to be shared with a called workflow, ensure that the visibility attribute is set to `public`. For example:

```
<variable name="VAR1" scope="instance" visibility="public">
```

Caller scope variables are:

- Public visibility instance variables only.
- Shared only with the called workflow.
- Available only to **scope=none** called workflows. A `scope=none` workflow is created as a new instance whenever it is called.
- Static. A copy of the caller variables is made from the calling workflow when the called workflow is created. These variables are not updated to reflect the variable state in the calling workflow after the called workflow is created.
- Read only. If the called workflow must write to a variable in the calling workflow, use workflow-to-workflow variable mappings instead.
- Eligible for use in substitution. For example:

```
/u/userid/${_caller-FOLDER}/${_caller-FILE}
```

Information about caller scope variables is not returned to callers of the Get Workflow Properties REST API. For information about this API, see [“Get the properties of a workflow” on page 736](#).

## Providing a workflow variable input file

This topic describes the format of the editable properties file that is called the *workflow variable input file*. With this file, you can supply preset values for the variables that you use in your workflow definition file. By including a variable input file with your workflow definition file, you save users from having to manually enter values for some or all of the variables in your workflow.

### How a workflow variable input file is used

A workflow variable input file is an optional properties file that you, the workflow author, can use to pre-specify one or more of the input variables that are defined in the workflow definition. By supplying variable values in this way, you make it possible for the user to create a workflow without having to interactively enter the inputs in the Workflows task Step Perform wizard. The Workflows task treats any variables set through the workflow variable input file the same as if the user entered them manually.

If you provide a workflow variable input file, include it with the other materials that you supply to the workflow user, such as the workflow definition file and the other files that comprise your workflow definition. Ensure that the documentation for your workflow definition makes note of the file name, and provides any related instructions for editing or storing the file on the user's z/OS system. The Workflows task accesses the workflow variable input file under the user's identity, thus, the file must be read-accessible by the user who is creating the workflow.

At workflow creation time, the user imports the workflow variable input file into the Workflows task, along with the workflow definition file. The Workflows task reads in the contents of the file and saves its values for use with the created workflow. The Workflows task uses the variable input file in addition to any global variables that are already defined to Workflows task. Any new variables that are defined with a global scope become available to the other workflow instances on the user's system. After the Workflows task imports the file, the task no longer refers to the file.

### Creating a workflow variable input file

As the workflow author, you can create a workflow variable input file as a text file, by using an editor of your choice. The file must be encoded in either of the following formats: ASCII or IBM-1047 (EBCDIC). Use a file type of .txt or .properties. Do not use Unicode encoding, such as UTF-8, for this file.

In the variable input file, specify the properties (variables and their respective values) as one or more key-value pairs. Valid separator characters are equal signs (=), colons (:), or spaces. [Figure 423 on page 871](#) shows the valid formats for specifying properties in the variable input file.

```
key1 = value1
key2 : value2
key3  value3
```

*Figure 423. Format of a workflow variable input file*

[Figure 424 on page 871](#) shows an example of the contents of a workflow variable input file.

```
Boolean1 = false
String3  = SYS1.LINKLIB
Integer2  35
Decimal2 : 3.3
Time2     03:03:00
Date1     = 2013-11-11
```

*Figure 424. Example of a workflow variable input file*

The example in [Figure 424 on page 871](#) is designed to work with the file `workflow_sample_variables.xml`, which is supplied as a sample with z/OSMF. For a description of this file and other coding examples, see [“Sample XML files for your reference” on page 809](#).

The Workflows task does no syntax checking of the properties that are specified in the workflow variable input file. Therefore, you must ensure that valid values are specified for each of the properties.

Also, observe the following considerations:

- Each property that is specified in the variable input file must correspond to a variable named in the workflow definition file. Otherwise, the Workflows task ignores the property.
- If the variable input file specifies a property that matches a variable that is already defined to the Workflows task as a global variable, the Workflows task detects the conflicting definitions and prompts the user for a selection. See [“Avoid conflicting variable definitions” on page 873](#)

You can provide the workflow variable input file in either a z/OS UNIX file or a z/OS data set. For a z/OS data set, use a sequential data set, a member of a partitioned data set (PDS), or partitioned data set extended (PDSE).

If you create the workflow variable input file on a workstation, it is recommended that you use File Transfer Protocol (FTP) in binary mode to transfer the XML files to a z/OS system. Doing so helps to ensure that the files are encoded properly for use on z/OS.

## Using variable substitution in the workflow variable input file

It is possible to use variable substitution in the workflow variable input file. With this ability, you can define a value once in the file and refer to that value on subsequent lines to use the same value.

To use variable substitution in the workflow variable input file, you must include the following specification in the file:

```
_IZU_VARIABLE_SUBSTITUTION_ON
```

Position this specification before any lines that refer to the variable.

Similarly, you can turn off variable substitution for subsequent lines in the file by including the following specification in the workflow variable input file:

```
_IZU_VARIABLE_SUBSTITUTION_OFF
```

Thereafter, subsequent variables are interpreted as literal values.

If you omit `_IZU_VARIABLE_SUBSTITUTION_ON` from workflow variable input file, the default is that no variable substitution is performed for the file.

[Figure 423 on page 871](#) shows an example of how you can turn on and turn off variable substitution in the same workflow variable input file.

```
base = HelloWorld
_IZU_VARIABLE_SUBSTITUTION_ON
var = ${base}
_IZU_VARIABLE_SUBSTITUTION_OFF
var2 = ${base}
```

*Figure 425. Using variable substitution in a workflow variable input file*

In [Figure 423 on page 871](#), notice that variable substitution is:

- On when the variable `var` is processed. As a result, the variable is replaced by the string `HelloWorld`.
- Off when the variable `var2` is processed. As a result, the variable is replaced by the string `${base}`.

Figure 424 on page 871 shows how the previous example would be processed in the Workflows task Step Perform wizard. In this example, a variable input file that is named `VariableInput.properties` is provided for the workflow. Notice that substitution is performed for the first variable `var`, but not for the second variable `var2`.

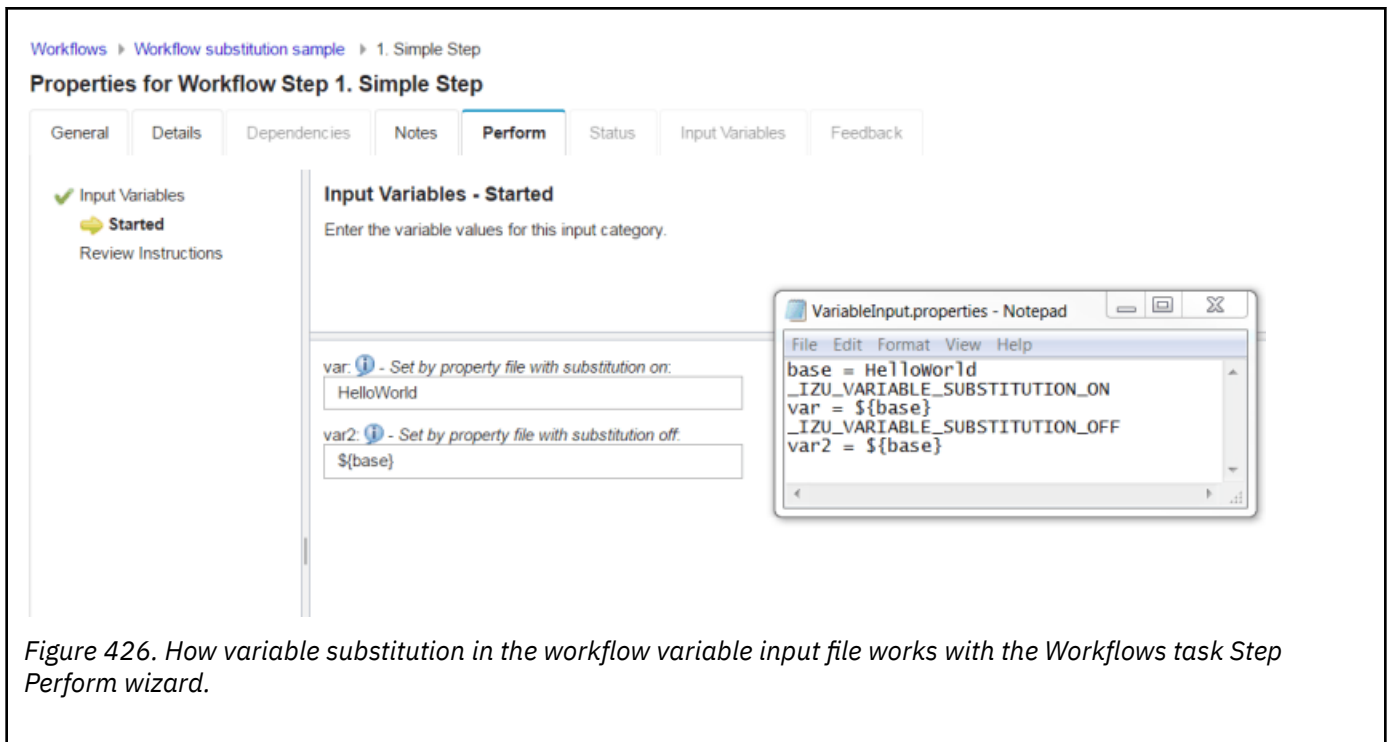


Figure 426. How variable substitution in the workflow variable input file works with the Workflows task Step Perform wizard.

## Avoid conflicting variable definitions

If the Workflows task detects that an imported variable conflicts with an existing global variable, the user is prompted to choose the appropriate value. The user can select to use the input file variable in place of existing global variables, or ignore the input file variable, and use the existing global variable instead. The user's selection determines which version of the variable is saved in the Workflows task global variable pool for use with other workflow instances. Thus, the user's selection affects any other workflows that refer to the same global variable.

It is recommended that you choose unique names for variables to avoid possible naming conflicts with unrelated workflows. Consider your naming conventions carefully and avoid using unspecific variable names. Similarly, consider qualifying your variables, for example, with the three-character prefix associated with your software product, or a unique identifier.

Depending on your design, you might determine that the output file variables must always be used in place of a workflow's existing instance variables. If so, you can include the `needResolveConflicts` attribute (needResolveConflicts) on the output subelement, and set it to false. If so, the Workflows task uses the output file variables in place of any existing values without prompting the user. This setting applies to instance variables only; global variables are not overridden. The default is true; if variable conflicts exist, the user is prompted to resolve the conflicts.

For an array variable, you can use the `load-output-file-array-value` attribute (`loadOutputFileArrayValue`) to manage potential variable conflicts. If set to true (the default), the workflow uses the array variable values from the output file, rather than from the Workflows task. Otherwise, if this attribute is set to false, the workflow uses the existing values from the Workflows task.

## Workflow XML reference

This language reference describes the elements and attributes that comprise a workflow definition.

This topic is organized in tables, with each table describing a major portion of the Workflows schema—the elements and their attributes, default values, the XML attribute data types, and whether a particular attribute is required. [Table 464 on page 874](#) lists the reference tables.

Table 464. Reference tables for the Workflows XML schema		
Element type	Description	Where described
<b>Workflow metadata elements</b>	The elements that make up the workflow metadata	<a href="#">Table 465 on page 876</a>
<b>Sub-elements for a configuration type workflow</b>	For a configuration type workflow, these sub-elements are required	<a href="#">Table 466 on page 880</a>
<b>Sub-elements for a provisioning type workflow</b>	For a provisioning type workflow, these sub-elements are required	<a href="#">Table 467 on page 882</a>
<b>Workflow upgrade elements</b>	The elements that define the workflow upgrade options	<a href="#">Table 468 on page 884</a>
<b>Manifest elements</b>	The elements that make up the manifest	<a href="#">Table 469 on page 886</a>
<b>Step elements: Elements to use for defining all steps</b>	The elements to use for defining all steps	<a href="#">Table 470 on page 888</a>
<b>Step elements: Additional sub-element for parent steps</b>	The step sub-element, which is used to identify the containing step as a parent step	<a href="#">Table 471 on page 890</a>
<b>Step elements: Additional sub-elements for leaf steps</b>	The sub-elements to use for defining a leaf step, which is a step that does not contain step elements	<a href="#">Table 472 on page 892</a>
<b>Step elements: Additional sub-elements for REST steps</b>	The sub-elements to use for defining a step that issues a REST request, such as GET or PUT. This type of step is referred to as a <i>REST step</i> .	<a href="#">Table 473 on page 906</a>
<b>Step elements: Additional sub-elements for steps that invoke another workflow</b>	The sub-elements to use for defining a step that invokes another workflow, which is referred to as the <i>called workflow</i> .	<a href="#">Table 474 on page 910</a>
<b>Variable definition elements</b>	The elements that make up a variable definition.	<a href="#">Table 475 on page 914</a>
<b>Variable definition type-specific sub-elements</b>	The type-specific sub-elements that make up a variable definition.	<a href="#">Table 476 on page 917</a>
<b>atCreate element</b>	For users of the Create Workflow REST service, the atCreate element provides additional options for working with variables.	<a href="#">Table 477 on page 921</a>

In the tables that follow, the elements are listed in the order in which they are required by the schema. Though you can omit optional elements, the elements that you specify must follow the order in which the elements are presented. In contrast, the attributes within an element can be specified in any order.

**Note:** The tables in this language reference are formatted in landscape view to improve usability when you print copies of these pages. To adjust the view in Adobe Reader, select **View > Rotate View > Clockwise**.

## **Workflow metadata elements summary**

Table 465. Workflow metadata elements				
Element name	Description	Required or optional	Type	Supported attributes
workflowSettingInfo	Specifies variables settings for the workflow.	Optional.	Contains the following sub-elements: <ul style="list-style-type: none"> <li>variablesSetting (optional)</li> <li>globalVariableGroup (optional)</li> </ul>	<p>The following attribute is supported for the <i>variablesSetting</i> element:</p> <p><b>isInstanceVariableWithoutPrefix</b> Indicates whether the simplified format is used for variable name references. If set to "true," the prefix instance- must be omitted from references to instance variables, such as in variable definitions and in conditional expressions.</p> <p>The following attribute is supported for the <i>globalVariableGroup</i> element:</p> <p><b>name</b> Specifies the global variable group name for global variables in the workflow.</p>
workflowInfo	Contains the workflow metadata.	Required.	Contains the following sub-elements: <ul style="list-style-type: none"> <li>parallelSteps (optional)</li> <li>workflowID (required)</li> <li>workflowDescription (required)</li> <li>workflowVersion (required)</li> <li>vendor (required)</li> <li>configuration or general (optional)</li> </ul>	No attributes are supported for this element.
parallelSteps	For a workflow with automated steps, this property indicates whether the automated steps can be run in parallel (concurrently), thus possibly completing more quickly. For a parallel-steps workflow, this property is yes. Otherwise, if this property is omitted or set to no, automated steps are run one-by-one in the sequence in which they appear in the workflow, starting from the top of the workflow definition.	Optional	Boolean	No attributes are supported for this element.

Table 465. Workflow metadata elements (continued)

Element name	Description	Required or optional	Type	Supported attributes
workflowID	A short, arbitrary value that identifies the workflow.	Required.	nonNullString	<p>The following attributes are supported for the <i>workflowID</i> element:</p> <p><b>scope</b> Indicates the singleton scope for the workflow. The following values are valid:</p> <p><b>system</b> A maximum of one instance of this workflow can exist on any one system in the sysplex.</p> <p><b>sysplex</b> A maximum of one instance of this workflow can exist in the sysplex.</p> <p><b>none</b> No limit exists for the number of instances of this workflow. For a callable workflow, this setting means that a new instance is always created on the calling system.</p> <p>The default setting is <i>none</i>. Omitting the scope attribute has the same effect as the default setting.</p> <p><b>isCallable</b> Indicates whether the workflow can be called by another workflow, and, if so, the callable range for the workflow. The following values are valid:</p> <p><b>system</b> This workflow can be called only by another workflow that is running on the same system.</p> <p><b>sysplex</b> This workflow can be called by any workflow that is running in the same sysplex.</p>
workflowDefaultName	Default name for the workflow. This value is shown in the <b>Workflow name</b> field of the Workflows task when a user creates the workflow. If you omit this value, the Workflows task creates a name for the workflow.	Optional.	nonNullString	No attributes are supported for this element.
workflowDescription	A short description of the workflow.	Required.	nlsString	No attributes are supported for this element.

Table 465. Workflow metadata elements (continued)				
Element name	Description	Required or optional	Type	Supported attributes
workflowVersion	The version of this workflow definition file. Update this value whenever you change any portion of the workflow definition file, including changes to the primary XML file or any or referenced files.  The Workflows task caches only the latest version of any imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify the workflow definition. For this reason, when you author a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.	Required.	nonNullString	No attributes are supported for this element.
vendor	The name of the workflow provider.	Required.	nonNullString	No attributes are supported for this element.
<i>Workflow category</i> is a classification of the activities to be performed in the workflow. To indicate a category, specify one of the following elements: <i>configuration</i> , <i>provisioning</i> , or <i>general</i> . Specifying a workflow category element is optional. By default, the workflow category is <i>general</i> . The category elements are described in the next rows.				
Configuration	A workflow that is used to configure system software is classified as a <i>configuration workflow</i> .	Optional.	Contains the sub-elements that are listed in <a href="#">Table 466 on page 880</a> .	No attributes are supported for this element.
Provisioning	A workflow that is used to provision system software is classified as a <i>provisioning workflow</i> .	Optional.	Contains the sub-elements that are listed in <a href="#">Table 467 on page 882</a> .	No attributes are supported for this element.
General	All other workflows are classified as <i>general workflows</i> .	Optional.	Empty	No attributes are supported for this element.



Table 466. Sub-elements for a configuration type workflow			
Element name	Description	Required or optional	Type
productID	A short, arbitrary value that identifies the workflow.	Required.	nonNullString
productName	The name of the product	Required.	nonNullString
productVersion	The product version	Required.	nonNullString



Table 467. Sub-elements for a provisioning type workflow

Element name	Description	Required or optional	Type
<b>productID</b>	Identifier of the product or component that is being provisioned by the workflow, such as the product identifier (PID) or function modification identifier (FMID).	Required.	nonNullString
<b>productName</b>	Name of the product or component that is being provisioned by the workflow.	Required.	nonNullString
<b>productVersion</b>	Version and release of the product or component that is provisioned by the workflow.	Required.	nonNullString
<b>softwareType</b>	Type of software to be provisioned by the workflow.	Required.	

## Workflow upgrade elements summary

Table 468. Workflow upgrade elements			
Element name	Description	Required or optional	Type
<b>preserveOptions</b>	Contains the workflow upgrade options.	Optional.	Contains the following sub-elements: <ul style="list-style-type: none"> <li>• version (required)</li> <li>• variableSet (optional)</li> <li>• stepSet (optional)</li> <li>• workflowHistory (optional)</li> <li>• workflowNotes (optional)</li> <li>• include (required)</li> <li>• exclude (optional)</li> <li>• upgradeNotes (optional)</li> </ul>
<b>version</b>	Identifies the workflow version that can be upgraded by this workflow definition file.	Required; at least one.	Contains the following sub-elements: <ul style="list-style-type: none"> <li>• value (required)</li> <li>• type (required)</li> </ul>
<b>variableSet</b>	The variables to copy to new workflow instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> <li>• defaultChecked (optional)</li> </ul>
<b>stepSet</b>	The steps to copy to new workflow instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> <li>• defaultChecked (optional)</li> </ul>
<b>workflowHistory</b>	Specifies whether to copy the workflow history from the existing workflow to the new instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> <li>• defaultChecked (optional)</li> </ul>
<b>workflowNotes</b>	Specifies whether to copy the workflow notes from the existing workflow to the new instance.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> <li>• defaultChecked (optional)</li> </ul>
<b>include</b>	Specifies the step or variable defined by prior workflow definition file to be copied. Can be specified multiple times. It supports regular expression or variable name.	Required.	Contains the following sub-elements: <ul style="list-style-type: none"> <li>• name (optional)</li> <li>• mapTo (optional)</li> <li>• regExp (optional)</li> </ul>
<b>exclude</b>	Specifies the variables to exclude from the set that is generated by <include> elements.	Optional.	Contains the following sub-element: <ul style="list-style-type: none"> <li>• name (required)</li> </ul>

## Manifest elements summary

Table 469. Manifest elements summary table

Element name	Description	Required or optional	Type	Supported attributes
<b>translatedTextFiles</b>	Contains the language file definitions.	Optional.	If specified, it must contain the following sub-elements: <ul style="list-style-type: none"> <li>• bundle (required)</li> <li>• language (required)</li> </ul>	No attributes are supported for this element.
<b>bundle</b>	Contains the set of language files for the bundle	Required if the translatedTextFiles element is specified. A workflow can contain 1 — 500 bundles.	A sequence of language elements.	The following attribute is supported for the <i>bundle</i> sub-element: <p><b>name</b> The name of the bundle. The name is required, and must be a single-token string.</p>
<b>language</b>	Locates a file for a particular language.	Required if the bundle element is specified. A bundle can contain 1 — 10 languages.	Empty	The following attributes are supported for the <i>language</i> sub-element: <p><b>name</b> The language identifier as defined in the XML standard and RFC 3066, but using only the language portion without the country suffix. The identifier is required, and must be unique within a bundle.</p> <p><b>path</b> The path name of the language file. The path is required, and its data type must be a nonNullString. A language file can be a UNIX file, a sequential data set, or a PDS member. The path name format is described in “References to external files” on page 810.</p>

## Step elements summary

Table 470. Step elements summary table: Elements to use for defining all steps

Element name	Description	Required or optional	Type	Supported attributes
<b>step</b>	Contains attributes of a step. Up to 500 step elements can be defined in a workflow.	Required. At least one step must be defined.	<p>The step element can contain the following sub-elements:</p> <ul style="list-style-type: none"> <li>• title (required)</li> <li>• description (required)</li> <li>• prereqStep (optional)</li> <li>• If a step is a parent step (contains step elements), use the sub-element that is listed in Table 471 on page 890 to define the step.</li> <li>• If a step is a leaf step (contains no step elements), use one or more of the sub-elements that are listed in Table 472 on page 892 to define the step.</li> <li>• If a step invokes another workflow, use one or more of the sub-elements that are listed in Table 474 on page 910 to define the step.</li> </ul>	<p>The following attributes are supported for the <i>step</i> element:</p> <p><b>name</b> The name of the step. The name is required, and must be a string that consists of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must begin with a letter, and must be unique within the workflow.</p> <p><b>optional</b> Indicates whether the step is optional. This attribute is optional. If specified, its data type must be Boolean. By default, the value is <i>false</i>.</p>
<b>title</b>	A short description of the task.	Required.	nlsString	No attributes are supported for this sub-element.
<b>description</b>	A more detailed description of the step.	Required.	nlsRichString	No attributes are supported for this sub-element.
<b>prereqStep</b>	Identifies a step that must be completed before this step can be performed. Up to 499 prerequisite steps can be defined for a step.	Optional.	Empty	<p>The following attribute is supported for the <i>prereqStep</i> sub-element:</p> <p><b>name</b> Name of the prerequisite step. The name is required, must refer to a defined step, and must be a string that consists of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must start with a letter.</p>



*Table 471. Step elements summary table: Additional sub-element for parent steps*

Element name	Description	Required or optional	Type	Supported attributes
<b>step</b>	<p>A substep. Existence of this sub-element defines a parent step. This sub-element is subject to the 500 step maximum.</p> <p>The step sub-element is mutually exclusive with template, instructions, variableValue, skills, and weight.</p>	Required.	Recursive step definition as defined by this table.	No attributes are supported for this sub-element.



Table 472. Step elements summary table: Additional sub-elements for leaf steps

Element name	Description	Required or optional	Type	Supported attributes
<b>runAsUser</b>	User ID under which a particular step is to be performed. The value that you specify on the runAsUser element is considered to be the <i>runAsUser ID</i> for the step. When a runAsUser is not specified for a step, the step is performed under the step owner user ID.  The runAsUser element is required if the approver element is specified.	Optional.	nonNullString	The following attribute is supported for the <i>runAsUser</i> sub-element:  <b>substitution</b> Indicates whether the runAsUser value contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .  If you use variable substitution, understand that the variable must be an instance variable; it cannot be a global variable.
<b>approver</b>	A user ID, or a list of user IDs separated by blanks. At least one user ID must approve the step before it is performed on behalf of the user ID that is specified with the runAsUser element.  To specify multiple required approvers, use multiple approver elements (up to 12). The approver element is optional. If it is specified, the runAsUser element is required.	Optional	approverType	The following attribute is supported for the <i>approver</i> sub-element:  <b>substitution</b> Indicates whether the approver user IDs contain variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
<b>condition</b>	A conditional step.	Optional.	Condition for performing this step.	The following attributes are supported for the <i>condition</i> sub-element:  <b>expression</b> Provides an expression based on input variables (global or instance) and Boolean logic. When the expression resolves to <i>true</i> for the current workflow instance, the step can be performed. An expression is required, and must have the data type <i>conditionType</i> .  <b>description</b> Describes the condition that must be satisfied before the step can be performed. A description is required.  <b>targetState</b> The state to which the step is set when the expression evaluates to <i>true</i> . The valid values are <i>Ready</i> and <i>Skipped</i> . The target state is optional. If not specified, the default is <i>Ready</i> .

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>suspend</b>	<p>A suspended step. When automation processing reaches a step that includes the suspend element, automation stops at the step. Here, z/OSMF can send an email notification to one or more recipients that you specify, such as a person who should be prompted to take action.</p> <p>With the suspend element, you can define an email notification by using the following sub-elements of suspend:</p> <ul style="list-style-type: none"> <li>• toRecipients</li> <li>• subject</li> <li>• content</li> </ul>	Optional.	suspendType	No attributes are supported for this sub-element.
<b>toRecipients</b>	For a suspended step, specifies the email addresses of the persons to be notified of the suspended step. If you omit this value, no email is sent.	Optional.	velocityString	No attributes are supported for this sub-element.
<b>subject</b>	For a suspended step, specifies a subject for the notification email. If you omit this value, the email subject is set to no subject by default.	Optional.	velocityString	No attributes are supported for this sub-element.
<b>content</b>	For a suspended step, specifies the text of the message for the notification email. If you omit this value, the email content is set to no content by default.	Optional.	velocityString	No attributes are supported for this sub-element.

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>variableValue</b>	<p>References a variable that is defined earlier.</p> <p>The variableValue sub-element is mutually exclusive with the step sub-element.</p>	Optional.	Empty	<p>The following attributes are supported for the <i>variableValue</i> sub-element:</p> <p><b>name</b> The name of the referenced variable. The name is required, and must be a string consisting of letters (uppercase or lowercase), numeric digits, and the underscore character. This value must begin with a letter.</p> <p>The name and scope combination must refer to a defined variable.</p> <p><b>scope</b> The scope of the referenced variable. The scope is optional. If specified, the value must be <i>instance</i> or <i>global</i>. The default is <i>instance</i>.</p> <p>The name and scope combination must refer to a defined variable.</p> <p><b>required</b> Whether the variable must have a value for this step. The required attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>When set to <i>true</i>, the Workflows task does not allow the user to complete the step without providing a value (if a default is not defined in the XML).</p> <p><b>noPromptIfSet</b> Whether the variable widget is displayed in read-only mode, if the variable already has a value. The noPromptIfSet attribute is an optional, Boolean value. If not specified, the default is <i>false</i>, that is, always display the variable widget in read/write mode.</p>

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>instructions</b>	Detailed documentation on what the user must do to perform the step. This sub-element is mutually exclusive with the step sub-element.	Required.	nlsRichVelocityString	The following attribute is provided for the <i>instructions</i> sub-element: <b>substitution</b> Indicates whether instructions contain variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.
<b>weight</b>	The relative difficulty of the step compared to other steps within this workflow. The weight sub-element is mutually exclusive with step sub-element.	Required.	Integer value 1 - 1000.	No attributes are supported for this sub-element.
<b>skills</b>	The type of skills that are required to perform this step. The skills sub-element is mutually exclusive with step sub-element.	Optional.	nlsString	No attributes are supported for this sub-element.
<b>autoEnable</b>	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required. If <i>autoEnable</i> is not specified, the default is <i>false</i> .	Optional.	Boolean	No attributes are supported for this sub-element.
<b>canMarkAsFailed</b>	Indicates whether the step can be marked as <i>Failed</i> manually by the step owner. If <i>canMarkAsFailed</i> is not specified, the default is <i>false</i> . When set to <i>true</i> , the <b>Review Instructions</b> page in the Step Perform wizard includes the option to allow the step owner to mark the step as <i>Failed</i> manually. When <i>false</i> , this option is not displayed to the user.	Optional.	Boolean	No attributes are supported for this sub-element.
<b>rest</b>	Identifies the step as a REST step.	Optional.	String.	The sub-elements and attributes of a REST step are described in <a href="#">Table 473 on page 906</a> .

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>template</b>	<p>Identifies the step as a <i>template step</i>, which is a step that runs an executable program, such as a JCL job, a REXX exec, or a UNIX shell script.</p> <p>The template sub-element is mutually exclusive with the step sub-element.</p>	Optional.	<p>The template sub-element must contain one of the following sub-elements:</p> <p><b>inlineTemplate</b> A file template or executable template that is specified in the workflow definition file. The value must be type <code>velocityString</code>.</p> <p><b>fileTemplate</b> Path name of the external file that contains the template. The contents of the file are treated as type <code>velocityString</code>.</p>	<p>The following attribute is supported for both the <code>inlineTemplate</code> and <code>fileTemplate</code> sub-elements:</p> <p><b>substitution</b> Indicates whether the template contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.  A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.</p>

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>submitAs</b>	Indicates the type of executable program.	Optional.	<p><b>"TSO-REXX"</b> Run a REXX exec program in real time.</p> <p><b>"TSO-UNIX-REXX"</b> Run a REXX exec program for the UNIX environment in real time.</p> <p><b>"TSO-UNIX-shell"</b> Run a UNIX shell script in real time.</p> <p><b>"JCL"</b> Submit a JCL job for batch processing on z/OS. The results are indicated in the job log.</p> <p><b>"TSO-REXX-JCL"</b> Submit a JCL job that contains a REXX program. The program runs as a batch job on z/OS; the results are indicated in the job log.</p> <p><b>"shell-JCL"</b> Submit a JCL job that contains a UNIX shell script. The program runs as a batch job on z/OS; the results are indicated in the job log.</p> <p>A REXX exec that is written to be run in a UNIX shell environment should be submitted as "TSO-UNIX-shell" or "shell-JCL".</p>	<p>The following attribute is supported for the <i>template</i> element when the <i>submitAs</i> sub-element is "JCL", "TSO-REXX-JCL", or "shell-JCL":</p> <p><b>maxRc</b> Maximum return code value to consider successful. This attribute is optional. If specified, the value must be an integer in the range 0 - 4095. If not specified, the default is 0.</p>

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>maxlrecl</b>	<p>For a step that submits a job, this value specifies the maximum record length, in bytes, for the input data for the job.</p> <p>This value is used when the step is performed automatically (<i>autoEnable=true</i>). If the step is performed manually, the user can optionally specify the maximum record length on the <b>Edit JCL</b> page in the Workflows task.</p>	Optional.	Integer value 80 - 1024. The default is 1024 bytes.	No attributes are supported for this sub-element.

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>output</b>	<p>For a step that creates a properties file, this element specifies the default name and location of the file. The properties file can be a data set, a UNIX file, or a JES spool file.</p> <p>The Workflows task allows the user to modify the file name and location, as needed.</p> <p>For more information about the properties file, see “Creating a properties file” on page 827.</p>	Optional.	String.	<p>The following attributes are supported for the <i>output</i> sub-element:</p> <p><b>substitution</b> Indicates whether the properties file name uses variable substitution. This is an optional, Boolean value. The default is <i>false</i>.</p> <p><b>needResolveConflicts</b> Indicates whether the user is prompted to resolve variable conflicts from the properties file. This is an optional, Boolean value. The default is <i>true</i>.</p> <p>If variable conflicts exist, the user is prompted to resolve the conflicts. A value of <i>true</i> must be specified for the Workflows task to display the variables in the <b>Input Variables</b> page. If set to <i>false</i>, the Workflows task uses the properties file variables in place of any existing values without prompting the user. This setting applies to instance variables only; global variables are not overridden by variables in the properties file.</p> <p><b>sysoutDD=</b> Indicates whether the properties file is a JES spool file that is produced by a job step in the executed job. This is an optional, Boolean value. The default is <i>false</i>, which means that the properties file is either a data set or a UNIX file, as specified on the output element. If set to <i>true</i>, the properties file is a JES spool file. The location that is specified on the output element must identify a valid DD name and, optionally, a job step name.</p> <p><b>loadOutputFileArrayValue=</b> Indicates the behavior for handling variable conflicts. If set to <i>true</i>, the workflow uses the array variable values from the output file, rather than from the Workflows task. If set to <i>false</i>, the workflow uses the existing values from the Workflows task.</p>
<b>successPattern</b>	Regular expression that is returned for a successful program execution. This element is required. You must specify one (and only one) regular expression for a successful program completion.	Required.	String.	No attributes are supported for this sub-element.

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>failedPattern</b>	Optional regular expression that can be returned for program execution failures. You can omit this element or specify up to 100 different specifications for <i>failedPattern</i> . This property might be null.	Optional.	String.	No attributes are supported for this sub-element.
<b>outputVariablesPrefix</b>	For a step that creates a variable, this property contains a prefix that identifies a string as a variable. This property might be null.	Optional.	String.	<p>The following attribute is supported for the <i>outputVariablesPrefix</i> sub-element:</p> <p><b>needResolveConflicts</b> Indicates whether the user is prompted to resolve variable conflicts from the program output variables. This attribute is an optional, Boolean value. If not specified, the default is <i>true</i>.</p> <p>If variable conflicts exist, the user is prompted to resolve the conflicts. A value of <i>true</i> must be specified for the Workflows task to display the variables in the <b>Input Variables</b> page.</p> <p>If set to <i>false</i>, the Workflows task uses the output file variables in place of any existing values without prompting the user. This setting applies to instance variables only; global variables are not overridden by variables in the output file.</p> <p><b>loadOutputFileArrayValue=</b> Indicates the behavior for handling variable conflicts. If set to <i>true</i>, the workflow uses the array variable values from the output file, rather than from the Workflows task. If set to <i>false</i>, the workflow uses the existing values from the Workflows task.</p>
<b>scriptParameters</b>	<p>For a step that runs a program, this property contains the input parameters that can be set by the step owner. This property might be null.</p> <p>The following sub-elements are included on the <i>scriptParameters</i> element:</p> <ul style="list-style-type: none"> <li>• <i>description</i></li> <li>• <i>value</i></li> </ul>	Optional.	String.	No attributes are supported for this sub-element.

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>description</b>	Text description of the parameter in the <i>scriptParameters</i> element, such as its intended use or recommended value.	Required if the <i>scriptParameters</i> element is specified.	String.	No attributes are supported for this sub-element.
<b>value</b>	Value of the parameter in the <i>scriptParameters</i> element.	Required if the <i>scriptParameters</i> element is specified.	String.	No attributes are supported for this sub-element.
<b>procName</b>	For a step that runs a program under TSO/E, this property contains the name of the logon procedure that is used to log into the TSO/E address space. If no value was specified for the step, the default is IZUFPROC.	Optional.	String.	The following attribute is supported for the <i>procName</i> sub-element: <b>substitution</b> Indicates whether the procedure name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
<b>regionSize</b>	For a step that runs a program under TSO/E, this property contains the region size for the TSO/E address space. If no value is specified for the step, the default is 50000.	Optional.	Integer value 50000 - 2096128.	No attributes are supported for this sub-element.
<b>timeout</b>	For a step that runs a REXX exec or UNIX shell script, this property contains the maximum amount of time that the program can run before it is ended by a timeout condition.	Optional.	Integer value 60 - 3600.	No attributes are supported for this sub-element.
<b>saveAsDataset</b>	Data set name (fully qualified, no quotations) specifying where to save the file after the user edits it.  When a file is generated, the presence of this element results in the <i>save as data set</i> option being presented to the user, primed with the element value, if specified.  When a program is run, this element can be used as the default value. However, the Workflows task widget is always displayed to the user.	Optional.	velocityFileString	The following attribute is supported for the <i>saveAsDataset</i> sub-element: <b>substitution</b> Indicates whether the data set name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .  A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>saveAsUnixFile</b>	Path name that specifies where to save the file after the user edits it.  When a file is generated, the presence of this element results in the <i>save as UNIX file</i> option being presented to the user, primed with the element value, if specified.  When a program is run, this element can be used as the default value. However the Workflows task widget is always displayed to the user.	Optional.	velocityFileString	The following attribute is supported for the <i>saveAsUnixFile</i> sub-element:  <b>substitution</b> Indicates whether the path name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .  A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.
<b>predefinedVariable</b>	For a step that submits a job, this sub-element sets a variable in the body of the job. A predefined variable is treated as a string substitution. The substitution applies to the current step only.  You can specify multiple predefined variables per step.	Optional.	predefinedVariableType	The following attribute is supported for the <i>predefinedVariable</i> sub-element:  <b>name</b> Name of the predefined variable. The name is required, and must be of data type string.  To avoid overriding the variables that are defined for the workflow, use a unique name for the predefined variable.
<b>feedbackItem</b>	Optionally includes a feedback form for the step in the Workflows task with questions for the step owner to answer. Up to 100 feedback items (questions) can be specified for a step.	Optional.	feedbackItemType	The following attributes are supported for the <i>feedbackItem</i> sub-element:  <b>name</b> Name of the feedback question. The name is required, and must be of data type string.  <b>required</b> Indicates a required question. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
<b>question</b>	The question to be asked (a text string). For example: <i>How difficult was this step?</i>	Required, if the element <i>feedbackItem</i> is specified.	String.	No attributes are supported for this sub-element.

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>answers</b>	<p>Answer format, which is defined by including one of the following sub-elements:</p> <p><b>singleSelect</b> One answer (and only one) can be selected from the available choices.</p> <p><b>multipleSelect</b> More than one answer can be selected from the available choices.</p> <p><b>text</b> No choices are provided; the user answers the question by entering a text value.</p>	Required, if the element <i>feedbackItem</i> is specified.	answersType	No attributes are supported for this sub-element.
<b>singleSelect</b>	A sub-element of the element <i>answers</i> . Indicates that one answer (and only one) can be selected from the available choices that are listed in <i>answers</i> .	Optional.	selectType	<p>The following attribute is supported for the <i>multipleSelect</i> sub-element:</p> <p><b>hasOtherAnswer</b> Displays the choice "Other" as a selectable answer. Is displayed last in a list of multiple choices.</p>
<b>label</b>	One or more answers to be listed as selectable choices for the question. At least one label (answer) must be specified, up to a maximum of 50.	Required.	labelType	<p>The following attributes are supported for the <i>label</i> sub-element:</p> <p><b>value</b> Answer that can be selected. Required.</p> <p><b>required</b> Indicates a required question. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p>
<b>multipleSelect</b>	A sub-element of the element <i>answers</i> . Indicates that more than one answer can be selected from the available choices.	Optional.	selectType	<p>The following attribute is supported for the <i>multipleSelect</i> sub-element:</p> <p><b>hasOtherAnswer</b> Displays the choice "Other" as a selectable answer. Is displayed last in a list of multiple choices.</p>

Table 472. Step elements summary table: Additional sub-elements for leaf steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>label</b>	One or more answers to be listed as selectable choices for the question. At least one label (answer) must be specified, up to a maximum of 50.	Required.	labelType	The following attributes are supported for the <i>label</i> sub-element: <b>value</b> Answer that can be selected. Required. <b>required</b> Indicates a required question. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> .
<b>text</b>	A sub-element of the element <i>answers</i> . Indicates that no choices are provided; the user answers the question by entering a text value.	Optional.	String.	No attributes are supported for this sub-element.



Table 473. Step elements summary table: Additional sub-elements for REST steps

Element name	Description	Required or optional	Type	Supported attributes
<b>httpMethod</b>	Indicates the HTTP method that is used for issuing the REST request. The following values are valid: <ul style="list-style-type: none"> <li>• GET</li> <li>• PUT</li> <li>• POST</li> <li>• DELETE.</li> </ul>	Required.	String	None.
<b>schemeName</b>	Scheme name that is associated with the REST request. If specified, this element must be set to "http."	Optional.	String	None.
<b>hostname</b>	Host name or IP address of the system to which the REST request is directed. For example: <code>www.ibm.com</code> .	Optional	String	The following attribute is supported for the hostname sub-element: <b>substitution</b> Indicates whether the host name contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.
<b>port</b>	Port number to use for the REST request.	Optional.	String	The following attribute is supported for the port sub-element: <b>substitution</b> Indicates whether the port contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i> . A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variableValue</i> sub-element must be specified for the step.

Table 473. Step elements summary table: Additional sub-elements for REST steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>uriPath</b>	URI path to use for the REST request.	Required.	String	<p>The following attribute is supported for the uriPath sub-element:</p> <p><b>substitution</b> Indicates whether the URI path contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variable/value</i> sub-element must be specified for the step.</p>
<b>queryParameters</b>	For a GET or POST request, this element contains the query parameters.	Optional.	String	<p>The following attribute is supported for the queryParameters sub-element:</p> <p><b>substitution</b> Indicates whether the query parameters contain variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variable/value</i> sub-element must be specified for the step.</p>
<b>requestBody</b>	For a PUT or POST request, this element contains the request body.	Optional.	String	<p>The following attribute is supported for the requestBody sub-element:</p> <p><b>substitution</b> Indicates whether the request body contains variable substitution. This attribute is an optional, Boolean value. If not specified, the default is <i>false</i>.</p> <p>A value of <i>true</i> must be specified for the Workflows task to allow the variable substitution. If <i>true</i> is specified, at least one <i>variable/value</i> sub-element must be specified for the step.</p>
<b>expectedStatusCode</b>	The expected HTTP status code from the REST API request. If this value does not match the actualStatusCode value, the workflow step fails. This behavior is similar to what happens when a job template step returns a return code that is greater than the allowed maximum return code.	Required.	Integer	None

Table 473. Step elements summary table: Additional sub-elements for REST steps (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>actualStatusCode</b>	The actual HTTP status code that is received from the REST request. To obtain this value, map it to a workflow variable.	Optional	Integer	None
<b>propertyMapping</b>	The property from the REST response body that is mapped to a workflow variable. You can specify multiple <code>propertyMapping</code> elements in a REST step.	Optional	String	<p>The following attributes are supported for the <code>propertyMapping</code> sub-element:</p> <p><b>mapFrom</b> Specifies the supplied value to be mapped. This attribute is optional. If specified, the type must be <code>nonNullString</code>.</p> <p><b>mapTo</b> Specifies the variable to which the supplied value is to be mapped. This attribute is optional. If specified, the type must be <code>nonNullString</code>.</p>



Table 474. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow)

Element name	Description	Required or optional	Type	Supported attributes
<b>variableMapping</b>	Used to transfer variable values between the called workflow and calling workflow.	Optional.	Any	No attributes are supported for this sub-element.
<b>fromCallingToCalled</b>	Used to transfer variable values from the calling workflow to the called workflow.	Optional.		No attributes are supported for this sub-element.
<b>regExpression</b>	Used for filtering on variable names. Specify a portion of the variable name with one or more wildcard characters.	Optional.	nonNullString	No attributes are supported for this sub-element.
<b>variableName</b>	Name of the variable.	Optional.	nonNullString	The following attribute is supported for the <i>variableName</i> sub-element: <b>mapTo</b> Specifies the variable to which the supplied value is to be mapped. This attribute is optional. If specified, the type must be nonNullString.
<b>fromCalledToCalling</b>	Used to transfer variable values from the called workflow to the calling workflow.	Optional		The following attribute is supported for the <i>fromCalledToCalling</i> sub-element: <b>override</b> Indicates whether the variable settings in this element are to take precedence over the variables in the calling workflow. The default is false. This attribute is optional. If specified, it must be a Boolean value.
<b>regExpression</b>	Used for filtering on variable names. Specify a portion of the variable name with one or more wildcard characters.	Optional.	nonNullString	No attributes are supported for this sub-element.
<b>variableName</b>	Name of the variable.	Optional.		No attributes are supported for this sub-element.
<b>callingStepWeight</b>	The relative difficulty of the step compared to other steps within this workflow. The <i>callingStepWeight</i> sub-element is mutually exclusive with the <i>step</i> sub-element.	Required.	Integer value 1 - 1000.	No attributes are supported for this sub-element.

Table 474. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow) (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>callingStepSkills</b>	The type of skills that are required to perform this step. The <i>callingStepSkills</i> sub-element is mutually exclusive with the <i>step</i> sub-element.	Optional.	nlsString	No attributes are supported for this sub-element.
<b>callingStepAutoEnable</b>	Indicates whether the step is to be performed automatically when all prerequisite steps are completed, and no user inputs are required. If <i>callingStepAutoEnable</i> is not specified, the default is <i>false</i> .	Optional.	Boolean	No attributes are supported for this sub-element.
<b>canCallingStepMarkAsFailed</b>	Indicates whether the step can be marked as <i>Failed</i> manually by the step owner. If <i>canCallingStepMarkAsFailed</i> is not specified, the default is <i>false</i> .  When set to <i>true</i> , the <b>Review Instructions</b> page in the Step Perform wizard includes the option to allow the step owner to mark a step as <i>Failed</i> manually. When <i>false</i> , this option is not displayed to the user.	Optional.	Boolean	No attributes are supported for this sub-element.
<b>calledWorkflowDefinitionFile</b>	Path name of an external file that contains the workflow definition for the called workflow. Can be a fully qualified path name, or a relative path name (that is, relative to the location of the calling workflow).  For a relative path, the path must begin with <i>./</i> or <i>../</i> . After beginning with this notation, all other instances of <i>./</i> or <i>../</i> in the path will be resolved.	Optional.	Path name to template file, the contents of which are treated as type <i>velocityString</i> .	No attributes are supported for this sub-element.
<b>calledWorkflowDescription</b>	A more detailed description of the step.	Required.	nlsRichString	No attributes are supported for this sub-element.
<b>calledWorkflowID</b>	The name of the workflow. The combination of calledWorkflowID and calledWorkflowVersion must be unique within the Workflows task. You can use the calledWorkflowID and calledWorkflowVersion to identify a called workflow.	Optional.	String consisting of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must begin with a letter.	No attributes are supported for this sub-element.

Table 474. Step elements summary table: Additional sub-elements for steps that invoke another workflow (a called workflow) (continued)

Element name	Description	Required or optional	Type	Supported attributes
<b>calledWorkflowVersion</b>	<p>The version of this workflow definition file. Update this value whenever you change any portion of the workflow definition file, including changes to the primary XML file or any sub-files or referenced files.</p> <p>The Workflows task caches only the latest version of an imported workflow definition file. Therefore, to ensure that the most current version is used, you must update the version value whenever you modify the workflow definition. For this reason, when you create a workflow definition file, you might want to complete the development phase on a workstation before you import the workflow definition into the Workflows task.</p> <p>The combination of calledWorkflowID and calledWorkflowVersion must be unique within the Workflows task. You can use the calledWorkflowID and calledWorkflowVersion to identify a called workflow.</p>	Optional.	nonNullString	No attributes are supported for this sub-element.
<b>calledWorkflowMD5</b>	An MD5 encrypted value (a 128-bit hash value) that you can use to identify the called workflow.	Optional.	nonNullString	No attributes are supported for this sub-element.

## Variable definition elements and types summary

Table 475. Variable definition elements summary				
Element name	Description	Required or optional	Type	Supported Attributes
<b>variable</b>	Contains the definition of a variable. Up to 1500 variables can be defined in a workflow.	Optional.	<p>If specified, the <i>variable</i> element can contain the following sub-elements:</p> <ul style="list-style-type: none"> <li>• label (required)</li> <li>• abstract (required)</li> <li>• description (required)</li> <li>• exposeToUser (optional)</li> <li>• category (required)</li> <li>• datastore (optional)</li> </ul> <p>And, the <i>variable</i> element must contain at least one of the type-specific sub-elements, which are listed in <a href="#">Table 476 on page 917</a>.</p>	<p>The following attributes are supported for the <i>variable</i> element:</p> <p><b>name</b> The name of the variable. The name is required, and must be a string consisting of letters (uppercase or lowercase), numeric digits, the hyphen, and the underscore character. This value must begin with a letter.</p> <p>The combination of name and scope must be unique within the workflow.</p> <p><b>scope</b> The scope of the variable. The scope is required, and the value must be <i>instance</i> or <i>global</i>. The default is <i>instance</i>.</p> <p>The combination of name and scope must be unique within the workflow.</p> <p><b>visibility</b> Specifies whether the variable is intended for public or private use. This attribute is intended for the workflow author's use. The visibility setting does not affect how the variable is processed by the Workflows task. This attribute is optional. The default is <i>private</i>.</p>
<b>label</b>	A short label for the UI widget.	Required if the variable element is specified.	nlsString	No attributes are supported for this sub-element.
<b>abstract</b>	A brief description of the variable for the UI widget.	Required if the variable element is specified.	nlsString	No attributes are supported for this sub-element.
<b>description</b>	A longer explanation of what the variable is used for, and perhaps what the syntactic requirements are.	Required if the variable element is specified.	nlsRichString	No attributes are supported for this sub-element.

Table 475. Variable definition elements summary (continued)

Element name	Description	Required or optional	Type	Supported Attributes
<b>exposeToUser</b>	For a step that runs a JCL job. If included, the exposeToUser element indicates that the variable is to be included in the <b>List variables for substitution window</b> of the Workflows task. This element supports both global variables and instance variables in Workflows task. If this element is not specified, the variable is not shown in the Edit JOB statement page, and thus, cannot be selected by the user for the substitution of another value in the JOB statement.	Optional. Applicable only when the variable element is specified.	exposeToUserType	<b>usage</b> Specify the purpose of the variable. This text is displayed next to the variable in the Edit JOB Statement window of the Workflows task. The usage is optional. If specified, its data type must be an nlsString.
<b>category</b>	The name of the logical grouping to which this variable belongs. The default is <i>general</i> .	Required if the variable element is specified.	nlsString	No attributes are supported for this sub-element.
<b>datastore</b>	Place where the variable value is stored over time. Because z/OSMF is the only supported location for variable values, it is not necessary to specify the datastore element.	Optional. Applicable only when the variable element is specified.	A single, empty, required zOSMF sub-element	No attributes are supported for this sub-element.



Table 476. Variable definition type-specific sub-elements summary			
Element name	Description	Type	Supported attributes
<b>Boolean</b>	Boolean type	Empty	No attributes are supported for this sub-element.
<b>default</b>	Default value	Boolean. The default is <i>true</i> .	No attributes are supported for this sub-element.
<b>string</b>	String type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p><b>minLength</b> Minimum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the maxLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the validationType and regularExpression sub-elements.</p> <p><b>maxLength</b> Maximum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the minLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the validationType and regularExpression sub-elements.</p> <p><b>validationType</b> Validation types. This sub-element is optional. If specified, it must have one of the following values: ALPHA, ALPHAB, ALPHANUM, BIT, DSMEMBERNAME, DSNNAME, DSQUAL, GROUP, HEX, IPADDR, IPADDR4, IPADDR6, TSUSERID, UNIXID, USERID, VOLSER.</p> <p>The validationType sub-element mutually excludes the minLength and maxLength combination and the regularExpression sub-element.</p> <p>The UNIXID validation type verifies that a z/OS UNIX UID or GID is in the range 0 – 2147483647. Here, a UID or GID is treated as a string, not an integer. If you have code that treats a UID or GID as numeric, use an integer type to define the variable, instead of a string validation type. You can enforce the minimum and maximum values within the integer variable definition.</p> <p><b>regularExpression</b> Standard regular expression that constrains the variable value. This sub-element is optional. If specified, it must have the bigNonNullString data type.</p> <p>The regularExpression sub-element mutually excludes the minLength and maxLength combination and the validationType sub-element.</p> <p><b>errorMessage</b> Overrides the default error message for an incorrect value. This sub-element is optional. If specified, it must have an nlsString data type.</p> <p><b>choice</b> Added to the choice list. This sub-element is required if <i>valueMustBeChoice=true</i>. Up to 1337 choices are allowed, and each choice must adhere to any minLength, maxLength, validationType, and regularExpression specified.</p> <p><b>default</b> Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be an nlsUnboundedString that adheres to any minLength, maxLength, validationType, and regularExpression specified. If <i>valueMustBeChoice=true</i>, the default value must be one of the choice values specified.</p>	<p>The following attributes are supported for the <i>string</i> sub-element:</p> <p><b>multiline</b> Specifies a single-line text box widget or a multi-line text box. This attribute is optional, and its data type is Boolean. If unspecified, its value is <i>false</i> (default).</p> <p><b>valueMustBeChoice</b> Indicates whether the value must come from the provided choices. This attribute is optional, and its data type is Boolean. If unspecified, its value is <i>false</i> (default). If set to <i>true</i>, at least one <i>choice</i> sub-element must be specified.</p>
<b>integer</b>	Integer type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p><b>minValue</b> Minimum value. This sub-element is optional. If specified, its value must be an integer. If the maxValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p><b>maxValue</b> Maximum value. This sub-element is optional. If specified, its value must be an integer. If the minValue is specified, the maxValue must be greater than or equal to the minValue.</p> <p><b>default</b> Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be an integer that adheres to any minValue and maxValue specified.</p>	No attributes are supported for this sub-element.

Table 476. Variable definition type-specific sub-elements summary (continued)			
Element name	Description	Type	Supported attributes
<b>decimal</b>	Decimal type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p><b>minValue</b> Minimum value. This sub-element is optional. If specified, its value must be a decimal. If the max<b>Value</b> is specified, the max<b>Value</b> must be greater than or equal to the min<b>Value</b>.</p> <p><b>maxValue</b> Maximum value. This sub-element is optional. If specified, its value must be a decimal. If the min<b>Value</b> is specified, the max<b>Value</b> must be greater than or equal to the min<b>Value</b>.</p> <p><b>default</b> Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be a decimal that adheres to any min<b>Value</b> and max<b>Value</b> specified. Decimal places are rounded by the Workflows task, based on the decimal<b>Places</b> attribute value.</p>	<p>The following attribute is supported for the <i>decimal</i> sub-element:</p> <p><b>decimalPlaces</b> Maximum number of decimal places that can be specified. The value can be an integer in the range of 1 - 6. The default is 1.</p>
<b>time</b>	Time type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p><b>minValue</b> Minimum value. This sub-element is optional. If specified, its value must be the time in <i>hh:mm:ss</i> format. If the max<b>Value</b> is specified, the max<b>Value</b> must be greater than or equal to the min<b>Value</b>.</p> <p><b>maxValue</b> Maximum value. This sub-element is optional. If specified, its value must be the time in <i>hh:mm:ss</i> format. If the min<b>Value</b> is specified, the max<b>Value</b> must be greater than or equal to the min<b>Value</b>.</p> <p><b>default</b> Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be the time in <i>hh:mm:ss</i> format and must adhere to any min<b>Value</b> and max<b>Value</b> specified.</p>	No attributes are supported for this sub-element.
<b>date</b>	Date type	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p><b>minValue</b> Minimum value. This sub-element is optional. If specified, its value must be the date in <i>yyyy-mm-dd</i> format. If the max<b>Value</b> is specified, the max<b>Value</b> must be greater than or equal to the min<b>Value</b>.</p> <p><b>maxValue</b> Maximum value. This sub-element is optional. If specified, its value must be the date in <i>yyyy-mm-dd</i> format. If the min<b>Value</b> is specified, the max<b>Value</b> must be greater than or equal to the min<b>Value</b>.</p> <p><b>default</b> Widget that is primed with the value specified. This sub-element is optional. If specified, the default value must be the date in <i>yyyy-mm-dd</i> format and must adhere to any min<b>Value</b> and max<b>Value</b> specified.</p>	No attributes are supported for this sub-element.

Table 476. Variable definition type-specific sub-elements summary (continued)			
Element name	Description	Type	Supported attributes
<b>password</b>	Password type.	<p>If specified, this sub-element can contain one or more of the following sub-elements:</p> <p><b>minLength</b> Minimum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the maxLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the regularExpression sub-element.</p> <p><b>maxLength</b> Maximum string length. This sub-element is optional. If specified, its value must be a non-negative integer. If the minLength is specified, the maxLength must be greater than the minLength. The minLength and maxLength combination mutually excludes the regularExpression sub-element.</p> <p><b>regularExpression</b> Standard regular expression that constrains the variable value. This sub-element is optional. If specified, it must have the bigNonNullString data type.</p> <p>The regularExpression sub-element mutually excludes the minLength and maxLength sub-elements.</p> <p><b>errorMessage</b> Overrides the default error message for an incorrect value. This sub-element is optional. If specified, it must have an nlsString data type.</p>	No attributes are supported for this sub-element.
<b>array</b>	Array type	None; no sub-elements are applicable.	No attributes are supported for this sub-element.

**atCreate element summary**

Table 477. atCreate element summary				
Element name	Description	Required or optional	Type	Supported Attributes
<b>atCreate</b>	For users of the Create Workflow REST service, the atCreate element provides more options for working with variables. Up to 1500 atCreate elements can be defined in a workflow.	Optional.	atCreateType	<p>The following attributes are supported for the atCreate element:</p> <p><b>name</b> Specifies the variable for which the variable attributes are being set. This attribute is required.</p> <p><b>scope</b> Specifies the scope of the variable. This value is set to <i>instance</i> (the only valid value) or is omitted. The default is <i>instance</i>.</p> <p><b>required</b> For a workflow that is created through the Create Workflow REST service, this attribute indicates whether the variable must be set to a value at the time of workflow creation. This attribute is optional; the default is <i>false</i>. If a variable is marked as "required," but the variable is not given a value, an attempt to create the workflow through the Create Workflow REST service fails with an error.</p> <p>For a workflow that is created through the Workflows task user interface, this option is ignored.</p> <p><b>prompt</b> For users of the Create Workflow REST service, this attribute identifies a variable that <i>should</i> be prompted for by the program that issues the REST service. By itself, the prompt attribute does not enforce any behavior for the workflow creation. However, by setting prompt to <i>true</i>, you can indicate that prompting is recommended for the variable. The user of the Create Workflow REST service can query the value of the prompt attribute for any variables in the workflow to determine whether any variables should be prompted for. This attribute is optional; the default is <i>false</i>.</p>



---

## Chapter 3. Creating your own z/OSMF plug-ins

z/OSMF provides a modular framework that you can use to bring together all of your z/OS system management applications. z/OSMF supports different levels of integration ranging from adding a resource link or an application link to creating your own z/OSMF plug-ins.

To help you decide which integration method is best for you, consider the following recommendations:

- If your installation uses web-based applications that can launch or be launched by z/OSMF tasks or other web-based applications and display a specific context, it is recommended that you use the Application Linking Manager to create context-sensitive launch points between the applications.
- If your installation has commonly used web-based resources that do not have natural integration points with existing tasks or applications, it is recommended that you use the Links task to add the resource as a link in z/OSMF.
- If your installation requires function that is not provided in z/OSMF or in another web-based application, it is recommended that you create your own plug-in and use the Import Manager task to import the property file into z/OSMF.

Using these integration methods reduces context shifts between disparate applications, helps simplify the management of your z/OS mainframe systems, and moves your installation one step closer to providing a central location for z/OS system management tasks.

The remainder of this section explains how to create your own z/OSMF plug-in. For more information about the Application Linking Manager task or the Links task, see the z/OSMF online help.

### Example of an z/OSMF external plug-in

For an example of how to create and deploy your own z/OSMF external plug-in, see the example in GitHub. It contains a user interface that is based on the popular Angular framework, and uses several z/OSMF Representational State Transfer (REST) APIs to perform operations on a z/OS host system.

The sample plug-in is provided at the following location: <https://github.com/IBM/IBM-Z-zOS/tree/master/zOSMF/ExternalPluginExample-TSOBackend>.

### Process of creating a plug-in

In z/OSMF, a plug-in is a collection of one or more web-based applications (referred to as tasks) that add function to z/OSMF. z/OSMF ships with several plug-ins, which are described in [Selecting which z/OSMF services to add](#) in *IBM z/OS Management Facility Configuration Guide*.

In addition to the shipped plug-ins, z/OSMF allows you to create your own plug-ins to add installation-specific function to z/OSMF. The process of adding your own plug-ins to z/OSMF includes the following activities:

1. Developing a web-based application and the supporting documentation for the functions you want to add to z/OSMF.
2. Storing the application and its documentation in the UNIX file system, and setting 644 permissions for files and 755 permissions for folders.
3. Creating a property file in the UNIX file system that defines the parameters required for z/OSMF to configure your plug-in.
4. Using the z/OSMF Import Manager task to import the property file.
5. Setting up security for your plug-in. After which, you must refresh the security management product on your system and restart the z/OSMF server to have your changes take affect.

For more details, see the sections that follow.

## Developing web-based applications

---

The z/OSMF framework provides the infrastructure, security, and services you need to create the web-based applications to be included in your plug-ins.

Specifically, z/OSMF provides the following resources:

### JavaScript APIs

z/OSMF core provides application programming interfaces (APIs) that allow you to create plug-ins that interact with z/OSMF core and share data with the tasks in your plug-ins. Exploiting these APIs provides users with a consistent experience across z/OSMF tasks. For more details, see [“Using the z/OSMF core JavaScript APIs” on page 924](#) and [“Using the Application Linking Manager JavaScript APIs” on page 935](#).

### REST Services

z/OSMF provides several REST interfaces, which you can use to simplify the creation of your z/OSMF plug-ins. For more details, see [Chapter 1, “Using the z/OSMF REST services,” on page 1](#).

### Client Side Logger

z/OSMF provides a client-side logging framework that you can use to route client messages to the z/OSMF log, which is the same log used by plug-ins that ship with z/OSMF. For more details, see [“Logging client messages in the z/OSMF log” on page 946](#).

### File Retrieval Service

z/OSMF provides a file retrieval service that you can use to specify the file for z/OSMF core to display when it launches your application and to specify any additional files or resources your application may need. For more details, see [“Retrieving files and resources for your application” on page 951](#).

### Secure Environment

z/OSMF uses the System Authorization Facility (SAF) interface on the z/OS host system to authenticate users and to grant users access to z/OS system management tasks. z/OSMF security also depends on plug-in developers identifying and remediating security vulnerabilities in their applications. For more details, see [“Securing your applications” on page 969](#).

## Requirements

Your application must:

- Be written in a markup or programming language that a Web browser can interpret, such as HTML, JavaScript, and CSS. z/OSMF does not support markup or programming languages that must be interpreted by the z/OSMF server.
- Use Dojo 1.7 or later to interface with z/OSMF core; however, you are not required to code your application using Dojo.
- Be stored in the UNIX file system with 644 permissions for files and 755 permissions for folders.
- Add new function to z/OSMF or update a plug-in that you created and previously imported. You cannot use plug-ins to modify the z/OSMF framework, the z/OSMF user interface, or other z/OSMF plug-ins.

## Using the z/OSMF core JavaScript APIs

z/OSMF provides the `zosmfExternalTools` JavaScript API, which z/OSMF tasks can use to define actions to be performed before z/OSMF closes the task, to obtain a unique user session identifier, and to store and manage public objects in z/OSMF core.

The API is located at the following path: `/zosmf/js/zosmf/util`. To access the functions in the API and to dynamically add functions to the API, you must import and instantiate the `zosmfExternalTools` object in your task's HTML file. Sample code is provided in [Figure 427 on page 925](#).

```
require(["dojo/_base/window","zosmf/util/zosmfExternalTools"],
        function(win, zosmfExternalTools) {
            win.global.zosmfExternalTools = new zosmfExternalTools();
        });
```

Figure 427. Sample JavaScript code for importing and instantiating the global `zosmfExternalTools` object

For information about the functions that are included in or can be added to the API, see the following sections:

- [“Functions provided in the `zosmfExternalTools` API” on page 925](#)
- [“Functions you can add to the `zosmfExternalTools` API” on page 925](#)

## Functions provided in the `zosmfExternalTools` API

Table 478 on page 925 lists the functions that are provided in the `zosmfExternalTools` API.

Table 478. Functions provided in the <code>zosmfExternalTools</code> API		
Function	Usage	Where described
<b><code>programmaticallyCloseTab</code></b>	Call this function to request for z/OSMF core to close your task.	<a href="#">“<code>programmaticallyCloseTab</code> function” on page 929</a>
<b><code>cleanupBeforeDestroyComplete</code></b>	Call this function to inform z/OSMF core that the actions performed by the <code>cleanupBeforeDestroy</code> function are complete. After which, z/OSMF closes your task.	<a href="#">“<code>cleanupBeforeDestroyComplete</code> function” on page 931</a>
<b><code>getUserSessionId</code></b>	Call this function to retrieve the unique session identifier that z/OSMF core creates for each user and z/OSMF instance combination.	<a href="#">“<code>getUserSessionId</code> function” on page 932</a>
<b><code>definePublicObject</code></b>	Call this function to define a public object in z/OSMF core.	<a href="#">“<code>definePublicObject</code> function” on page 932</a>
<b><code>retrievePublicObject</code></b>	Call this function to retrieve a public object from z/OSMF core.	<a href="#">“<code>retrievePublicObject</code> function” on page 934</a>
<b><code>deletePublicObject</code></b>	Call this function to delete a public object from z/OSMF core.	<a href="#">“<code>deletePublicObject</code> function” on page 934</a>

## Functions you can add to the `zosmfExternalTools` API

Table 479 on page 926 lists the functions you can define and dynamically add to the `zosmfExternalTools` API. z/OSMF core calls these functions when specific events are triggered.

Table 479. Functions you can add to the `zosmfExternalTools` API

Function	Usage	When called	Where described
<b>isContentChanged</b>	Define this function if you want your task to check for unsaved changes before z/OSMF closes the task.	z/OSMF core calls your <code>isContentChanged</code> function when: <ul style="list-style-type: none"> <li>A user clicks the X icon to close the z/OSMF tab that contains your task.</li> <li>Your task calls the <code>programmaticallyCloseTask</code> function.</li> </ul>	<a href="#">“isContentChanged function” on page 926</a>
<b>shouldClose</b>	Define this function if you want to delay the close task request for an unspecified period of time so that your task can perform some additional actions.	z/OSMF core calls your <code>shouldClose</code> function when a user clicks the X icon to close the z/OSMF tab that contains your task.	<a href="#">“shouldClose function” on page 928</a>
<b>cleanupBeforeDestroy</b>	Define this function if you want to delay the close task request for up to one second so that your task can perform some cleanup actions before z/OSMF closes the task.	z/OSMF core calls your <code>cleanupBeforeDestroy</code> function when a user: <ul style="list-style-type: none"> <li>Logs out of z/OSMF.</li> <li>Clicks the X icon to close the z/OSMF tab that contains your task.</li> <li>Closes the browser tab or window.</li> <li>Changes the URL in the browser and redirects away from z/OSMF.</li> </ul>	<a href="#">“cleanupBeforeDestroy function” on page 930</a>

## isContentChanged function

If your task allows users to save input values, modified content, or selections, consider defining an `isContentChanged` function that will check for unsaved changes before z/OSMF closes the task. Doing so prevents users from inadvertently discarding their work.

## Invoking the function

The `isContentChanged` function must be defined off the `zosmfExternalTools` object. To define the `isContentChanged` function, use the syntax shown in [Figure 428 on page 926](#).

```
win.global.zosmfExternalTools.isContentChanged = function ( ) {
  //Define your function here.

  //Return either true or false.
  return true|false;
}
```

Figure 428. Syntax to use for the `isContentChanged` function

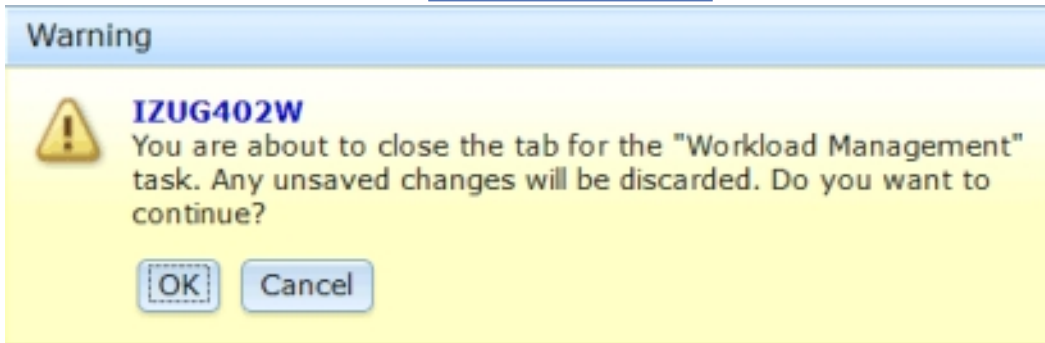
z/OSMF core calls the *isContentChanged* function provided for your task when a user clicks the X icon to close the tab that contains your task or when your task calls the *programmaticallyCloseTask* function.

## Return values

The *isContentChanged* function returns a Boolean value, which indicates the following:

### **true**

Indicates that changes are pending. In this case, z/OSMF core displays a prompt that warns the user that unsaved changes will be discarded, and gives the user the option to proceed with closing the task or to cancel the close task request. [Figure 429 on page 927](#) depicts a sample prompt.



*Figure 429. Sample confirmation window for a close task request*

### **false**

Indicates that no changes are pending. In this case, z/OSMF core closes the task tab and does not display a prompt.

**Note:** If the *zosmfExternalTools* object or the *isContentChanged* function do not exist, z/OSMF core will display a prompt regardless of whether changes are pending.

## Example

Suppose your plug-in contains a task that has multiple editable fields on a single page. When the user clicks the X icon to close your task tab, you want to check for changes for each field. [Figure 430 on page 928](#) provides sample code you can use for this scenario.

```

//Define the default zosmfExternalTools.isContentChanged() function.
function init(){
  win.global.zosmfExternalTools = new zosmfExternalTools();
  win.global.zosmfExternalTools.isContentChanged = function (){
    return false;
  }
}

//When the page is created, call the internal _isContentChanged() function.
postCreate: function() {

  //Define the isContentChanged function.
  var that = this;
  win.global.zosmfExternalTools.isContentChanged = function (){
    return that._isContentChanged();
  }

},

//Determine if the content was changed.
_isContentChanged: function(){
  ...
},

//When the page is destroyed, restore the default isContentChanged function.
uninitialize: function() {
  win.global.zosmfExternalTools.isContentChanged = function (){
    return false;
  }
}

```

*Figure 430. Sample code for the isContentChanged function*

## shouldClose function

When a close task request is submitted, if your task needs to perform cleanup actions or collect information from users before z/OSMF closes the task, consider defining a *shouldClose* function to override requests to close your task. Doing so gives your task an unlimited amount of time to perform the actions, and allows your task to inform z/OSMF when it is ready to be closed.

## Invoking the function

z/OSMF core calls the *shouldClose* function provided for your task when a user clicks the X icon to close the tab that contains your task. The *shouldClose* function must be defined off the *zosmfExternalTools* object. To define the *shouldClose* function, use the syntax shown in [Figure 431 on page 928](#).

```

win.global.zosmfExternalTools.shouldClose = function ( ) {
  //Define your function here.

  //Return either true or false.
  return true|false;
}

```

*Figure 431. Syntax to use for the shouldClose function*

## Return values

The *shouldClose* function returns a Boolean value, which indicates the following:

### true

Indicates that z/OSMF core should close the task.

### false

Indicates that z/OSMF core should not close the task at this time. The task will submit a close task request when it is ready to be closed. Return *false* if you need to perform additional steps or cleanup

before closing the task. The task will remain open until your task calls the `programmaticallyCloseTab` function.

**Tip:** There is no time limit on how long the tab can remain open after a user requests for it to be closed. If the additional actions will take more than a few seconds to complete, consider providing an indicator so that users know the close request is being processed.

**Note:** If the `zosmfExternalTools` object or the `shouldClose` function do not exist, z/OSMF core will close the task.

## Example

```
var functionToCall = function(){
  //Task tab is now open indefinitely.
  //Perform all cleanup work.
  ....
  //When its finished, call the programmaticallyCloseTab function.
  win.global.zosmfExternalTools.programmaticallyCloseTab(myPluginId,myTaskId);
}

win.global.zosmfExternalTools.shouldClose = function(){
  if(longCleanupNeeded){
    //If cleanup actions will take longer than one second,
    //call the shouldClose function, and sSet the cleanup
    //function to start asynchronously after the shouldClose
    //function returns false.
    setTimeout(functionToCall,5);
    return false;
  }else{
    //if we dont need more time and should close
    return true;
  }
}
```

## programmaticallyCloseTab function

If your task overrode or intercepted a user's request to close your task, call the *programmaticallyCloseTab* function to request for z/OSMF core to close your task. Otherwise, your task will remain open.

## Invoking the function

To call the *programmaticallyCloseTab* function, use the syntax shown in [Figure 432 on page 929](#).

```
win.global.zosmfExternalTools.programmaticallyCloseTab(pluginId, taskId);
```

*Figure 432. Syntax to use to call the *programmaticallyCloseTab* function*

where,

### **pluginId**

Unique identifier assigned to the plug-in that contains the task.

### **taskId**

Unique identifier assigned to the task that you want to close.

When your task calls this function, z/OSMF core looks up the *pluginId* and *taskId* and attempts to close the corresponding z/OSMF tab. This close task request has the same behavior as the close request that is submitted when a user clicks the X to close the task tab.

For this service to work, z/OSMF core must be the parent of the tab that contains your task. For example, z/OSMF can use the *programmaticallyCloseTab* function to close your task when your task is launched from the z/OSMF navigation tree or when the task is launched with context by the Application Linking Manager.

z/OSMF cannot use this service to close your task if the task is open in another browser tab or window or if a user made your task a link and launched the task directly.

## Example

Suppose you have a plug-in with the ID *com.company.product*, and it contains a task with the ID *Product Name*. To close the task using the *programmaticallyCloseTab* function, you can use the sample code provided in Figure 433 on page 930.

```
function closeMyself(){
  //Ensure that the Close Task confirmation window is not displayed.
  win.global.zosmfExternalTools.isContentChanged = function (){
    return false;
  }

  //Call the parent to refer to z/OSMF core. Then, call the function.
  parent.programmaticallyCloseTab("com.company.product","Product Name");
}
```

Figure 433. Sample code for the *programmaticallyCloseTab* function

## cleanupBeforeDestroy function

When a close task request is submitted, if your task needs to perform cleanup actions before z/OSMF closes the task, such as ending a TSO/E address space, consider defining a *cleanupBeforeDestroy* function. Doing so delays a close task request for up to one second so that your task can perform the cleanup actions.

## Invoking the function

The *cleanupBeforeDestroy* function must be defined off the *zosmfExternalTools* object. To define the *cleanupBeforeDestroy* function, use the syntax shown in Figure 434 on page 930.

```
function init(){
  win.global.zosmfExternalTools = new zosmfExternalTools();

  //Define the cleanupBeforeDestroy function.
  win.global.zosmfExternalTools.cleanupBeforeDestroy = function (obj) {

    //Perform some cleanup
    .....

    //When cleanup is complete, call the cleanupBeforeDestroyComplete
    //function using the same parameter you used for the
    //cleanupBeforeDestroy function.
    win.global.zosmfExternalTools.cleanupBeforeDestroyComplete(obj);
  }
}
```

Figure 434. Syntax to use for the *cleanupBeforeDestroy* function

where,

### obj

Object that z/OSMF core creates to identify the task. You can specify any parameter name, but you must use the same name for the *cleanupBeforeDestroy* and *cleanupBeforeDestroyComplete* functions.

z/OSMF core calls the *cleanupBeforeDestroy* function provided for your task when a user:

- Logs out of z/OSMF.
- Clicks the X icon to close the z/OSMF tab that contains your task.
- Closes the browser tab or window.

- Changes the URL in the browser and redirects away from z/OSMF.

z/OSMF core calls the *cleanupBeforeDestroy* function every time one of the aforementioned events occurs. If a user has multiple instances of your task open or is working with multiple tasks in your plug-in, ensure that you do not release resources until the last task or the last instance of your task is closed.

After z/OSMF core calls your *cleanupBeforeDestroy* function, the task will remain open until you call the *cleanupBeforeDestroyComplete* function or until one second elapses, at which point, z/OSMF core will automatically close your task.

**Tip:** If you define a *cleanupBeforeDestroy* function, you do not need to define a *shouldClose* function because z/OSMF assumes *shouldClose* will return *false*.

## Example

```
win.global.zosmfExternalTools.cleanupBeforeDestroy = function(obj) {  
    //A synchronous cleanup method that cleans up objects on the client  
    //to free up memory.  
    myApi.cleanupMethod();  
  
    //Inform z/OSMF core that the cleanup actions are complete  
    //and that the task is ready to be closed.  
    win.global.zosmfExternalTools.cleanupBeforeDestroyComplete(obj);  
};
```

Figure 435. Sample code for the *cleanupBeforeDestroy* function

If *myApi.cleanupMethod()* is an asynchronous method, call the *cleanupBeforeDestroyComplete* function after the asynchronous action completes. Otherwise, the asynchronous method might not have time to complete.

## cleanupBeforeDestroyComplete function

If your task defined a *cleanupBeforeDestroy* function, call the *cleanupBeforeDestroyComplete* function to inform z/OSMF core that your cleanup actions are complete and that your task is ready to be closed.

## Invoking the function

To call the *cleanupBeforeDestroyComplete* function, use the syntax shown in [Figure 436 on page 931](#).

```
win.global.zosmfExternalTools.cleanupBeforeDestroyComplete(obj);
```

Figure 436. Syntax to use to call the *cleanupBeforeDestroyComplete* function

where,

### obj

Name of the parameter that was passed to the *cleanupBeforeDestroy* function.

The *cleanupBeforeDestroyComplete* function must be called after your *cleanupBeforeDestroy* function completes. Otherwise, z/OSMF might close the task before the cleanup actions are complete.

For example, if your *cleanupBeforeDestroy* function is performing an XMLHttpRequest (XHR) call, the *cleanupBeforeDestroyComplete* function must be called at the end of the XHR call and not immediately after the XHR call is issued.

**Note:** If your task does not call the *cleanupBeforeDestroyComplete* function within one second after z/OSMF invokes your *cleanupBeforeDestroy* function or if errors occur with the *cleanupBeforeDestroy* function, z/OSMF will close your task after one second elapses.

## Example

```
win.global.zosmfExternalTools.cleanupBeforeDestroy = function(obj) {  
    //A synchronous cleanup method that cleans up objects on the client  
    //to free up memory.  
    myApi.cleanupMethod();  
  
    //Inform z/OSMF core that the cleanup actions are complete  
    //and that the task is ready to be closed.  
    win.global.zosmfExternalTools.cleanupBeforeDestroyComplete(obj);  
};
```

Figure 437. Sample code for the *cleanupBeforeDestroyComplete* function

If `myApi.cleanupMethod()` is an asynchronous method, call the *cleanupBeforeDestroyComplete* function after the asynchronous action completes. Otherwise, the asynchronous method might not have time to complete.

## getUserSessionId function

z/OSMF core creates a unique ID for the authenticated user for every browser tab or window in which the user has z/OSMF or a z/OSMF task opened. Your task can use the ID for any purpose. To retrieve the ID, call the *getUserSessionId* function.

## Invoking the function

To call the *getUserSessionId* function, use the syntax shown in [Figure 438 on page 932](#).

```
win.global.zosmfExternalTools.getUserSessionId();
```

Figure 438. Syntax to use to call the *getUserSessionId* function

The session ID has the form *userName\_timestamp*, where *userName* is the ID the user used to log into z/OSMF, and the *timestamp* is the date and time the user logged into z/OSMF or opened z/OSMF or a z/OSMF task in a new browser tab or window. The timestamp is based on the locale and time zone setting for the user's browser.

## Example

```
var userSessionID;  
function init(){  
    //Create a unique session ID for the session to track if multiple  
    //instances of the application are open.  
    userSessionID=win.global.zosmfExternalTools.getUserSessionId();  
  
    //This session ID can be used in any client side logging.  
}
```

Figure 439. Sample code for the *getUserSessionId* function

## definePublicObject function

In z/OSMF, a public object is a Dojo AMD class that IBM, you, or a vendor defined that has been instantiated and stored in z/OSMF core, making the class accessible to any vendor and any z/OSMF task. If your plug-in contains multiple tasks and you want to pass data between those tasks, consider creating a public object for the tasks to share. To do so, call the *definePublicObject* function.

For information about creating Dojo classes, see the following web pages:

- <http://dojotoolkit.org/documentation/tutorials/1.7/declare/>

- <http://dojotoolkit.org/reference-guide/1.7/dojo/declare.html>

## Invoking the function

To call the *definePublicObject* function, use the syntax shown in [Figure 440](#) on page 933.

```
win.global.zosmfExternalTools.definePublicObject(class,handle,callback[,
    dojoPackageAlias,dojoPackageLocation]);
```

*Figure 440. Syntax to use to call the *definePublicObject* function*

where,

### **class**

Specify either the path to the Dojo class or the object of the Dojo class that you want to publicly store in z/OSMF core. When specifying the path, you can specify a path that is relative to the Dojo package location, or you can specify the absolute path. The class is required.

### **handle**

Specify the name you want z/OSMF core to assign to the public object. The handle is required.

### **callback**

Specify the function you want z/OSMF core to call when the object is available for your plug-in to use. The callback function is required.

### **dojoPackageAlias**

Specify the shortcut or alias to use when referring to the location of the package that contains the Dojo classes. z/OSMF core will map the alias to the actual location of the package. The Dojo package alias is required only if you specify a path for the **class** parameter.

### **dojoPackageLocation**

Specify the absolute path to the Dojo package. z/OSMF core will map this path to the Dojo package alias you specified. The Dojo package location is required only if you specify a path for the **class** parameter.

When your plug-in calls the *definePublicObject* function, if a path is specified for the **class** parameter, z/OSMF core:

1. Resolves the alias, and locates the Dojo package.
2. Instantiates the Dojo class, turning it into an object.
3. Stores the object in z/OSMF core under the specified handle.
4. Calls the callback function, which you provided, and passes the object to the function for your plug-in to use.

When your plug-in calls the *definePublicObject* function, if an object of the Dojo class is specified for the **class** parameter, z/OSMF core:

1. Stores the object in z/OSMF core under the specified handle.
2. Calls the callback function, which you provided, and passes the object to the function for your plug-in to use.

## Example

In the following example, the path to the Dojo package is `/zosmf/IzuUICommon/externalfiles/util/js`. The alias for this path is `zosmfUtil`.

The path to the class that will be defined as a public object is `/zosmf/IzuUICommon/externalfiles/util/js/UtilConstants`. With the alias, the path to the class becomes `zosmfUtil/UtilConstants`. The public object will be called `utilGlobal`.

After z/OSMF core creates a new instance of the public object, z/OSMF core calls the *testCallback* function and supplies the handle for the object.

```
var testCallback = function(obj){
    console.log("callback called!");
    console.log("with object "+obj);
}

function init(){
    win.global.zosmfExternalTools.definePublicObject(
        "zosmfUtil/UtilConstants", "utilGlobal", testCallback,
        "zosmfUtil", "/zosmf/IzuUICommon/externalfiles/util/js");
}
```

Figure 441. Sample code for the *definePublicObject* function

## retrievePublicObject function

To retrieve a public object that is stored in z/OSMF core, call the *retrievePublicObject* function.

If a plug-in contains multiple tasks, for the first task opened during a z/OSMF session, call the *definePublicObject* function to create the object. When subsequent tasks in the plug-in are opened, call the *retrievePublicObject* function to obtain an instance of the object that has already been created.

If the *retrievePublicObject* function returns an empty string, the object has not been created in z/OSMF core. This may be an indicator that the calling task needs to define the public object.

## Invoking the function

To call the *retrievePublicObject* function, use the syntax shown in [Figure 442 on page 934](#).

```
var publicObject=win.global.zosmfExternalTools.retrievePublicObject(handle);
```

Figure 442. Syntax to use to call the *retrievePublicObject* function

where *handle* is the handle used when the object was created.

## Example

```
var util=win.global.zosmfExternalTools.retrievePublicObject("utilGlobal");
console.log("utilGlobal "+utilGlobal);
```

Figure 443. Sample code for the *retrievePublicObject* function

## deletePublicObject function

To delete a public object that is stored in z/OSMF core, call the *deletePublicObject* function. After you delete a public object, it is not retrievable through the *retrievePublicObject* function because the object is no longer stored in z/OSMF core.

## Invoking the function

To call the *deletePublicObject* function, use the syntax shown in [Figure 444 on page 935](#).

```
win.global.zosmfExternalTools.deletePublicObject(handle,param);
```

*Figure 444. Syntax to use to call the deletePublicObject function*

where,

**handle**

Handle used when the object was created.

**param**

JSON object array that contains the name and value for each parameter that z/OSMF core will pass to the object's destroy() method. Specifying parameters is optional. The syntax to use follows:

```
{parm1: value1, parm2: value2, parm3: value3}
```

When your plug-in calls the *deletePublicObject* function, z/OSMF core:

1. Uses the handle to retrieve the public object.
2. Cleans up the public object.
3. Calls the destroy() method if the method is defined in the object. The destroy() method is used to further clean up the object.
4. Passes any parameters to the destroy() method.

The public object is also deleted when a user:

- Logs out of z/OSMF.
- Clicks the X icon to close the z/OSMF tab that contains your task.
- Closes the browser tab or window.
- Changes the URL in the browser and redirects away from z/OSMF.

## Example

```
win.global.zosmfExternalTools.deletePublicObject("utilGlobal");
```

*Figure 445. Sample code for the deletePublicObject function*

## Using the Application Linking Manager JavaScript APIs

If your installation uses multiple, disparate Web interfaces to manage your z/OS systems, use the z/OSMF Application Linking Manager to connect the applications. Doing so allows one task or application -- an event requestor -- to request that specific function or context be launched in another task or application -- the event handler, providing a smoother transition between applications.

z/OSMF provides the following resources for working with the Application Linking Manager:

- **Application Linking Manager task**, which provides a graphical user interface that you can use to add, query, or remove event type and event handler definitions.
- **Application Linking Manager REST APIs**, which are a set of REST services that allows a client application to add, query, or remove event type and event handler definitions.
- **AppLinker JavaScript API**, which is a set of JavaScript services that allows a client application to send events to the Application Linking Manager or to define the context to be displayed. The JavaScript services are applicable only if you are creating your own z/OSMF plug-in.

The remainder of this section describes the AppLinker JavaScript API. For information about the Application Linking Manager task, see the z/OSMF online help. For details about the REST APIs, see [“Application Linking Manager interface services”](#) on page 7.

z/OSMF predefines several event types, requestors, and handlers. For a list, see [“Event types, requestors, and handlers shipped with z/OSMF”](#) on page 9.

## Importing and instantiating the AppLinker API

To participate in the application linking process as an event requestor or an event handler, your task needs access to the functions provided in the AppLinker API, which is located at the following path: /zosmf/js/zosmf/izual.

To access the functions in the API, you must import and instantiate the AppLinker API in your task’s HTML file. Sample code is provided in [Figure 446](#) on page 936.

```
//Add a package for izual
packages: [{
  name: "izual",
  location: "/zosmf/js/zosmf/izual"
}]

//Import the AppLinker API.
require(["izual/api/PluginAppLinker17"],
  function(PluginAppLinker){
  }
//Instantiate the global AppLinker variable.
win.global.applinker = new PluginAppLinker();

//After instantiation, call the zOSMFTools getAppLinker function
win.global.zosmfTools=new Tools();
var localAppLinkerVariable = win.global.zosmfTools.getAppLinker();
```

*Figure 446. Sample code for importing and instantiating the AppLinker API*

## Functions provided in the AppLinker API

Table 480 on page 936 lists the functions that are provided in the AppLinker API.

Table 480. Functions provided in the AppLinker API		
Function	Usage	Where described
<b>sendEvent</b>	If your task is an event requestor, call this function to send an event to the Application Linking Manager.	<a href="#">“sendEvent function” on page 938</a>
<b>getHandlers</b>	If your task is an event requestor, call this function to determine if handlers are available to process an event.	<a href="#">“getHandlers function” on page 939</a>
<b>hasLaunchContext</b>	If your task is an event handler and supports the <i>launch with context</i> or <i>launch with context and reload</i> launching option, call this function to determine if your task is being loaded as a result of an application linking event.	<a href="#">“hasLaunchContext function” on page 940</a>
<b>getEventFromUrl</b>	If your task is an event handler and the hasLaunchContext function returns <i>true</i> , call the getEventFromUrl function to retrieve the event information that was supplied with the event.	<a href="#">“getEventFromUrl function” on page 941</a>

Table 480. Functions provided in the Applinker API (continued)		
Function	Usage	Where described
<b>subscribe</b>	If your task is an event handler and supports the <i>launch with context and switch</i> launching option, call this function to define a JavaScript function that z/OSMF core will call when an event of the specified type is delivered to your task.	<a href="#">“subscribe function” on page 943</a>
<b>onLoadingComplete</b>	If your task is an event handler and supports the <i>launch with context and switch</i> launching option, after your task subscribes to the event types it can handle, call this function to inform the Application Linking Manager that your task is ready to handle events.	<a href="#">“onLoadingComplete function” on page 945</a>

## Using the Applinker functions in the application linking process

The application linking process consists of the following steps:

1. An event requestor defines a user interface control that invokes the *sendEvent* function when a user performs an action.
2. An event requestor calls the *getHandlers* function to determine if handlers are available to process the request. If handlers are not available, the event requestor might perform an action such as disabling or hiding the user interface control.
3. A user performs an action on the user interface control that triggers the call of the *sendEvent* function.
4. The Application Linking Manager searches the set of known event types for the type identified by the event.
5. If a match is found, the Application Linking Manager searches for event handlers that are registered for this event type. If only one handler is found, it is launched. Otherwise, the user is prompted to select the handler to launch.
6. After the handler is identified, z/OSMF core identifies the launch context the handler supports.
7. If the launch context is *launch without context*, z/OSMF core uses the URL provided in the handler definition to launch the handler. If the handler is already open, it receives focus.
8. If the launch context is *launch with context*, *launch with context and reload*, or *launch with context and switch*, z/OSMF core does the following:
  - Appends the event type and parameters to the URL provided in the handler definition.
  - Completes one of the following actions:
    - If the launch context is *launch with context*, z/OSMF core uses the modified URL to launch the handler. If the handler is already open, it receives focus.
    - If the launch context is *launch with context and reload*, z/OSMF core uses the modified URL to launch the handler. If the handler is already open, a message is displayed warning the user that the current context will be overwritten.
    - If the launch context is *launch with context and switch*, the following steps are completed:
      - a. z/OSMF core uses the modified URL to launch the handler.
      - b. If the handler is loading for the first time, the handler calls the *subscribe* function for each event type to which it wants to subscribe and provides the function that z/OSMF core will call to determine the context to display.
      - c. After the handler subscribes to all the event types it can process, the handler calls the *onLoadingComplete* function to inform z/OSMF core that it is ready to accept events.

- d. When an event occurs, z/OSMF core verifies that the handler has called the *onLoadingComplete* function. If the function has been called, z/OSMF core searches the list of handlers that have subscribed for this event type, and identifies the callback function for the selected handler.  
**Important:** Register your task for each event type to which your task subscribes. Otherwise, z/OSMF core will not use the subscription because users will not have the option of selecting your task as the handler for the event.
  - e. z/OSMF core calls the *subscribe* callback function.
  - f. The function returns the context for z/OSMF core to display.
  - g. z/OSMF core displays the context and finishes loading the handler.
9. To display the correct context for the *launch with context* and *launch with context and reload* launching options, the handler must do the following while it is being loaded:
- a. Call the *hasLaunchContext* function to determine if an event has occurred.
  - b. Call the *getEventFromUrl* function to extract the event information from the URL that z/OSMF used to launch the handler.
  - c. Use the event information to display the correct context. For example, the handler might call another JavaScript function, which you provide, that can process the event information and return the correct context.

## sendEvent function

If your task is an event requestor, call the *sendEvent* function to send an event to the Application Linking Manager. The *sendEvent* function initiates the application linking process.

### Overview

The *sendEvent* function is typically attached to a user interface control, such as a link, a button, or an action. When the user interacts with this control, the task calls the *sendEvent* function, which supplies the event type ID and parameters to the Application Linking Manager. The Application Linking Manager performs several actions to identify the handler to be launched, and then it launches the handler.

To complete the application linking process, the event type specified for the *sendEvent* must be registered with the Application Linking Manager, and at least one handler must be available to process the request.

To minimize errors during the application linking process, call the *getHandlers* function and verify that at least one handler is available to handle the event. If no handlers are available, consider hiding or disabling the user interface control that calls the *sendEvent* function.

### Invoking the function

To call the *sendEvent* function, use the syntax shown in [Figure 447 on page 938](#).

```
win.global.applinker.sendEvent(eventTypeId, params);
```

Figure 447. Syntax to use to call the *sendEvent* function

where,

#### eventTypeId

ID that identifies the type of event.

#### params

JSON object array that contains the name and value for each parameter that your task will provide with the event. Specifying parameters is optional. The syntax to use follows:

```
{parm1: value1, parm2: value2, parm3: value3}
```

## Example

```
win.global.applinker.sendEvent("IBM.ZOSMF.VIEW_DATASET",{dataSetName:"myDataSet"});
```

*Figure 448. Sample code for the sendEvent function*

## getHandlers function

If your task is an event requestor, consider calling the *getHandlers* function to determine if handlers are available to process your request. The *getHandlers* function does not initiate application linking. It helps your task determine if application linking is possible.

### Overview

The *getHandlers* function identifies handlers that satisfy the following criteria:

- The handler is registered as a handler for the event type.
- The handler is enabled for the event type.
- The user is authorized to access the handler.

If one or more handlers satisfy this criteria, application linking is possible. Otherwise, application linking is not possible. In the latter case, consider hiding or disabling the user interface control that will initiate the application linking process. Doing so increases the usability of your task because the control is enabled or displayed only when the Application Linking Manager can successfully process the user's request.

**Tip:** If the *getHandlers* function does not find handlers that satisfy the aforementioned criteria, ensure that the event type is registered with the Application Linking Manager. If the event type is registered, verify that the event type ID is spelled correctly in your task.

## Invoking the function

To call the *getHandlers* function, use the syntax shown in [Figure 449 on page 939](#).

```
//Define a function for the getHandlers function to call if the request
//completes without errors.
var callback = function(response){
    //Specify what to do if there are handlers.
}
else {
    //Specify what to do if there are no handlers.
}
}

//Define a function for the getHandlers function to call if errors
//occur with the request.
var errback = function(error){
    //Specify how to proceed.
}

//Call the getHandlers function.
win.global.applinker.getHandlers(eventTypeId,callback,errback);
```

*Figure 449. Syntax to use to call the getHandlers function*

where,

**response**

JSON object array, which is provided by the *getHandlers* function, that contains the name of each handler that is available to process the event. To access the handlers in the array, use `response.results`.

**errors**

JSON object array, which is provided by the *getHandlers* function, that contains the error messages the function received. To access the messages in the array, use `error.messages`.

**eventTypeId**

ID that identifies the type of event.

**callback**

Function, which you provide, that the *getHandlers* function will call if it completes without errors.

**errback**

Function, which you provide, that the *getHandlers* function will call if errors occur when it is processing the request.

**Example**

```
var callback = function(response){
    if(response && response.results && response.results.length>0){
        //Handlers exist; enable the user controls.
    }else{
        //No handlers exist; disable the user controls.
    }
}
var errback = function(error){
    //Error occurred when retrieving handlers; retrieve the messages.
    var messages = error.messages;
}
win.global.applinker.getHandlers("IBM.ZOSMF.VIEW_DATASET",callback,errback);
```

**hasLaunchContext function**

If your task is an event handler, when your task is loading, call the *hasLaunchContext* function to determine if your task is being launched as a result of an application linking event. Doing so allows your task to determine if it needs to collect event information and display a specific context, or if it can display the main page.

**Invoking the function**

Call the *hasLaunchContext* function only if the launching option specified for your task in the handler definition is *launch with context* or *launch with context and reload*. For the *launch without context* launching option, z/OSMF core does not collect event information because there is no context to display; therefore, it is not necessary for your task to call the *hasLaunchContext* function.

For the *launch with context and switch* launching option, z/OSMF core alerts your task that an event has occurred when it calls the callback function you specified for the subscribe function; therefore, it is not necessary for your task to call the *hasLaunchContext* function to determine if an event has occurred.

To call the *hasLaunchContext* function, use the syntax shown in [Figure 450 on page 940](#).

```
win.global.applinker.hasLaunchContext();
```

*Figure 450. Syntax to use to call the *hasLaunchContext* function*

**Return values**

The *hasLaunchContext* function returns a Boolean value, which indicates the following:

**true**

Indicates that the Application Linking Manager delivered an event to the task. Call the *getEventFromUrl* function to parse the URL that z/OSMF core used to launch the task and retrieve the event information.

**false**

Indicates that the Application Linking Manager has not delivered an event to the task. In this case, z/OSMF core will launch the task without context.

## Example

```
function init(){
{
  if(win.global.applinker.hasLaunchContext()){
    //The task was opened with application linking; therefore, update context.

    //Use the getEventFromUrl function to obtain the context.
    var result = win.global.applinker.getEventFromUrl();
    var eventType = result.type;
    var params = result.params;

    //Use the eventType and parameters to switch the context.
  }
  else{
    //The task was not opened with application linking; therefore, display
    //the standard starting page.
  }
}
}
```

Figure 451. Sample code for the *hasLaunchContext* function

## getEventFromUrl function

If your task is an event handler and the *hasLaunchContext* function returns *true*, call the *getEventFromUrl* function to parse the URL that z/OSMF core used to launch your task. Doing so will provide your task with the event type ID and parameters it needs to display the correct context. Otherwise, your task will launch without context.

## Invoking the function

To call the *getEventFromUrl* function, use the syntax shown in [Figure 452 on page 941](#).

```
win.global.applinker.getEventFromUrl();
```

Figure 452. Syntax to use to call the *getEventFromUrl* function

## Return values

The *getEventFromUrl* function returns an event object that includes the following information:

**type**

The event type ID supplied with the event.

**params**

Array that contains the name and value of each parameter supplied with the event.

**Tip:** While your task is loading, it must use the event information to display the correct context. For example, your task might call another JavaScript function, which you provide, that can process the event information and return the correct context.

## Expected results

The event information returned by the *getEventFromUrl* function depends on the launching option that is specified in the event handler definition for your task and whether your task is already open when an event is received.

Table 481 on page 942 describes the expected result for each launching option and task state (open or closed) combination.

Table 481. Expected results by launching option and task state	
Launching option and task state	Expected result
<b>The launching option is <i>launch without context</i> and the task is closed.</b>	<p>The <i>getEventFromUrl</i> function will return <i>null</i>, and z/OSMF core will launch the task using the URL supplied in the handler definition.</p> <p><b>Tip:</b> z/OSMF core does not collect event information for this launching option because there is no context to display; therefore, it is not necessary for your task to call the <i>getEventFromUrl</i> function.</p>
<b>The launching option is <i>launch without context</i> and the task is already open.</b>	<p>The <i>getEventFromUrl</i> function will return <i>null</i>, and z/OSMF core will bring the existing task tab into focus.</p> <p><b>Tip:</b> z/OSMF core does not collect event information for this launching option because there is no context to display; therefore, it is not necessary for your task to call the <i>getEventFromUrl</i> function.</p>
<b>The launching option is <i>launch with context</i> and the task is closed.</b>	<p>The <i>getEventFromUrl</i> function will return the event type ID and parameters for the current event, and the task will display the current context.</p>
<b>The launching option is <i>launch with context</i> and the task is already open.</b>	<p>The <i>getEventFromUrl</i> function will return one of the following values:</p> <ul style="list-style-type: none"> <li>• The event type ID and parameters for the event that z/OSMF core used to initially open the task.</li> <li>• <i>Null</i> if the task was not initially opened as a result of an application linking event.</li> </ul> <p>z/OSMF core will bring the existing task tab into focus, and the task will display its last context.</p>
<b>The launching option is <i>launch with context and reload</i> and the task is closed.</b>	<p>The <i>getEventFromUrl</i> function will return the event type ID and parameters for the current event, and the task will display the current context.</p>
<b>The launching option is <i>launch with context and reload</i> and the task is already open.</b>	<p>z/OSMF core brings the existing tab into focus and displays a message, which warns the user that the current context will be overwritten and gives the user the option to proceed with displaying the new context or keeping the previous context.</p> <p>If the user clicks <b>OK</b>, the <i>getEventFromUrl</i> function will return the event type ID and parameters for the new event, and the task will display the new context.</p> <p>If the user clicks <b>Cancel</b>, the <i>getEventFromUrl</i> function will return one of the following values, and the task will display its last context:</p> <ul style="list-style-type: none"> <li>• The event type ID and parameters for the previous event.</li> <li>• <i>Null</i> if no event has occurred.</li> </ul>

Table 481. Expected results by launching option and task state (continued)

Launching option and task state	Expected result
<b>The launching option is <i>launch with context and switch</i> and the task either is closed or is already open.</b>	<p>The <i>getEventFromUrl</i> function will return the event information for the current event, and your task will determine the context to display based on the callback function you provided as a parameter for the <i>subscribe</i> function.</p> <p><b>Tip:</b> z/OSMF core supplies your task with the event type ID and event parameters when it calls the <i>subscribe</i> callback function; therefore, it is not necessary for your task to call the <i>getEventFromUrl</i> function to retrieve the event information.</p> <p>For information about the <i>subscribe</i> function, see <a href="#">“subscribe function”</a> on page 943.</p>

## Example

```
function init(){
{
  if(win.global.applinker.hasLaunchContext()){
    //The task was opened with application linking; therefore, update context.

    //Use the getEventFromURL function to obtain the context.
    var result = win.global.applinker.getEventFromURL();
    var eventType = result.type;
    var params = result.params;

    //Use the eventType and parameters to switch the context.

  }
  else{
    //The task was not opened with application linking; therefore, display
    //the standard starting page.
  }
}
```

Figure 453. Sample code for the *getEventFromUrl* function

## subscribe function

If your task is an event handler and supports the *launch with context and switch* launching option, you must define the context for your task to display. To do so, your task must call the *subscribe* function for each event type it can handle and provide a JavaScript function for z/OSMF core to call when the Application Linking Manager delivers an event of the specified type.

## Difference between subscribing and registering for an event type

Subscribing your task to an event type is different than registering your task as a handler for an event type. Registration is done regardless of launch context and it informs z/OSMF core that your task can handle events of the specified type. As such, z/OSMF core will present your task to users as a possible handler for events of that type.

Subscriptions are supported for the *launch with context and switch* launching option only, and it tells z/OSMF core how to process events of the specified type.

For the *launch with context and switch* launching option, your task must register for and subscribe to an event type. If your task subscribes but does not register as a handler for an event type, z/OSMF core will not use your task's subscription because it is impossible for users to select your task as the handler for the event type. If your task registers but does not subscribe to an event type, z/OSMF core will launch your task without context.

## Invoking the function

Your task must call the *subscribe* function while it is still loading so that when it is loaded the correct context is shown. To call the *subscribe* function, use the syntax shown in [Figure 454 on page 944](#).

```
//Define a function that specifies how to handle the event.
var subscribeCallback = function(params, eventData) {
    //Define your callback function here.
}

//Call the subscribe function.
win.global.applinker.subscribe(eventTypeId, subscribeCallback);
```

*Figure 454. Syntax to use to call the subscribe function*

where,

### **params**

Object that contains an array of the parameters supplied with the event. z/OSMF core will provide this object when it calls your callback function.

### **eventData**

Object that contains the event type ID. z/OSMF core will provide this object when it calls your callback function.

### **eventTypeId**

ID of the event type to which your task is subscribing.

### **subscribeCallback**

Function, which you provide, that specifies the context for your task to display when z/OSMF core delivers an event with the specified ID to your task.

**Tip:** If your *subscribeCallback* function is a function in your current object, you might be required to use `dojo.hitch(this,"subscribeCallback")` to specify it in the *subscribe* function call.

To subscribe to multiple event types, your task must call the *subscribe* function for each event type and, more than likely, provide a unique subscribe callback function for each call. For example, if your task is subscribing to three event types, it will call the *subscribe* function three times and, possibly, provide three unique callback functions.

After your task subscribes to all the event types it can handle, your task must call the *onLoadingComplete* function to inform z/OSMF core that it is ready to accept events. If your task does not call the *onLoadingComplete* function, z/OSMF core will not deliver the event that initially launched your task. z/OSMF core will, however, deliver subsequent events if your task has subscribed to the corresponding event type.

When z/OSMF core delivers an event to your task, z/OSMF core searches the list of tasks that have subscribed for the corresponding event type and identifies the callback function for your task. Then, z/OSMF core calls the callback function, and displays the context the function returns.

**Tip:** Your task does not have to unsubscribe the callback functions because z/OSMF core automatically unsubscribes the callback functions when your task uninitializes the AppLinker API.

## Example

```
var viewDataset = function(response){
    //View the parameters in the response, and perform the context switching.
}

function init(){
    win.global.applinker.subscribe("IZU.ZOSMF.VIEW_DATASET",viewDataset);
}
```

*Figure 455. Sample code for the subscribe function*

## onLoadingComplete function

If your task is an event handler and supports the *launch with context and switch* launching option, after your task subscribes to the event types it can handle, call the *onLoadingComplete* function to inform z/OSMF core that your task is ready to receive events.

### Invoking the function

To call the *onLoadingComplete* function, use the syntax shown in [Figure 456 on page 945](#).

```
win.global.applinker.onLoadingComplete(suppressInitialEvents);
```

Figure 456. Syntax to use to call the *onLoadingComplete* function

where, *suppressInitialEvents* is a Boolean variable that indicates the following:

#### true

Indicates that z/OSMF core will suppress the event that was used to initially launch your task. That is, z/OSMF core will ignore the first event, but will deliver subsequent events to the task.

**Tip:** Consider setting the *suppressInitialEvents* variable to *true* if you always want your task to load a certain way. For example, you might opt to set this variable to *true*, if you want your task to display a warning message or other important information that users must review before proceeding with your task.

If you want your task to process the initial event at a later time, assuming that no subsequent events have been received, you can retrieve the event information by calling the *getEventFromUrl* function and pass that information to the callback function that z/OSMF core would have called.

#### false

Indicates that z/OSMF core will deliver the event that was used to initially launch your task so that your task can display the correct context.

**Tip:** If you want to set the *suppressInitialEvents* variable to *false*, you can use the syntax that is provided in either of the following options:

```
//Option 1:  
call win.global.applinker.onLoadingComplete(false);  
  
//Option 2:  
call win.global.applinker.onLoadingComplete();
```

### Example

```
function init(){  
    setupMyPlugin();  
}  
  
function setupMyPlugin(){  
    //Perform setup actions.  
    ...  
    //Call the onLoadingComplete function to indicate that loading is  
    //complete so the task can switch context.  
    win.global.applinker.onLoadingComplete(false);  
}
```

## Logging client messages in the z/OSMF log

z/OSMF provides a client side logging framework, which you can use to log your plug-in's client messages in the z/OSMF log. Using the client side logger helps simplify debugging activities because messages for all z/OSMF plug-ins are included in the same log.

For more information about the z/OSMF log, including how to access and view the log, see the topic about working with z/OSMF runtime log files in [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

### Setting up client side logging

To implement client side logging, add code to your plug-in that:

- Specifies the name and location of the Dojo package that contains the client side logger.
- Imports the client side logger package.
- Creates an instance of the client side logger, and assigns a unique identifier to the logger. To uniquely identify your plug-in's client side logger, include the name of your company, product, and module in the identifier, and specify "ui" as the lower level qualifier. For example, `com.ibm.zosmf.NavigationTree.ui`.
- Defines the methods for the logging levels to be included in the z/OSMF log. The logging levels include: FINE, FINER (trace), FINEST (debug), INFO, WARNING, and SEVERE (fatal).

To help you identify your plug-in's messages in the log, append a prefix to each message that contains the module name, package name, and method name.

```
var LOGPREFIX=MODULE+" "+PACKAGE_NAME+" "+methodName+": ";  
LOGGER.entering(LOGPREFIX+"entry stuff");
```

*Figure 457. Sample code for creating a log prefix*

For sample log data, see “[Sample z/OSMF client side log data](#)” on page 948.

- Issues JavaScript log statements that match the logging levels you specified. When issued, the client side logger stores the statements in a message queue until the interval, threshold, or trigger is activated, as follows:

#### **interval**

By default, the client side logger sends the queued messages to the z/OSMF log every 60 seconds.

#### **threshold**

By default, the message queue can contain a maximum of 50 JavaScript log statements. When this threshold is reached, the client side logger sends the queued messages to the z/OSMF log.

#### **trigger**

By default, the push level for the client side logger is INFO. When the severity of a JavaScript log statement equals or exceeds the push level, the client side logger sends the queued messages to the z/OSMF log.

**Note:** To prevent denial of service attacks or filling the log with useless data, the client side logger logs only messages that your plug-in issues on behalf of z/OSMF authenticated users.

### Sample code for setting up client side logging

[Figure 458 on page 947](#) provides sample code that you can use as a reference when setting up the client side logger.

```

var dojoConfig = {
  isDebug: false,
  parseOnLoad: false,
  usePlainJson: true,
  async: true,

  //Specify the name and location of the client side logger package.
  packages: [{name: "izuLogger", location: "/zosmf/IzuUICommon/js"}]
};

//Import the client side logger package.
require(["izuLogger/izuUILogger/log4js17"],

function(log4js){
  //Create an instance of the client side logger.
  var LOGGER = log4js.getLogger("com.mycompany.myproduct.mymodule.ui");
  var MODULE = "MYMODULE";
  var PACKAGE_NAME = "MYPACKAGE.jsp";

  //Define methods for logging messages.
  function init(){
    var methodName="init";
    var LOGPREFIX=MODULE+" "+PACKAGE_NAME+" "+methodName+": ";
    LOGGER.entering(LOGPREFIX+"entry stuff");
    LOGGER.finest(LOGPREFIX+"finest stuff");
    LOGGER.finer(LOGPREFIX+"finer stuff");
    LOGGER.info(LOGPREFIX+"info stuff");
    LOGGER.warning(LOGPREFIX+"warning stuff");
    LOGGER.severe(LOGPREFIX+"severe stuff");
    LOGGER.exiting(LOGPREFIX+"exiting stuff");
  }
});

```

Figure 458. Sample code for setting up client side logging

## Methods provided for the client side logger

Table 482 on page 947 lists the methods you can use when setting up the client side logger, and provides summary descriptions and sample JavaScript code for each method.

Table 482. Methods provided for the client side logger		
Method	Description	Sample Code
<b>log4js.getLogger(logger-ID);</b>	Creates an instance of the client side logger for your plug-in, and assigns a unique identifier ( <i>logger-ID</i> ) to the logger.	<pre>var LOGGER = log4js.getLogger(   "com.ibm.zosmf.NavigationTree.ui");</pre>
<b>loggerName.log-level(param);</b>	Defines the log levels ( <i>log-level</i> ) you want the specified logger ( <i>loggerName</i> ) to include in the z/OSMF log, and takes a single string parameter ( <i>param</i> ) that specifies the information to log for each message. For the <i>loggerName.severe(param)</i> method, the client side logger automatically flushes messages to the z/OSMF log.	<pre>LOGGER.warning(LOGPREFIX+"Test warning message"); LOGGER.severe(LOGPREFIX+" Exception: " + text);</pre>
<b>loggerName.entering(params);</b>	Logs the entry point of each method, and takes zero or more comma-separated parameters ( <i>params</i> ).	<pre>LOGGER.entering(LOGPREFIX+"entry stuff");</pre>
<b>loggerName.exiting(params);</b>	Logs the exit point of each method, and takes zero or more comma-separated parameters ( <i>params</i> ).	<pre>LOGGER.exiting(LOGPREFIX+"exiting stuff");</pre>
<b>loggerName.logp(Level.log-level, params)</b>	Allows you to specify the parameters ( <i>params</i> ) to include in the z/OSMF log for the specified log level ( <i>log-level</i> ). This method takes a variable number of comma-separated parameters, and is an alternative to the <i>loggerName.log-level(param)</i> method.	<pre>LOGGER.logp(Level.WARNING, PACKAGE_NAME, MODULE, methodName, msg, " targetid=", targetid); LOGGER.logp(Level.SEVERE, PACKAGE_NAME, MODULE, methodName, "Exception received attempting to authenticate user", userid, exception);</pre>
<b>loggerName.isLoggable(Level.log-level)</b>	Verifies that the specified log level ( <i>log-level</i> ) is currently being logged.	<pre>if(LOGGER.isLoggable(Level.FINEST)){   LOGGER.finest(LOGPREFIX+ "FTP Host : " + dojo.byId('&lt;%= HADDataAdaptor.FTPHOSTNAME %&gt;'). value); }</pre>
<b>loggerName.logMessage("msgID", msgvars);</b>	Logs the message ID as well as the values that were substituted for each parameter. The values are stored as an array.	<pre>var msgvars = ["tkdole", 10, "try again"]; LOGGER.logMessage("IZUG400E", msgvars);</pre>

Table 482. Methods provided for the client side logger (continued)		
Method	Description	Sample Code
<code>loggerName.turnOnTracing();</code>	Enables tracing by setting the log level to FINER.	<code>LOGGER.turnOnTracing();</code>
<code>loggerName.turnOffTracing();</code>	Disables tracing, and sets the log level to its initial level.	<code>LOGGER.turnOffTracing();</code>
<code>loggerName.turnOnPopup();</code>	Appends messages to a popup window. The default log level is FINEST. You can also append messages to a popup window by setting the <code>setpopup</code> property to <code>true</code> . For more details, see <a href="#">"Logging messages to a popup window" on page 950</a> .	<code>LOGGER.turnOnPopup();</code>
<code>loggerName.turnOffPopup();</code>	Stops appending messages to a popup window (default), and closes the popup window. You can also stop appending messages to a popup window by setting the <code>setpopup</code> property to <code>false</code> . For more details, see <a href="#">"Logging messages to a popup window" on page 950</a> .	<code>LOGGER.turnOffPopup();</code>
<code>loggerName.setLogToConsole(true   false);</code>	If set to <code>true</code> , the client side logger appends the messages to the browser console, for example, Firebug. If set to <code>false</code> (default), the messages are not appended to the browser console.	<code>LOGGER.setLogToConsole(true);</code> <code>LOGGER.setLogToConsole(false);</code>
<code>loggerName.flush();</code>	Sends the queued messages to the z/OSMF log. You can also use the <code>push</code> property to push queued messages to the z/OSMF log. For more details, see <a href="#">"Pushing messages to the z/OSMF log" on page 949</a> .	<code>LOGGER.flush();</code>

## Sample z/OSMF client side log data

The message entries in the z/OSMF log have a uniform structure, which is depicted in [Figure 459 on page 948](#).

```
2009-04-29T22:08:09.609Z|00000031|com.ibm.zosmf.util.log.servlet.UILoggerServlet|UILoggerServlet::doPost()
FINER: [2009-04-29T22:07:13.938Z] ENTRY MYMODULE MYPACKAGE.jsp init: entry stuff
[tx0000000000002183:zosmfad@localhost (POST) /zosmf/IzuUICommon/UILoggerServlet?preventCache=1240947046138]
2009-04-29T22:08:09.609Z|00000031|com.ibm.zosmf.util.log.servlet.UILoggerServlet|UILoggerServlet::doPost()
FINER: [2009-04-29T22:07:13.474Z] RETURN MYMODULE MYPACKAGE.jsp init: exiting stuff
[tx0000000000002184:zosmfad@localhost (POST) /zosmf/IzuUICommon/UILoggerServlet?preventCache=1240946722135]
```

Figure 459. Sample z/OSMF client side log data

The first line of a log record contains the following data:

- Date and time the message was added to the log in ISO8601 format, set to UTC timezone. Example: 2009-04-29T22:08:09.609Z.
- Thread ID as an 8 digit hex number. Example: 00000031.
- Class name. Example: com.ibm.zosmf.util.log.servlet.UILoggerServlet.
- Method name. Example: UILoggerServlet::doPost().

The next line of a log record contains the following data:

- Logging level. Possible logging levels include FINE, FINER, FINEST, INFO, WARNING, and SEVERE.
- Date and time the message occurred in ISO8601 format, set to UTC timezone. Example: [2009-04-29T22:07:13.938Z].
- Indicator of the beginning (ENTRY) or end (RETURN) of a routine if the logging level is FINER.
- Log prefix if a prefix was specified for the logging level.
- Message ID and message text. Message IDs that begin with "IZU" are part of the z/OSMF product.

If the log record includes an exception, the exception class and the message text are logged next followed by the traceback information that is embedded in the exception. If the exception has attached causes, each cause is also logged with "+->" indicating the start of an attached cause.

The final line of a log record contains the following data:

- Transaction ID, which is an internal counter that applies to all actions between a specific set and is clear of a context.

- User ID of the user who was logged into z/OSMF when the message was issued.
- Host name of the system where the user logged into z/OSMF.
- Servlet "verb". Examples include (GET) and (POST).
- URL of the request and query string.

## Modifying the default settings for the client side logger

z/OSMF core provides default settings for several properties used to manage the client side logger. This section explains how to modify those default settings.

The default settings are described in the following sections:

- [“Displaying the default settings” on page 949](#)
- [“Pushing messages to the z/OSMF log” on page 949](#)
- [“Logging messages to a popup window” on page 950](#)
- [“Redirecting messages that cannot be written to the z/OSMF log” on page 950](#)
- [“Setting the interval for flushing the queue” on page 951](#)

## Displaying the default settings

To display a list of all the default values for the client side logger properties, use the following HTTP request:

---

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=getprops&loggername={name}
```

---

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "\_\_OPER=getprops&loggername=name" returns all the default settings for the specified client side logger (*name*).

## Pushing messages to the z/OSMF log

The client side logger uses the severity of JavaScript log statements to determine when queued messages are pushed to the z/OSMF log. That is, when the severity of a JavaScript log statement equals or exceeds the push level, the client side logger immediately sends the queued messages to the z/OSMF log. The default push level is INFO. To manage the push level, use the following HTTP requests:

---

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=getLevel&loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=setLevel&level={level}&
loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=push&loggername={name}
```

---

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "\_\_OPER=getLevel&loggername={name}" returns the current push level for the specified client side logger (*name*).

- "`__OPER=setLevel&level={level}&loggername={name}`" sets the push level (*level*) for the specified client side logger (*name*), and returns the push level. Valid levels are FINE, FINER, FINEST, INFO, WARNING, and SEVERE.
- "`__OPER=push&loggername={name}`" pushes the queued messages for the specified client side logger (*name*) to the z/OSMF log.

## Logging messages to a popup window

By default, the client side logger routes the JavaScript log statements to the z/OSMF log. You can use the `setpopup` property to route the statements to a popup window in the z/OSMF user interface. To indicate where to display the JavaScript log statements, use the following HTTP requests:

---

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=getpopup&loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?
__OPER=setpopup&loggername={name}
&popup=true|false
```

---

where:

- "`https://{host}:{port}`" specifies the target system address and port.
- "`zosmf/IzuUICommon/UILogManager`" identifies the client logger interface.
- "`__OPER=getpopup&loggername=name`" returns *true* if the specified client side logger (*name*) will append messages to a popup window. Otherwise, *false* is returned.
- "`__OPER=setpopup&loggername=name&popup=true|false`" indicates whether to append messages to a popup window for the specified client side logger (*name*). Set this property to *true* to append messages to a popup window. Otherwise, set this property to *false*.

If you specify *true* for the `setfailnotify` property, messages will also be displayed in a popup window.

## Redirecting messages that cannot be written to the z/OSMF log

If an error occurs that prevents the client's messages from being written to the z/OSMF log, you can use the `setfailnotify` property to indicate whether to redirect those messages to a popup window. This failover action allows for the client data to be retained until the error is resolved. By default, this property is set to *false*.

---

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=getfailnotify&
loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=setfailnotify&
loggername={name}&failnotify=true|false
```

---

where:

- "`https://{host}:{port}`" specifies the target system address and port.
- "`zosmf/IzuUICommon/UILogManager`" identifies the client logger interface.
- "`__OPER=getfailnotify&loggername=name`" returns *true* if the specified client side logger (*name*) will redirect messages to a popup window when an unexpected error occurs. Otherwise, *false* is returned.
- "`__OPER=setfailnotify&loggername=name&failnotify=true|false`" indicates whether to redirect messages to a popup window for the specified client side logger (*name*). Set this property to *true* to redirect messages to a popup window when an error occurs. Otherwise, set this property to *false*.

## Setting the interval for flushing the queue

The client side logger uses the flush interval to determine how often to flush the message queue and send messages to the z/OSMF log. By default, the logger flushes the message queue every 60 seconds. You can use the *setinterval* property to modify this value.

---

```
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=getinterval&
  loggername={name}
GET https://{host}:{port}/zosmf/IzuUICommon/UILogManager?__OPER=setinterval&
  loggername={name}&interval={n}
```

---

where:

- "https://{host}:{port}" specifies the target system address and port.
- "zosmf/IzuUICommon/UILogManager" identifies the client logger interface.
- "\_\_OPER=getinterval&loggername=name" returns the flush interval for the specified client side logger (*name*).
- "\_\_OPER=setinterval&loggername=name&interval=n" sets the flush interval for the specified client side logger (*name*) to the specified number of seconds (*n*). You can specify a minimum of 10 seconds, and a maximum of 3600 seconds.

## Retrieving files and resources for your application

z/OSMF provides a file retrieval service that a client application can use to retrieve the files and resources required for the application to display and function properly.

To request files and resources from the z/OSMF file retrieval service, use the following URL format:

---

```
GET https://{host}:{port}/zosmf/IzuUICommon/externalfiles/{resourcePath}
```

---

where:

- "https://{host}:{port}" specifies the hostname or IP address and the port of the target system.
- "zosmf/IzuUICommon/externalfiles" identifies the z/OSMF file retrieval service.
- "resourcePath" identifies the file to be retrieved. The path must start with the plug-in context root subdirectory, which is specified in the plug-in's property file. For more details about the plug-in context root, see ["Adding your applications to z/OSMF" on page 963](#).

When you issue this HTTP request, the file retrieval service:

- Retrieves the specified file from the UNIX file system.
- Provides the file to z/OSMF core to be displayed.
- Loads the code for the user interface.

Typically, the file retrieval service is used to provide z/OSMF core with the URL to use to launch the application when a user clicks the corresponding task name in the z/OSMF desktop. It can also be used to retrieve additional files and resources as requested by your application.

### Example

The following example retrieves the file *myapp.js* for the *myapp* application.

```
GET /zosmf/IzuUICommon/externalfiles/myappcontextroot/myapp.js
Host: 1.56.82.158:80
```

Figure 460. Sample request to retrieve a file

The HTTP response body depends on the type of file to be retrieved. For the previous example, the expected response follows:

```
HTTP/1.1 200 OK
Date: Thu, 13 Jan 2014 05:39:28 +0000GMT
Content-Type: application/x-javascript

(file content...)
```

Figure 461. Sample response for a request to retrieve a file

## Authoring end user assistance

z/OSMF provides a help system that familiarizes users with the interface, teaches users the concepts required to perform the supported tasks, and helps users troubleshoot errors and transition from one step to another. z/OSMF allows you to add documentation to the help system so you can provide end user assistance that enables users to effectively and easily use your application.

### Overview of the z/OSMF help system

The z/OSMF help system, which is integrated into the software product, contains user assistance for each page, window, message, and action supported in the z/OSMF interface. Users can access the help system by clicking the help link, help button, or message ID link provided in the interface. When clicked, context-sensitive help is displayed within the z/OSMF help system framework, which is depicted in [Figure 462 on page 952](#).

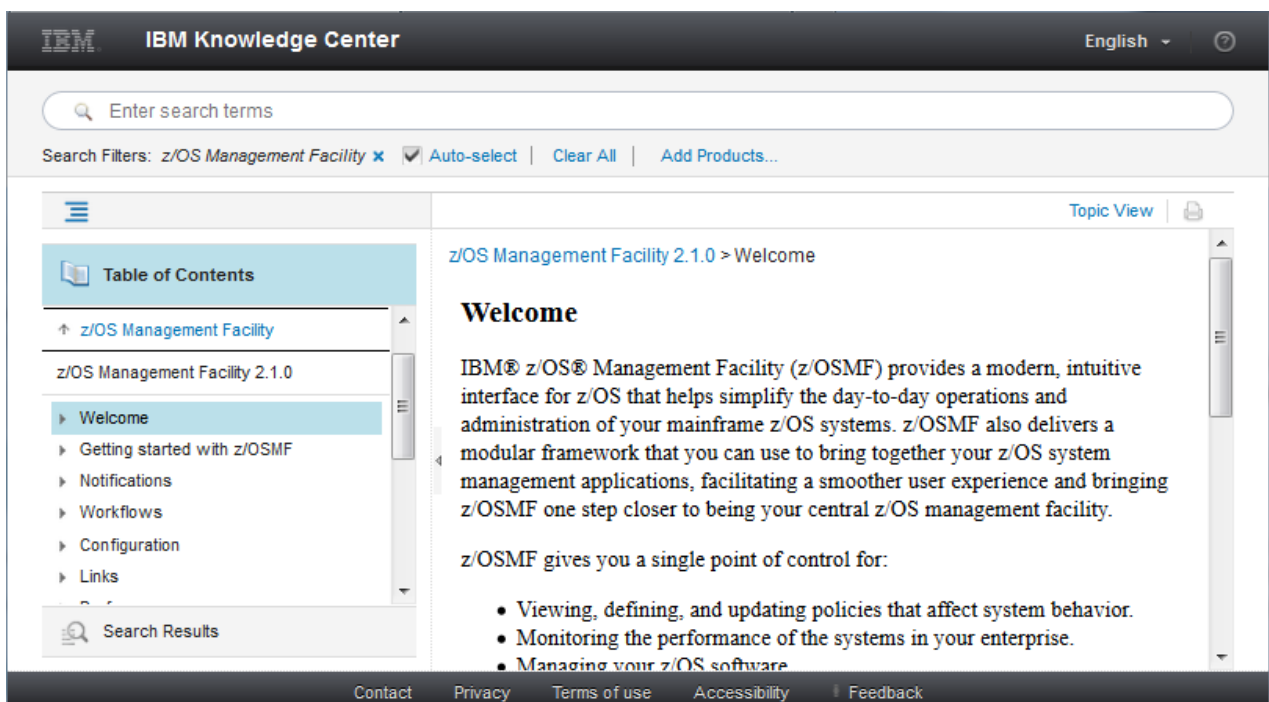


Figure 462. Screen capture of the z/OSMF help system

The framework displays the table of contents and the requested help file. The table of contents mirrors the navigation tree provided in the z/OSMF interface. That is, an application (task) and its help content are included in the same category. For example, the user interface and help content for the Incident Log task are contained in the Problem Determination category.

The help system contains the following additional categories:

- **Getting started with z/OSMF.** This section introduces users to the common features of z/OSMF including using tables and wizards, logging in, and navigating.
- **z/OSMF messages.** This section provides a detailed explanation of each z/OSMF message; describes the reason codes (if any) that are listed in each message; and, suggests actions you can perform to resolve the issue.
- **Tools and techniques for troubleshooting.** This section describes the tools and techniques that are available for troubleshooting problems with z/OSMF.

## Format of the files in the z/OSMF help system

The help files in the z/OSMF help system are stored as help plug-ins. A *help plug-in* is a folder that contains the following content:

- **doc.zip file:** Contains the help files, which are coded using the XHTML tagging language.
- **toc.xml file:** Provides the table of contents for the plug-in in XML format.
- **index.xml file:** Defines the index entries for the plug-in in XML format.
- **plugin.xml file:** Describes the plug-in to the z/OSMF help system using the XML tagging language.
- **nl folder:** Contains language sub-folders, identified by the 2-character language code, for each language into which the help plug-ins are translated. For example, if the help files are translated into Japanese, the *nl* folder will contain a *ja* sub-folder that contains the Japanese version of the content.

To create help plug-ins and add links to your help files, complete the tasks described in the sections that follow.

## Creating help plug-ins

A help plug-in contains the files required to display your application's help files in the z/OSMF help system.

### Procedure

1. To create a panel help plug-in, which is the plug-in that contains the help files for each page, window, and action provided in your application, complete the following steps:
  - a) Create a folder for the panel help that has the name *com.company-name.task-name.help.doc*, where *company-name* is your company's name and *task-name* is the task name that will be displayed in the z/OSMF desktop. For example, *com.ibm.incidentlog.help.doc*.
  - b) Store the folder in the UNIX file system, and set 755 permissions for the folder. For all files stored in this folder, set 644 permissions.
  - c) Create the help content, and code it using the XHTML tagging language. Then, combine the files into a single compressed folder named *doc.zip*. For more details, see [“Developing panel help” on page 954](#).
  - d) Create the table of contents in XML format, and name the file *toc.xml*. For instructions, see [“Creating the table of contents” on page 958](#).
  - e) Assign the plug-in to the same category you plan to use for your application in the z/OSMF desktop. For instructions, see [“Categorizing help plug-ins” on page 960](#).
  - f) Use the XML tagging language to create a file named *plugin.xml*, which describes the plug-in to the z/OSMF help system. For instructions, see [“Identifying the contents of help plug-ins” on page 961](#).

- g) Store the *doc.zip* folder, *toc.xml* file, *index.xml* file, and *plugin.xml* file in the folder you created.
  - h) In the plug-in folder, create an *nl* sub-folder that contains a folder for each language in which the help content is translated, if any. Store the translated files in the correct language folder, which is identified by a 2-character language code. For a list of language codes, see [http://www.loc.gov/standards/iso639-2/php/code\\_list.php](http://www.loc.gov/standards/iso639-2/php/code_list.php).
2. To create a message help plug-in, which is the plug-in that contains the help files for each message your application issues, complete all of the previous steps with the following exceptions:
- Store the resulting *doc.zip* folder, *toc.xml* file, *index.xml* file, *plugin.xml* file, and *nl* folder in a folder named *com.company-name.task-name.message.help.doc*.
  - Assign the message help plug-in to the z/OSMF messages category.
- For information about creating messages and message help, see “[Developing message help](#)” on page 957.

## Results

If you created a panel help plug-in named *com.mycompany.mytask.help.doc* that is translated into Spanish and Japanese, the plug-in will have the following structure:

- com.mycompany.mytask.help.doc
  - doc.zip
  - index.xml
  - plugin.xml
  - toc.xml
  - nl
    - es
      - doc.zip
      - index.xml
      - plugin.xml
      - toc.xml
    - ja
      - doc.zip
      - index.xml
      - plugin.xml
      - toc.xml

## Developing panel help

A panel-help plug-in contains a set of topics, which are independent units of information that are meaningful when displayed alone. This topic-based structure allows you to create context-sensitive help for each page and window displayed in your application. Topic-based content can contain task, concept, or reference information.

These information types are described in the following sections:

- “[Task topics](#)” on page 955
- “[Concept topics](#)” on page 955
- “[Reference topics](#)” on page 956

To provide context-sensitive help for your application, use the XHTML tagging language to create the following topics:

- An introductory topic that provides a brief description of the task and highlights its key features.

- A separate topic for each page or window in the user interface that explains the purpose of the panel and describes the elements on the panel including any fields, columns, actions, or buttons.
- A separate topic for each action that provides step-by-step instructions for performing the action.
- A separate topic for any concept or reference information required for users to effectively operate your application.

## Task topics

A task topic uses a series of steps to explain how to accomplish a goal. Task topics provide procedures, typically in step-by-step instructions. Some task topics might list choices, as bulleted points rather than steps, or they might describe a single action rather than a sequence of steps. Task topics also provide information about the context (where to perform a task and when), the rationale (why perform the task), prerequisites, and examples.

Supertasks (high-level tasks) are the starting points for most users. Steps in a supertask often link to sub-tasks. Each sub-task is documented as a separate task topic. These tasks can be part of one supertask or many. For example, in database programming, opening a database connection is a task that can be reused in several high-level programming tasks.

A task topic documents what users need to know to successfully complete their work. As you write task information, note what concepts need to be documented, and to what depth, if users are to complete the tasks successfully. Consider the probable skills and experience of users, and write with those characteristics in mind. For example, if users need to configure TCP/IP (a task) and might not know about routers, create and link to a concept topic on routers.

Remember these main points when you write task topics:

- Document the steps that users follow to accomplish their goals.
- Use a verb phrase (gerund) as the heading for a task topic.
- Use an opening paragraph to provide context for and introduce the task.
- If the task has a prerequisite, provide that information or provide a link to that prerequisite before the list of steps.
- Use a numbered list for the steps that users must follow to complete the task.
- Write the steps as brief imperative sentences.
- When the user's context changes, introduce the step with a phrase that establishes that new context (for example, On the Configuration page, ...).
- Write one step for each significant user action.
- If a task has more than nine steps, try to divide it into two or more separate tasks.
- If the task is part of a sequence of tasks, provide a link to the next task.

## Concept topics

A concept topic describes a system, solution, product, tool, feature, or background information that users need to complete a task. Users typically read concept material before tackling some large project or starting to use a product or tool. In contrast, users need task or reference topics when they perform a task.

A concept topic describes the scope of the topic and clearly defines what the topic is about. Use minimalist writing techniques to create content that users can quickly understand. Beware of turning a concept topic into a white paper by explaining the whole design philosophy of a product or component. Every concept topic performs at least one of the following functions:

- Introduces a solution, process, product, tool, or feature.
- Provides background information and explains issues that users must know before working with a system or component, or before starting a task.

- Describes the benefits of using one approach rather than another, or provides information about when one particular choice or tool is more appropriate than another.
- Describes how one feature, tool, or product is related to others, and how they work together or do not work together.
- Describes any restrictions that limit the circumstances in which a tool can be used successfully.
- Expands the significance of an important term beyond the scope of a glossary definition.
- Explains how and why some behavior changes as time passes or work progresses.
- Helps users form a mental picture that builds on the experience and knowledge that they are already likely to have.

Remember these main points when you write concept topics:

- Document the background knowledge that users need to successfully use the system, process, product, tool, or feature.
- Use a noun or noun phrase as the heading.
- Use paragraphs.
- If the topic is longer than two screens of information, use subheadings to break it into sections.
- When introducing a new term, begin with a definition; then, expand that definition.
- Add graphics when they simplify the explanation, for example, to show a process or the relationship among concepts.
- Provide examples to bridge from unknown knowledge to known.
- Address only one complete idea.
- Keep the concept topic short and concise, but describe the concept completely.

## Reference topics

Reference topics provide quick access to information that users are likely to need as they complete tasks. Use reference topics to document the purpose of an element, and any restrictions (such as case sensitivity), required authorizations, or anything else that might limit the use of an element.

There are several types of elements for which you can provide reference information. A few examples follow:

- APIs
- Commands
- Language and programming elements
- Class descriptions
- Keyboard shortcuts
- Protocols
- Schemas
- Settings
- Symbols
- Templates
- Column, field, and action descriptions

Remember these main points when you write reference topics:

- Use tables and lists to make reference information easy to scan.
- Use a noun or noun phrase as the title for a reference topic.
- For a particular category of information, such as API documentation, use a consistent format so users can find information quickly.
- Be brief, but write in full sentences.

- Do not go into long explanations; assume that readers understand the basic technology.
- Make the topic as long as it takes to explain the subject.
- Provide links to closely related reference topics, and in some cases, to related concept and task topics.

## Developing message help

A message is any communication that is passed from the application to a user or to another application. z/OSMF uses messages to inform users of important events, such as state changes and errors that require resolution.

For the messages your application issues, use the XHTML tagging language to create a separate help file for each message. Include the following attributes for each message:

- **message ID.** Specifies a unique alphanumeric identifier that provides a quick means to distinguish one message from others. Use the message ID as the filename and topic title.

Vendor message IDs can start with the letters J-Z; the letters A-I are reserved for IBM. The last character indicates the severity of the message, which can have one of the following values:

- **I for information.** Describes information or status for normal conditions and operations.
- **W for warning.** Alerts users to a condition that might cause problems in the future. When a warning message is displayed, users can generally continue with their tasks, but those tasks might not complete in a way that is expected.
- **E for error.** Alerts users to a problem that already occurred. Users or systems cannot continue their tasks.

For example, `YYZR134I`.

- **message text.** Briefly describes the problem or situation from the user's perspective. Messages can have variables (or arguments), which are typically numbers or placeholders that are used in a message in place of a specific file name, command, component, or other object. In the message text, ensure that you:
  - Focus on the problem, not the error.
  - Describe the problem briefly, use full sentences, and ensure that the information is accurate.
  - Avoid wording that seems to blame the user.
  - Do not concatenate multiple messages to create a single message.
  - Make variables meaningful and unique.
  - Replace variables only with proper nouns.
  - Use double quotation marks around variables only when necessary.
- **explanation.** Expands the message text and provides more detail. In the explanation, ensure that you:
  - Explain why the message was issued.
  - For error and warning messages, describe the cause of the problem (when and where the error occurred), explain the consequences of the error, and provide information to help users avoid the problem in the future if possible.
  - Do not repeat the message text in the explanation section.
  - Avoid using codes to build messages and resolve problems in error handling. However, if your application uses codes, describe the codes consistently in your messages and include corrective action so users do not need to look elsewhere for an explanation of the code.
- **userResponse** and **sysprogResponse.** Describes what the user (userResponse) or system programmer and administrator (sysprogResponse) must do to proceed, to recover from the error, or to prevent a problem. If no response is required, enter *No action is required*. For warning and error messages, a response must be provided for the system programmer, the user, or both.

In the response, ensure that you:

- Use active voice when possible.

- Provide complete and specific instructions to resolve the problem.
- Link to other information if necessary.
- Categorize the actions.
- Do not leave the response section empty.
- Ensure that wrapped, associated, and stacked messages are consistently presented.

### **Checklist for writing effective messages**

User perceptions of a software product are strongly influenced by how well messages convey relevant information and help the user solve a problem. Use the following checklist to ensure that messages are clear, accurate, complete, and helpful.

- Write accurate messages:
  - \_\_\_ Ensure that all facts in the message are accurate.
  - \_\_\_ Avoid product names and versions because they change over time.
- Avoid telling users to call a support organization or system administrator:
  - \_\_\_ Check whether there are alternative solutions.
  - \_\_\_ Ask users to check documentation, Web sites, and so on to find solutions.
- Do not blame the user:
  - \_\_\_ Write messages so that they do not appear to blame the user, even if it the cause is a user error.
  - \_\_\_ Avoid "doomsday" phrases such as *catastrophic failure* or *fatal error*.
- Use clear language:
  - \_\_\_ Check for ungrammatical or incorrectly punctuated sentences.
  - \_\_\_ Avoid garbled, long (over 25 words), or convoluted sentences.
  - \_\_\_ Check for unnecessary passive voice.
  - \_\_\_ Avoid abstract language, general language, and jargon.
  - \_\_\_ Use a consistent style and word usage.
  - \_\_\_ Always use full sentences with correct punctuation.
  - \_\_\_ Use nouns after command and API names, for example, the BaseException class. (class is the noun.)
  - \_\_\_ Do not leave out articles such as *a*, *an*, or *the*.
- Provide complete information:
  - \_\_\_ Ensure that each message is needed in the system. Do not display a message when the code should handle the issue.
  - \_\_\_ Provide all the instructions for resolving a problem in one message whenever possible.
  - \_\_\_ If the user actions are too long or topics for the appropriate corrective actions exist elsewhere, provide a link from the message to other appropriate topics.
  - \_\_\_ Add explanation, user response, and system programmer response sections to all messages.
  - \_\_\_ Do not give simplistic user actions such as "See the log files" without more guidance. For example, if you ask users to see log files, tell them where to look in the directory structure to find the log and what to look for in the log.
  - \_\_\_ Provide examples for commands, APIs, or other code unless the code is extensive or complicated.
  - \_\_\_ Do not expose unnecessary information such as documenting system actions that do not affect the user, providing information that is too detailed for the target audience, and describing internal workings that the user has no control over. This information should be logged only in a trace for support.

### **Creating the table of contents**

A table of contents is required to integrate each help plug-in into the z/OSMF help system. Providing a table of contents improves the navigability of help plug-ins and ultimately makes it easier for users to find

relevant help information. This topic describes the structure of the table of contents for panel and message help plug-ins.

## Panel help plug-ins

To create a table of contents for panel help plug-ins, create an XML file that has the following structure:

```
<toc label="task-name" link_to="category" topic="path-to-parent-topic">
  <topic label="task-name" href="path-to-parent-topic">
    <topic label="topic-name" href="path-to-topic"/>
    <topic label="topic-name" href="path-to-topic">
      <topic label="topic-name" href="path-to-topic"/>
      <topic label="topic-name" href="path-to-topic"/>
      <topic label="topic-name" href="path-to-topic"/>
    </topic>
    <topic label="topic-name" href="path-to-topic"/>
  </topic>
</toc>
```

Figure 463. Table of contents template for panel help plug-ins

where:

### task-name

Name that will be displayed in the z/OSMF navigation area for your application followed by the word *task*.

### category

Path to the category that will contain the help for the task. For more details, see [“Categorizing help plug-ins”](#) on page 960.

### path-to-parent-topic

Path to and name of the file that introduces the task and its key features. The introductory topic must be the container (parent) for all the other help topics in the plug-in.

### topic-name

Name of the topic, which is the label that will be displayed in the table of contents in the z/OSMF help system.

### path-to-topic

Path to and name of the file that contains the help content.

Figure 464 on page 959 provides a sample table of contents for the System Status task.

```
<toc label="System Status task"
link_to="../com.ibm.zosmfcore.performance.help.doc/
izuG00hpPerformance.xml#sysplex"
topic="izuR00hpSysplexStatusTask.html">
  <topic label="System Status task" href="izuR00hpSysplexStatusTask.html">
    <topic label="Managing system resources" href="izuR00hpSysplexStatusPanel.html">
      <topic label="Adding resource entries" href="izuR00hpAddSysEntry.html">
        <topic label="Add and Modify Entry pages" href="izuR00hpAddModSysPanel.html"/>
      </topic>
      <topic label="Modifying resource entries" href="izuR00hpModSysEntry.html"/>
      <topic label="Removing resource entries" href="izuR00hpRemSysEntry.html"/>
    </topic>
  </topic>
</toc>
```

Figure 464. Sample table of contents for the System Status task

## Message help plug-ins

To create a table of contents for message help plug-ins, create an XML file that has the following structure:

```
<toc label="message-ID-range" link_to="category" topic="path-to-parent-topic">
  <topic label="message-ID-range" href="path-to-parent-topic">
    <topic label="message-ID" href="path-to-topic">
    <topic label="message-ID" href="path-to-topic"/>
    <topic label="message-ID" href="path-to-topic"/>
    <topic label="message-ID" href="path-to-topic"/>
  </topic>
</toc>
```

Figure 465. Table of contents template for message help plug-ins

where:

### message-ID-range

Range of message IDs included in the message help plug-in.

### category

Path to the z/OSMF messages category, which will contain the message help for the task. For more details, see [“Categorizing help plug-ins”](#) on page 960.

### path-to-parent-topic

Path to and name of the file that is the container (parent) for all of the message help topics. The parent topic should state the following: *This topic describes the z/OSMF messages that have a message ID between message-ID-range.*

### topic-name

Name of the topic, which is the ID of the message.

### path-to-topic

Path to and name of the file that contains user assistance for the message.

Figure 466 on page 960 provides a sample table of contents for messages issued by z/OSMF core.

```
<toc label="IZUG0400-IZUG9999"
link_to="./com.ibm.zosmfmessages.help.doc/
izuAllhpzOSMFMessages.xml#core_messages"
topic="IZUG0400-IZUG9999.html">
  <topic label="IZUG0400-IZUG9999" href="IZUG0400-IZUG9999.html">
    <topic label="IZUG400I" href="IZUG400I.html"/>
    <topic label="IZUG401E" href="IZUG401E.html"/>
    <topic label="IZUG401W" href="IZUG401W.html"/>
    <topic label="IZUG402W" href="IZUG402W.html"/>
    <topic label="IZUG403E" href="IZUG403E.html"/>
    <topic label="IZUG404I" href="IZUG404I.html"/>
  </topic>
</toc>
```

Figure 466. Sample table of contents for the messages issued by z/OSMF core

When the message help plug-in is displayed within the z/OSMF help system, it is listed under the z/OSMF messages category and each message topic is nested under the parent topic.

## Categorizing help plug-ins

In the z/OSMF help system, a task and its panel help plug-in are included in the same category, and the task's message help plug-in is listed under the z/OSMF messages category, where all z/OSMF messages

are listed. This structure ensures consistency across z/OSMF tasks, making it easier for users to find relevant content.

To categorize the panel and message help plug-ins, on the <toc> element in the *toc.xml* file, specify the appropriate URL for the *link\_to* attribute. Table 483 on page 961 lists the URL to use for each z/OSMF category. For example, to add the message help plug-in to the z/OSMF messages category, type the following value: *link\_to="./com.ibm.zosmfmessages.help.doc/izuAllhpzOSMFMessages.xml#messages"*.

Table 483. URL to use for each z/OSMF category	
Category	URL
Commands and Logs	./com.ibm.zosmfcore.commandlog.help.doc/izuG00hpCommandLogs.xml#commands_and_logs
Configuration	./com.ibm.zosmfcore.configuration.help.doc/izuG00hpzOSMFConfiguration.xml#configuration
Jobs and Resources	./com.ibm.zosmfcore.jobresources.help.doc/izuG00hpJobResources.xml#jobs_and_resources
Links	./com.ibm.zosmfcore.linksuser.help.doc/izuG00hpLinksUser.xml#links
Performance	./com.ibm.zosmfcore.performance.help.doc/izuG00hpPerformance.xml#performance
Problem Determination	./com.ibm.zosmfcore.problemdetermination.help.doc/izuAllhpzOSMFProbDet.xml#problem_determination
Software	./com.ibm.zosmfcore.software.help.doc/izuG00hpSoftware.xml#software
z/OS Classic Interfaces	./com.ibm.zosmfcore.classicinterfaces.help.doc/izuG00hpClassicInterfaces.xml#classic_interfaces
z/OSMF Administration	./com.ibm.zosmfcore.administration.help.doc/izuG00hpzOSMFAdministration.xml#administration
z/OSMF messages	./com.ibm.zosmfmessages.help.doc/izuAllhpzOSMFMessages.xml#messages
z/OSMF Settings	./com.ibm.zosmfcore.settings.help.doc/izuG00hpSettings.xml#settings

## Identifying the contents of help plug-ins

Every plug-in requires a file called *plugin.xml* to identify the plug-in contents to the z/OSMF help system.

### About this task

This file includes the following items:

- The table of contents file that contributes to the navigation for the help plug-in.
- The name, ID, and the version of the help plug-in.
- The index file for building the Index view.

### Procedure

To create the *plugin.xml* file for your help plug-ins, follow these steps:

1. Create a new Extensible Markup Language (XML) file called *plugin.xml*.
2. Make the following XML declaration the first two lines of the file:

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
```

3. Add the root <plugin> element and specify the following attributes for the help plug-in:

#### name

For the panel help plug-in, the name that will be displayed in the z/OSMF desktop for your application, followed by the word *task*.

For the message help plug-in, the range of message IDs included in the plug-in.

#### id

Name of the folder that contains the panel or message help contents.

#### version

Version of the help plug-in.

#### provider-name

Name of your company.

For example, the plug-in identification metadata for the Incident Log task can be as follows:

```
<plugin name = "Incident Log task"
        id = "com.ibm.zosmfincidentlog.help.doc"
        version ="2.1"
        provider-name = "IBM">
```

4. Within the context of the <plugin> element, create an <extension> element with a *point* attribute value of *org.eclipse.help.toc*. Then, within the context of the <extension> element, create a <toc> element to declare the table of contents file.

```
<extension point="org.eclipse.help.toc">
  <toc file="toc.xml" primary="true" />
</extension>
```

5. If you indexed your content, within the context of the <plugin> element, create an <extension> element with a *point* attribute value of *org.eclipse.help.index*. Then, within the context of the <extension> element, create an <index> element to declare the index file.

```
<extension point="org.eclipse.help.index">
  <index file="index.xml" />
</extension>
```

6. Close the <plugin> element.

```
</plugin>
```

### Example

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>

<plugin name = "Incident Log task"
        id = "com.ibm.zosmfincidentlog.help.doc"
        version ="2.1"
        vendor-name = "IBM">

  <extension point="org.eclipse.help.toc">
    <toc file="toc.xml" primary="true" />
  </extension>

  <extension point="org.eclipse.help.index">
    <index file="index.xml" />
  </extension>

</plugin>
```

## Adding links to help plug-ins

z/OSMF tasks provide a help link, help button, or message ID link on every page, window, or message that is provided in the user interface. When users click the corresponding widget, user assistance is displayed for the current context.

### Procedure

To provide context-sensitive help for your applications, complete the steps that follow:

1. Create a help topic for each page, window, and message that can be displayed in your applications. For instructions, see [“Creating help plug-ins” on page 953](#).
2. Add a help link, help button, message ID link, or another widget to each page, window, and message.
3. For each widget, specify the URL for the related help topic. The URL must have the following format:

```
https://host:port/context-root/helps/release-id/help-plugin-name/help-topic
```

where:

**host**

Hostname or IP address of the system where z/OSMF is installed.

**port**

Secure application port for the z/OSMF configuration. If you specified a secure port for SSL encrypted traffic during the configuration process (through the IZUPRMxx parmlib member keyword HTTP\_SSL\_PORT), specify the same port in the URL. If you omit the port, it is assumed that you are using port 443, the default.

**context-root**

Context root of the z/OSMF application. By default, the context root is zosmf.

**release-id**

Value that identifies the release, for example SSB2H8\_2.4.0.

**help-plugin-name**

Name of the help plug-in that contains the help topic to which you want to link.

**help-topic**

Filename of the help topic to which you want to link.

**Example**

Figure 467 on page 963 provides sample code for linking a button to a specific help topic in the z/OSMF help system.

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
  <script type="text/javascript">
    function open_help()
    {
      window.open(
        "https://abc.com:81/zosmf/helps/SSB2H8_2.4.0/com.ibm.task.help.doc/
about.html", "help"
      );
    }
  </script>
</head>
<body>
  <input value="Help" onclick="open_help()" type="button">
</body>
</html>
```

Figure 467. Sample code for linking to a help topic

## Adding your applications to z/OSMF

To add your application to z/OSMF, create a property file that defines the parameters required for z/OSMF to configure your application, and use the z/OSMF Import Manager task to import the property file.

**Before you begin**

- Develop a web-based application and the supporting documentation for the functions you want to add to z/OSMF. For instructions, see [“Developing web-based applications”](#) on page 924 and [“Authoring end user assistance”](#) on page 952.
- Identify and resolve any security vulnerabilities in your application. For more details, see [“Verifying the security of applications”](#) on page 969.

## About this task

A property file is a flat file, such as a text file, that contains a set of attributes for one or more instances of an object. The attributes are specified as name and value pairs, and must be enumerated for each instance of the object. z/OSMF supports property files that are encoded using the platform default encoding, which by default is EBCDIC 1047, or ASCII.

When you use a property file to import a plug-in into z/OSMF, the z/OSMF server performs the following actions:

- Validates the structure of the property file, and verifies that a value is provided for the required properties.
- Creates symbolic links to the client-side code for your application so that any updates to that code are reflected in z/OSMF.
- Stores the symbolic links in the plug-in context root subdirectory, which is specified in the plug-in property file.
- Adds a task that is defined in the property file to the z/OSMF desktop, if a URL is specified for its `taskNavigationURL` property.

## Procedure

To use a property file to add a plug-in to z/OSMF, complete the following steps:

1. Create a property file in the z/OS UNIX System Services (z/OS UNIX) file system, and set 644 as its permissions.  
For example, you can create a text file called *plugin.properties*.
2. Type `importType=plugin` at the beginning of the file. Doing so informs z/OSMF that the property file contains a plug-in definition. You cannot define other types of objects, such as event types or handlers, in a property file for a plug-in.
3. Define only one plug-in to be added to z/OSMF. To do so, specify the following properties for the new plug-in:

```
izu.externalapp.file.version=plugin-version
izu.externalapp.local.context.root=plugin-context-root
izu.externalapp.code.root=plugin-code-root

pluginId=plugin-Id
pluginDefaultName=plugin-name
pluginDescription=plugin-description
aboutPanelPath=about-panel-path

izu.externalapp.help.root=help-root
izu.externalapp.helpdoc=help-plugin-name
```

### plugin-version

Specify the version of the plug-in.

### plugin-context-root

Specify the name of the symbolic link directory under the `USERDIR/data/externalapps` directory to use for your applications. The `USERDIR` setting, which identifies the mount point of the z/OSMF user file system, is set when your installation configures z/OSMF. By default, the mount point is `/global/zosmf/`.

If you specify `myapp` as the plug-in context root, a symbolic link that is named `myapp` is created in the path of `USERDIR/data/externalapps/myapp`, and it points to the source code path `/usr/lpp/myapp/ui`.

z/OSMF uses the context root to build the navigation URL and the bundle URL.

The plug-in context root is required, and the directory name must comply with the z/OS UNIX naming guidelines.

**plugin-code-root**

Specify the path to the z/OS UNIX directory that contains the source code for the tasks that are included in the plug-in. If the plug-in property file and the source code directory reside in the same directory, you can specify a relative path, for example, `ui`. Otherwise, you must specify the absolute path, for example, `/usr/lpp/myapp/ui`. The plug-in code root is required.

**plugin-ID**

Specify a unique identifier for the plug-in. The ID is required, must be unique, and can contain a maximum of 64 characters.

**plugin-name**

Specify the name of the plug-in. The plug-in name is required, and can contain a maximum of 64 characters.

**plugin-description**

Provide a description of the plug-in. The description is required, and can contain a maximum of 256 characters.

**about-panel-path**

Specify the absolute path of the flat file that contains the plug-in version information to display on the z/OSMF **About** page. For example, `/zosmf/IzuUICommon/externalfiles/myapp/myappVersion.txt`. The path is required.

**Tip:** If the file resides in the same directory as the plug-in property file, you can specify the file name with no path information. For example, `myappVersion.txt`.

**help-root**

Specify the path to the z/OS UNIX directory that contains the help files for the tasks that are included in the plug-in. If the plug-in property file and the help root directory reside in the same directory, you can specify a relative path, for example, `helps`. Otherwise, you must specify the absolute path, for example, `/usr/lpp/myapp/helps`. The plug-in help root is optional.

**help-plugin-name**

Specify the name of the directory that contains the help files for the tasks in your application. For example, `com.ibm.zosmfmyapp.task1.help.doc`. The directory name is required only if you are including help files in your plug-in. The directory name must comply with the z/OS UNIX naming guidelines.

Each help plug-in must be enumerated and must be listed in numerical order. z/OSMF expects the first help plug-in to be enumerated as *izu.externalapp.help.root.1*, the second as *izu.externalapp.help.root.2*, and so on. The first time the enumeration does not match the position of the help plug-in in the file, z/OSMF stops looking for help plug-ins.

4. Include one to 32 task definitions in the plug-in property file. Each task definition must be enumerated and must be listed in numerical order because z/OSMF expects the first task definition to be enumerated as one, the second definition as two, and so on. The first time the enumeration does not match the position of the task definition in the file, z/OSMF stops reading the file. The remaining task definitions are not processed.

The following attributes are supported for tasks:

```
taskId=task-ID
taskVersion=task-version
taskCategoryId=category-ID
taskDispName=task-name
taskDispDesc=task-description
taskSAFResourceName=task-SAF-resource-name
taskMultiSysplexScope=task-multi-sysplex-scope
taskHandlerEligible=task-handler-eligible
taskAuthenticatedGuestEligible=task-authenticated-guest-eligible
taskNavigationURL=task-navigation-URL
taskBundleUrl=bundle-URL
taskBundleFileName=bundle-file-name
taskMinZOS=minimum-z/OS-level
taskMinZOSMF=minimum-z/OSMF-level
```

**task-ID**

Specify a unique identifier for the task. The ID is required, must be unique, and can contain a maximum of 64 characters.

**task-version**

Specify the version of the task.

**category-ID**

This value is ignored.

**task-name**

Specify the task name to be displayed in the z/OSMF desktop. The task name is required, and can contain a maximum of 30 characters.

**task-description**

Provide a description of the task. The description is required, and can contain a maximum of 200 characters.

**task-SAF-resource-name**

Specify a unique SAF resource name to be used for managing user authorizations to the task. The resource name must start with ZOSMF, and must conform to the following structure to ensure uniqueness:

```
ZOSMF.<vendor>_<plugin-ID>.<task-ID>.<task-name>
```

Where:

**vendor**

Name of your company.

**plugin-ID**

Unique identifier that you assigned to the plug-in.

**task-ID**

Unique identifier that you assigned to the task.

**task-name**

Name that you assigned to the task.

For example, ZOSMF . IBM\_TESTPLUGIN . COMMANDS . Commands.

If the SAF resource name does not begin with ZOSMF, z/OSMF adds prefix ZOSMF.IMPORT to the SAF resource name.

The resource name is required, and can contain up to 231 alphanumeric characters (A-Z a-z 0-9) and the following special characters: Underscore (\_), dash (-), period (.). The use of a period in a resource name is treated as a qualifier. As such, the first character after a period must be A-Z or a-z.

**task-multi-sysplex-scope**

z/OSMF provides a multi-sysplex capability, which allows you to manage multiple z/OS sysplexes from a single z/OSMF instance. To do so, a z/OSMF instance must be running in each sysplex to be managed.

Set the *task-multi-sysplex-scope* property to `true` to indicate that this task can be used to manage or display data for multiple z/OS sysplexes. Otherwise, omit this property or set it to `false`.

**task-handler-eligible**

z/OSMF provides the application linking capability, which allows you to create context-sensitive launch points between tasks or applications. The task or application that initiates the launch request is referred to as the event requester. The task or application that processes the request and displays the appropriate context is referred to as the event handler.

Set the *task-handler-eligible* property to `true` to indicate that the task is eligible to be an event handler in the application linking process. Doing so allows users to define the task as a handler for one or more event types. To disallow the task to participate in the application linking process as an event handler, omit this property or set it to `false`.

**task-authenticated-guest-eligible**

Set this property to `true` to extend task authorization to users who are logged in to z/OSMF, but are not defined to a z/OSMF SAF security group. Otherwise, omit this property or set it to `false`. Extending task authorization to users who are not logged in to z/OSMF is not supported.

**task-navigation-URL**

Specify the relative or absolute path of the home page for the task. For example, if the home page is named `index.html` and resides in the `myapp` directory (context root), you can specify `index.html` or `/zosmf/IzuUICommon/externalfiles/myapp/index.html`.

If the plug-in contains more than one task, append the following to the path: `?`

`task=contextRoot.taskName`, where `contextRoot` and `taskName` are the values you specified for the `izu.externalapp.local.context.root` property and the `taskDispName` property. For example, `/zosmf/IzuUICommon/externalfiles/myapp/index.html?`

`myapp.settings.task`.

The path can contain a maximum of 4000 characters, including alphanumeric characters (A-Z a-z 0-9), blanks, mathematical symbols (+ - = | ~ ( ) { } \), punctuation marks (? , . ! ; : ' " / [ ]), and the following special characters: %, \$, #, @, ^, \*, and \_. Any leading or trailing white space is ignored.

The navigation URL is required if the bundle URL is specified. Otherwise, omit the navigation URL.

**bundle-URL**

Specify the relative or absolute path of the language resource bundle for the task. For example, if the bundle file resides in the `/myapp/js` directory, you can specify `/js` or `/zosmf/IzuUICommon/externalfiles/myapp/js`.

The path can contain a maximum of 256 characters, including alphanumeric characters (A-Z a-z 0-9), blanks, mathematical symbols (+ - = | ~ ( ) { } \), punctuation marks (? , . ! ; : ' " / [ ]), and the following special characters: %, \$, #, @, ^, \*, and \_. Any leading or trailing white space is ignored.

The bundle URL is required if the navigation URL is specified. Otherwise, omit the bundle URL.

**bundle-file-name**

Specify the name of the language resource bundle file. The file name can contain a maximum of 256 characters. For example, `bundle.js`. The file name is required if the bundle URL is specified.

**minimum-z/OS-level**

Specify the minimum z/OS operating system level that the task requires. You can specify one of the following values:

- **04.27.00:** Indicates that the minimum z/OS level is V2R4.
- **04.26.00:** Indicates that the minimum z/OS level is V2R3.
- **04.25.00:** Indicates that the minimum z/OS level is V2R2.

**minimum-z/OSMF-level**

Specify the minimum z/OSMF level that the task requires. You can specify one of the following values:

- **04.27.00:** Indicates that the minimum z/OSMF level is V2R4.
- **04.26.00:** Indicates that the minimum z/OSMF level is V2R3.
- **04.25.00:** Indicates that the minimum z/OSMF level is V2R2.

Figure 468 on page 968 provides a sample property file that defines the *myapp* plug-in, which contains a Commands task and a Settings task.

```

importType=plugin

izu.externalapp.file.version=1.0.0
izu.externalapp.local.context.root=myapp
izu.externalapp.code.root=ui
pluginId=com.ibm.zosmf.myapp
pluginDefaultName=myapp
pluginDescription=Operate myapp.
aboutPanelPath=/zosmf/IzuUICommon/externalfiles/myapp/myappVersion.txt

izu.externalapp.help.root=helps
izu.externalapp.helpdoc.1=com.ibm.zosmfmyapp.commands.help.doc
izu.externalapp.helpdoc.2=com.ibm.zosmfmyapp.settings.help.doc

taskId1=COMMANDS
taskVersion1=1.0
taskCategoryId1=12
taskDispName1=Commands
taskDispDesc1=The Commands task lets you enter z/OS commands.
taskSAFResourceName1=ZOSMF.IBM_COMMANDS.COMMANDS.Commands
taskMultiSysplexScope1=true
taskHandlerEligible1=true
taskAuthenticatedGuestEligible1=true
taskNavigationURL1=/zosmf/IzuUICommon/externalfiles/myapp/index.html?
task=myapp.commands
taskBundleUrl1=/zosmf/IzuUICommon/externalfiles/myapp/js/nls
taskBundleFileName1=bundle.js
taskMinZOS1=04.27.00
taskMinZOSMF1=04.27.00

taskId2=SETTINGS
taskVersion2=1.0
taskCategoryId2=10
taskDispName2=Settings
taskDispDesc2=The Settings task allows you to define task-specific settings.
taskSAFResourceName2=ZOSMF.IBM_SETTINGS.SETTINGS.Settings
taskHandlerEligible2=true
taskAuthenticatedGuestEligible2=true
taskNavigationURL2=/zosmf/IzuUICommon/externalfiles/myapp/index.html?
task=myapp.settings
taskBundleUrl2=/zosmf/IzuUICommon/externalfiles/myapp/js/nls
taskBundleFileName2=bundle.js
taskMinZOS2=04.27.00
taskMinZOSMF2=04.27.00

```

Figure 468. Sample plug-in property file

**Tip:** To remove a plug-in and all of its tasks from z/OSMF, type *deletePlugin=true* at the end of the property file. Otherwise, omit this property.

5. Save the property file.
6. Optionally, you can add a customized icon for each new task.

By default, z/OSMF uses a "puzzle piece" icon to represent a task in the z/OSMF desktop. However, you can supply a customized icon for the task by following these steps:

- a. Create the icon as a Portable Network Graphics (PNG) file. For consistency with the other z/OSMF desktop icons, set the file dimensions to 470x470 pixels.
- b. Save the icon image file by using the following naming convention: *task-name.png*
- c. Store the icon image file in the same directory path as the main page for the task URL. For example, if the task name is myapp and the task main page is `/zosmf/IzuUICommon/externalfiles/myapp/index.html`, store the file as follows:

```
/zosmf/IzuUICommon/externalfiles/myapp/myapp.png
```

7. Import the property file. To do so, complete the following steps:

- a) In the z/OSMF desktop, select **Import Manager**.
- b) On the **Import** tab on the **Import Manager** page, specify the full path and name of the property file you created, and click **Import**.

## Results

A message is displayed to indicate whether the plug-in was added. If so, the tasks where you provided a URL for the taskNavigationURL property are displayed in the z/OSMF desktop.

If z/OSMF finds any errors in the property file, the plug-in and its tasks are not added to z/OSMF. If so, you must resolve the errors and import the property file again.

## What to do next

If the server-side code for your plug-in does not reside on the z/OSMF server, you must associate an application server with each task included in the plug-in. Otherwise, the plug-in will not work correctly. For instructions, see the topic about associating application servers with imported tasks in the z/OSMF help system.

Set up security for your plug-in. After which, you must refresh the security management product on your system and restart the z/OSMF server to have your changes take affect. For more details, see the section about securing your applications in [“Securing your applications” on page 969](#).

## Securing your applications

---

To secure your applications, identify and resolve any security vulnerabilities, and work with your security administrator to grant users access to your applications. When the required security controls are established on your system, a user can begin using z/OSMF to perform system management tasks.

### Verifying the security of applications

Before importing applications into z/OSMF, ensure that the vendor or developer who supplied the application adhered to security best practices for Web applications. If the software installed is not secure, it is possible to expose your system or company to security issues.

### Controlling access to applications

After importing your plug-in into z/OSMF, work with your security administrator to authorize users to your applications. z/OSMF security is based on the following concepts:

#### user authentication

When a user attempts to log in to z/OSMF through a web browser, the user’s credentials are verified by the z/OS host system through the SAF interface or a security management product (for example, RACF). This processing ensures that the user ID is known to the z/OS system, and the password is valid.

#### user authorization

Access to your application is controlled through SAF resource profile <safPrefix>.<taskSAFResourceName>, where <safPrefix> is configured in z/OSMF and is by default IZUDFLT and <taskSAFResourceName> is the SAF resource name you specified for the task in the plug-in property file. The SAF resource profile is defined in the ZMFAPLA class.

If your installation is using RACF and you want to assign administrators CONTROL access and users READ access to your application, you can create a profile like the following:

```
RDEFINE ZMFAPLA +
  (IZUDFLT.ZOSMF.IBM_COMMANDS.COMMANDS.Commands) UACC(NONE)
PERMIT +
  IZUDFLT.ZOSMF.IBM_COMMANDS.COMMANDS.Commands +
  CLASS(ZMFAPLA) ID(IZUADMIN) ACCESS(CONTROL)
PERMIT +
```

```
IZUDFLT.ZOSMF.IBM_COMMANDS.COMMANDS.Commands +  
CLASS(ZMFAPLA) ID(IZUUSER) ACCESS(READ)
```

z/OSMF automatically manages the authorization of non-authenticated guests (not logged in) and authenticated guests (logged in, but are not defined to a z/OSMF SAF security group). By default, a non-authenticated guest user can access the z/OSMF Welcome task and access the default links. An authenticated guest can access everything a non-authenticated guest can, and also view the online help.

To authorize authenticated guest users to your task, in the plug-in property file, set the *task-authenticated-guest-eligible* property to *true*. Extending task authorization to users who are not logged into z/OSMF is not supported.

## Actions for security update

Changes to your security setup require applicable refreshes of the security product and a restart of the z/OSMF server for them to take effect.

---

## Chapter 4. Preparing software to exploit cloud provisioning

This topic describes how software providers can prepare software to exploit IBM Cloud Provisioning and Management for z/OS.

IBM Cloud Provisioning and Management for z/OS allows software consumers, from a selection of software services, to quickly provision and deprovision software as needed. For more information, see [“Cloud provisioning services” on page 45](#) and the online help for the Cloud Provisioning tasks in z/OSMF.

Software is provisioned from a software services template. A software services template requires the following files.

### Workflow definition file

A workflow definition file is the primary XML file that defines the workflow that performs the provisioning. The workflow definition file includes information about the workflow, such as name and version, as well as step and variable definitions.

The provisioning workflow definition file must be located in a z/OS UNIX file. File templates (specified with the fileTemplate element) that are referenced by a provisioning workflow, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set.

For information about workflow definition files, see [Chapter 2, “Creating workflow definitions for z/OS,” on page 803](#).

### Variable input file

A workflow variable input file is a properties file that is used to specify values for one or more of the input variables that are defined in the workflow definition. For information, see [Chapter 2, “Creating workflow definitions for z/OS,” on page 803](#).

### Action definition file

An action definition file describes the actions that can be performed against the provisioned software, which is known as a software services instance.

The actions definition file can be located in a z/OS UNIX file or a z/OS data set. File templates (specified with the fileTemplate element) that are referenced by a workflow action, and any corresponding callable workflows, can be located in a z/OS UNIX file system or a data set.

For more information, see [“Actions definition file” on page 973](#).

### Documentation files

A documentation file is an optional text or PDF file that provides information that is important to provisioning the software. For example, it might describe the workflow and other files, and describe requirements for using them to provision software. There can be one documentation file for administrators, who create the software services template and prepare the software for provisioning, and one for consumers, who use the published software services template to provision the software. The document for administrators might indicate, for example, whether a network resource pool or WLM resource pool is required.

### Manifest, or template source file

A manifest file is optional. It provides a shortcut when a user creates the software services template by using the z/OSMF Software Services task. Rather than specifying each of the files (workflow definition,

input variable file, action, and documentation) individually, the user can specify just the manifest file, then click **Load** to supply values for the other files.

The file must be in Java property file format:

- Each entry is a single line, in *property=value* or *property:value* format.
- The \ character is a continuation character so that a value can span lines.
- For newline, carriage return, and tab, use \n, \r, and \t.
- Comment characters are ! and #. Lines that start with those characters are ignored.

The fields in the manifest file are:

**workflow-definition-file**

Name of the workflow definition file

**workflow-variable-input-file**

Name of the workflow variable input file

**action-definition-file**

Name of the action definition file

**description**

Brief description of the workflow. This is optional.

**admin-documentation-file**

Name of the file that describes the workflow and other files. This file is intended for an administrator who prepares a software services template that consumers can use to provision the software. This is optional.

**admin-documentation-type**

File type of the administrator documentation file: text or pdf. This is optional, and valid only if admin-documentation-file is specified.

**cconsumer-documentation-file**

Name of the file that describes the workflow and other files, intended for a consumer who uses the software services template to provision the software. This is optional.

**cconsumer-documentation-type**

File type of the consumer documentation file: text or pdf. This is optional, and valid only if consumer-documentation-file is specified.

You can specify relative or absolute paths for the files that you identify in the manifest file, as follows.

- If the manifest file is a z/OS UNIX file, specify:
  - z/OS UNIX files with a full path, for example, /a/b/c/d.xml, or a relative path, for example, ../b/c/d.xml.
  - Data sets with // followed by a fully qualified name. Data sets can be either partitioned or sequential. For example, you might specify //IBMUSER.DS.PDS(XML) or //IBMUSER.DS.SEQ.
- If the manifest file is in a data set, specify:
  - z/OS UNIX files with a full path, for example, /a/b/c/d.xml.
  - Data sets with fully qualified or relative names, as follows.
    - Fully qualified names follow // and can be either partitioned or sequential. For example, you might specify //IBMUSER.DS.PDS(XML) or //IBMUSER.DS.SEQ.
    - Relative names vary with the type of the manifest file data set, as follows:
      - Partitioned: Specify just the member, which identifies a member in the manifest file data set. For example, if the manifest file is IBMUSER.DS.PDS(MF), specifying a file path of XML in the manifest file requests IBMUSER.DS.PDS(XML).
      - Sequential: Specify one or more qualifiers that are added to the manifest file data set name. For example, if the manifest file is IBMUSER.DS.SEQ, specifying a file path of XML in the manifest file requests IBMUSER.DS.SEQ.XML.

The following is an example of a manifest file.

```
# provision.mf
#
# Manifest file to be used when adding a template for provisioning an MQ for z/OS Queue Manager.
#
# <copyright
#   notice="lm-source"
#   pids="@PID###@"
#   years="2015,2016"
#   crc="3073404564">
#   Licensed Materials - Property of IBM
#
#   @PID###@
#
#   (C) Copyright IBM Corp. 2015, 2016 All Rights Reserved.
# </copyright>
#
# Provision Queue Manager workflow file (steps to provision a Queue Manager)
workflow-definition-file:provision.xml
# Provision Queue Manager workflow variables properties file (properties to be used when provisioning
a Queue Manager)
workflow-variable-input-file:workflow_variables.properties
# Queue Manager actions file (defines the actions that can be performed against a Queue Manager)
action-definition-file:qmgrActions.xml
# Provision Queue Manager workflow description
description:This workflow provisions an MQ for z/OS Queue Manager
# Provision Queue Manager readme file
admin-documentation-file:mqaas_readme.pdf
# Provision Queue Manager readme file type
admin-documentation-type:pdf
```

## Actions definition file

An actions definition file has XML syntax and conforms to the rules of the actions schema. The schema defines the required and optional properties (XML elements and attributes). It imposes constraints on the order in which the elements are specified, and on the values that can be specified for each element and attribute.

The schema file is UTF-8 encoded.

If you are developing an actions definition file, you require access to the schema, and therefore access to the z/OS system on which z/OSMF is installed.

The primary XML file must start with a processing instruction (in column 1 of line 1) for the XML processor. This instruction defines the version of XML used and the encoding of the file. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The remaining elements are as follows.

### <actionList>

Is the root element. It contains the actions definitions.

### <action>

Contains an action definition. There must be 1 - 50 actions in an actions definition file. The action contains either a command, workflow, or instruction element. The attributes are:

- name, which specifies the name of the action
- deprovision, which accepts true or false, to indicate whether the action is for deprovisioning. There must be at least one deprovision action.

### <command>

Contains a command definition. It contains the following elements.

### <commandValue>

Command to be issued. This is required.

**<runAsUser>**

User ID under which the action is to be performed. This is optional. The attribute is substitution, which accepts true or false.

**<approver>**

A user ID, or a list of user IDs separated by blanks. At least one user ID must approve the action before it is performed on behalf of the user ID that is specified with the runAsUser element. To specify multiple required approvers, use multiple approver elements (up to 12). The approver element is optional. If it is specified, then the runAsUser element is required. The action definition supports the same forms as the workflow definition. For more information and examples, see [“Specifying approvers for a step” on page 853](#).

**<unsolkey>**

Key to search for in the unsolicited messages, for a command-type action.

**<solkey>**

Key to search for in the solicited messages command response, for a command-type action.

**<detectTime>**

Time in seconds to search for the unsolkey in the unsolicited messages. Also, the minimum time before a command response is checked for after the command is submitted for execution.

**<workflow>**

Contains a workflow definition, consisting of the elements that follow.

**<cleanAfterComplete>**

Indicates whether the workflow is removed if it completes successfully. Accepts true, false, and inherit, which specifies that the value is inherited from the value of the workflow-clean-after-provisioned field for the instance. The default is inherit.

**<wfDefFile>**

Workflow definition file path. This element is required. The maximum length of the file path is 1024 characters.

**<wfVarInFile>**

Workflow variable input file path. This element is optional. The maximum length of the file path is 1024 characters.

**<wfVar>**

Assigns a value to a workflow instance variable defined in the action workflow. During processing of an action workflow, values for variables are obtained from the property file that is specified with element wfVarInFile. Use wfVar to assign a different value for an action workflow variable. You can:

- Specify an explicit value for the variable.
- Request that the variable value is obtained from the registry for the software services instance. Because all the variables and their values are captured in the software instance registry when the template is provisioned, using the wfVar element lets you share the selected variables between provisioning and other action workflows such as the deprovisioning action workflow.

This wfVar element is optional. Up to 1500 variables are allowed. It includes these attributes:

**name**

Is the name of the variable in the action workflow definition. The name must be unique.

**updateRegistry**

Indicates whether to update the variables in the software services instance registry from the action workflow. This is allowed only if the value for the action workflow variable is obtained from the registry. The value for updateRegistry must be true or false. The default is false. When the value is true, after the action is completed:

- If the variable already exists in the software services instance, the value for the variable is updated from the action workflow.
- If the variable does not already exist in the instance, the variable is created in the instance with the value from the workflow.

**Examples:** In the following example, the value for action workflow variable DFS\_PORTID is obtained from the registry variable that is identified by \${DFS\_PORTID}. The registry is updated with the new value for the DFS\_PORTID variable set during action processing.

```
<wfVar name="DFS_PORTID" updateRegistry="true">${DFS_PORTID}</wfVar>
```

In the following example, the value for action workflow variable DFS\_REGION\_TCPIPPORT is explicitly set to 8080. You cannot specify updateRegistry="true" when the variable value is explicitly specified.

```
<wfVar name="DFS_REGION_TCPIPPORT" updateRegistry="false">"8080"</wfVar>
```

### <instructions>

Defines an instruction.

### <description>

Contains a brief description of the action and the function it performs. This element is optional.

## Examples

The following illustrates the elements of an action definition.

```
<?xml version="1.0" encoding="utf-8"?>
<actionList>
  <action name="workflow1">
    <workflow>
      <wfDefFile>workflow1.xml</wfDefFile>
      <wfVar name="var1" updateRegistry="false">var1val</wfVar>
      <wfVar name="var2" updateRegistry="true">var2val</wfVar>
      <wfVar name="var3">var3val</wfVar>
    </workflow>
    <description>Description of Workflow 1</description>
  </action>
  <action name="workflow2">
    <workflow>
      <wfDefFile>workflow2.xml</wfDefFile>
    </workflow>
    <description>Description of Workflow 2</description>
  </action>
  <action name="instructions1">
    <instructions>The instructions</instructions>
    <description>For best results, read every word!</description>
  </action>
  <action name="command1">
    <command>
      <commandValue>d iplinfo</commandValue>
    </command>
    <description>What this command does</description>
  </action>
  <action name="deprovision" deprovision="true">
    <workflow>
      <wfDefFile>deprovision.xml</wfDefFile>
    </workflow>
    <description>This workflow can be deprovisioned</description>
  </action>
</actionList>
```

The following example shows definitions for deprovision, start, and stop actions.

```
<?xml version="1.0" encoding="utf-8"?>
<actionList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./actions_v1.xsd">
  <action name="deprovision">
    <workflow>
      <wfDefFile>./deprovision.xml</wfDefFile>
      <wfVarInFile>./workflow_variables.properties</wfVarInFile>
      <wfVar name="DFS_PORTID" updateRegistry="false">${DFS_PORTID}</wfVar>
    </workflow>
  </action>
</actionList>
```

```

        <wfVar name="DFS_SSLPORTID" updateRegistry="false">${DFS_SSLPORTID}</wfVar>
    </workflow>
    <description>This workflow can be deprovisioned</description>
</action>
<action name="start">
    <workflow>
        <wfDefFile>./Start.xml</wfDefFile>
        <wfVarInFile>./workflow_variables.properties</wfVarInFile>
        <wfVar name="url" updateRegistry="true">${ipadd}:${port}</wfVar>
    </workflow>
</action>
<action name="stop">
    <workflow>
        <wfDefFile>./Stop.xml</wfDefFile>
        <wfVarInFile>./workflow_variables.properties</wfVarInFile>
        <wfVar name="processID" updateRegistry="false">${pid}</wfVar>
    </workflow>
</action>
</actionList>

```

---

## Appendix A. Enabling tracing for the z/OS jobs REST interface

For diagnostic purposes, your installation might be asked by IBM Support to enable tracing for the z/OS jobs REST interface. This topic provides instructions for enabling several commonly-used traces.

Your installation can trace the use of the z/OS jobs REST interface on the z/OSMF system. Also, you can trace a variety of JES related activities, which can result from the use of the z/OS jobs REST interface services.

Understand that tracing carries a performance cost. Do not activate tracing for z/OSMF unless directed to do so by IBM Support.

### Tracing the z/OS jobs REST interface services

To trace the use of z/OS jobs REST interface services on the z/OSMF system, use the **MODIFY** command with the LOGGING option. Your user ID must be permitted to enter this operator command.

The command has the following format:

```
f server-name,logging='com.ibm.zosmf.restjobs.*=all'
```

where:

#### **server-name**

Is the server for your z/OSMF configuration. Set this value to the job name of the z/OSMF server, which is IZUSVR1, by default.

#### **com.ibm.zosmf.restjobs.\*=all**

Is the trace specification for the z/OS jobs REST interface.

Enter the command from the operator console. The command output is displayed in the operator console and in the z/OS system log.

Your changes take effect immediately and remain in effect while the server is running. Your changes are discarded when the server is restarted, and the previous settings are used.

To end this level of tracing and revert to the previous setting, enter the command again, and specify "reset" as the trace specification, for example:

```
f izusvr1,logging='reset'
```

### Tracing the JES related activities for your programs

For callers of the z/OS jobs REST interface, your installation can trace the JES related activities that can occur on behalf of program requests.

Specifically, you can trace the following JES related activities:

- Usage of the following subsystem interface (SSI) function codes:
  - Extended status function call (SSI function code 80), which allows a user-supplied program to obtain detailed status information about jobs and SYSOUT in the JES queue
  - Modify job function call (SSI function code 85), which allows a user-supplied program to modify job properties and to manage memory associated with the request.
- VSAM related activities.
- JES symbolic parameter substitutions.
- HSM recall activities.

To capture this type of information, you must add the appropriate trace specifications to the z/OSMF bootstrap.template file, which ensures that the server is started with the proper traces enabled. z/OSMF writes the trace output to files in the z/OSMF logs directory.

To start this type of tracing, do the following:

1. Locate the z/OSMF bootstrap.template file. By default, the location is: /etc/zosmf/servers/zosmfServers/bootstrap.template
2. Save a copy of the existing bootstrap.template file as a back-up.
3. Edit the bootstrap.template file, as needed:
  - Add the property `zosjes.logging=t` to capture information about the following activities:
    - Usage of the extended status function call (SSI function code 80)
    - VSAM related activities.
  - Add the property `izurestjobs.logging=t` to capture information about the following activities:
    - Usage of the modify job function call (SSI function code 85)
    - JES symbolic parameter substitutions
    - HSM recall activities.

A portion of the file is shown in Figure 469 on page 978.

---

```
# Licensed Materials - Property of IBM
#
# "Restricted Materials of IBM"
#
# Copyright IBM Corp. 2013 All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with
# IBM Corp.
#
# -----
#

izu.hostname=*
izu.https.port=443

# Trace options follow...
zosjes.logging=t
izurestjobs.logging=t
```

*Figure 469. Bootstrap properties for z/OSMF*

---

4. Save the bootstrap.template file.
5. Restart the z/OSMF server and resume z/OSMF operations.

Your changes will take effect immediately and are maintained across z/OSMF server restarts.

To work with the z/OSMF log files, you require a user ID with z/OSMF administrator authority (that is, a user ID defined to the z/OSMF administrator security group).

For information about how to enable other trace options for z/OSMF, and how to work with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

## Appendix B. Creating product information files for the Software Management task

A *product information file* is a flat file, such as a text file, that contains information about one or more products. This information includes, for example, the product announce date, general availability date, and end of service date. You can create your own product information files or obtain them from a provider, such as IBM, another vendor, or a third party.

z/OSMF displays data from product information files in several views in the Software Management task. For example, this information is displayed in the Products page, the Products, Features, and FMIDs page, and the End of Service report.

### Syntax for product information files

To be processed by z/OSMF, product information files must be formatted as JSON data and have the following syntax:

```
{
  "Version": "date-modified",
  "Products":
  [
    {
      "prodName": "product-name",
      "prodId": "product-identifier",
      "prodVRM": "version-release-modification",
      "GAAnnounceDate": "date-announced",
      "GADate": "general-availability-date",
      "URL": "URL",
      "EOSDate": "end-of-service-date",
      "country": "country"
    }
  ]
}
```

where,

#### **date-modified**

Date the file was created or last updated. The date must have the format YYYY-MM-DD. The date is required.

#### **product-name**

Name of the product. The name is optional, and is not used by z/OSMF. To omit the product name, exclude the field, type `null` as the value, or set the value equal to an empty string.

#### **product-identifier**

Identifier of the product. The product ID is required.

#### **version-release-modification**

Version, release, and modification level of the product. The value has the format *VV.RR.MM*, where *VV* is the two-digit version, *RR* is the two-digit release, and *MM* is the two-digit modification level. The version, release, and modification level are required.

#### **date-announced**

Date the vendor publicly announced the details of the product. The date must have the format YYYY-MM-DD. The date is optional. To omit the date, exclude the field or type `null` as the value.

#### **general-availability-date**

Date that a version or release of the product is available to all users. The date must have the format YYYY-MM-DD. The date is optional. To omit the date, exclude the field or type `null` as the value.

#### **URL**

URL that links to additional information about the product. This information can include, for example, product life cycle dates, product highlights, planning information, and technical descriptions. The URL

is optional. To omit the URL, exclude the field, type `null` as the value, or set the value equal to an empty string.

**end-of-service-date**

Last date on which the vendor will deliver standard support services for a given version or release of the product. This date is the general end of service date. It does not account for lifecycle extensions. The date must have the format YYYY-MM-DD. The date is optional. To omit the date, exclude the field or type `null` as the value.

**country**

Country for which the end of service date is applicable. The country is optional. To omit the country, exclude the field, type `null` as the value, or set the value equal to an empty string.

The information for each product must be contained within separate braces (`{ }`) inside the brackets (`[ ]`), and each set of braces must be comma separated. For a sample file that contains the information for two products, see [Figure 470 on page 981](#).

## Sample product information file

```
{
  "Version": "2011-06-30",
  "Products":
  [
    {
      "prodName": "z/OS",
      "prodId": "5694-A01",
      "prodVRM": "01.10.00",
      "GAAnnounceDate": "2008-08-05",
      "GADate": "2008-09-26",
      "URL": "http://www-03.ibm.com/systems/z/os/zos/",
      "EOSDate": "2011-09-30",
      "country": "US"
    },
    {
      "prodName": "z/OS",
      "prodId": "5694-A01",
      "prodVRM": "01.13.00",
      "GAAnnounceDate": "2011-07-12",
      "GADate": null,
      "URL": "",
      "country": "US"
    }
  ]
}
```

Figure 470. Sample product information file for the Software Management task

## Working with the IBM product information file

The product information file that IBM supplies for System z® software is located at the following URL: [Product information file for IBM Z software products \(public.dhe.ibm.com/services/zosmf/JSONs/IBMProductEOS.txt\)](http://public.dhe.ibm.com/services/zosmf/JSONs/IBMProductEOS.txt).

To load the contents of the file into z/OSMF, do one of the following:

- Load directly from the URL.
- Manually download the file at the URL to your local workstation.
- Manually download the file at the URL to a z/OS data set or UNIX file that the primary z/OSMF host system can access.

When transferring the file from a workstation to a z/OS data set or UNIX file, transfer the file in binary format. To avoid errors, do not convert the file to the EBCDIC character set.

After you store the file in your desired location, to retrieve its contents, complete the steps provided in the *Retrieving product information from product information files* topic in the z/OSMF online help.



## Appendix C. Understanding the Portable Software Instance descriptor file

A software instance is a collection of data sets containing installed software, and other data sets that may be associated with that installed software. The software may be SMP/E managed, in which case the collection of data sets also contains, and is described by, one or more SMP/E target and distribution zone pairs, defined by a single global zone.

A portable software instance is, exactly as the name implies, a portable form of a software instance, which can be used to simplify distribution of a software instance across a network, and can be deployed by the z/OSMF Software Management task. A portable software instance is a set of portable archive files created by the SMP/E GIMZIP service routine for each of the data sets defined to the software instance, including SMPCSI data sets with all associated SMP/E managed target and distribution libraries, and a descriptor file to describe in detail the entire originating software instance.

### Portable Software Instance Descriptor File

The portable software instance descriptor file contains detailed information to describe the content of the originating software instance. It contains the information required by the z/OSMF Software Management task to perform a deployment operation on the content of the portable software instance. The portable software instance descriptor file has a file name of IZUD00DF.json, and is created by z/OSMF during the operation of an Export action on a software instance.

The content of the portable software instance descriptor file is shown in the following example:

```
{ "izud.pswi.descriptor":
{
  "version": "pswi-descriptor-version"
  "created": "yyyy-mm-ddThh:mm:ssZ",
  "gimpaflocation": "relative-path-for-GIMPAF.XML-file",
  "name": "software-instance-name",
  "description": "software-instance-description",
  "globalzone": "csi-data-set-name",
  "zones": [ {
    "name": "zone-name",
    "type": "zone-type",
    "related": "related-zone-name",
    "csi": "csi-data-set-name"
  } ]
  "datasets": [ {
    "dsname": "data-set-name",
    "volumes": [ "volume-serial" ],
    "storclas": "storage-class",
    "dstype": "data-set-type",
    "tracks": "allocated-tracks",
    "zoneddefs": [ {
      "zone": "zone-name",
      "dddefs": [ { "dddef": "dddef-name", "path": "unix-directory", "pathcontainedinds": "unix-
directory-subset" } ]
    } ],
    "mountpoint": "UNIX-path",
    "unixdirs": [ "unix-directory" ],
    "isextendedformat": true | false,
    "recfm": "record-format",
    "lrecl": "logical-record-length",
    "blksize": "block-size",
    "used": "used-tracks-percent",
    "extents": "allocated-extents",
    "dscategory": [ "data-set-category" ],
    "archid": "gimzip-archive-id",
  } ]
  "smpeproducts": [ {
    "prodname": "product-name",
    "prodid": "product-id",
    "release": "vv.rr.mm",
    "vendor": "vendor-name",
    "url": "product-url",
  } ]
}
```

```

    "srels":["srel"],
    "prodsups":[{"prodid":"product-id",
    "release":"vv.rr.mm"
    }],
    "features":[{"featname":"feature-name",
    "featid":"feature-id",
    "fmids":["fmid-name"]
    }],
    "fmids":["fmid-name"],
    "products":[{"prodname":"product-name",
    "prodid":"product-id",
    "release":"product-level",
    "vendor":"vendor-name",
    "url":"product-url",
    "features":["feature-name"],
    "gadate":"yyyy-mm-ddThh:mm:ssZ",
    "eosdate":"yyyy-mm-ddThh:mm:ssZ" | "NotAnnounced",
    "prodinfoversion":"yyyy-mm-ddThh:mm:ssZ"
    }],
    "workflows":[{"name":"workflow-name",
    "description":"workflow-description",
    "location": {
    "smptype":"smp-type",
    "smpname":"smp-name",
    "dsname":"workflow-dsname",
    "path":"workflow-path"},
    "performonhostsystm":true | false
    }],
    "datasetproperties":[{"dddefname":"dddef-name",
    "zone":"zone-name",
    "dsname":"data-set-name",
    "volume":"volume-serial",
    "dstype":"DLIB",
    "properties":[{"key":"value"}]
    }],
    "datasetpropertylabels":[{"propertyname":"property-name",
    "label":"property-label"
    }],
    "productproperties":[{"prodid":"product-id",
    "release":"product-level",
    "prodname":"product-name",
    "properties":[{"key":"value"}]
    }],
  }
}

```

Where:

#### version

Indicates the version of the portable software instance descriptor. The version value may be:

1. The initial version of the descriptor file.
2. Adds support for non-SMP/E managed product information.
3. Adds support for SMP/E Product end-of-service information.
4. Adds support for workflows.
5. Adds support for datasetproperties and productproperties.
6. Adds support for pathcontainedinds and performonhostsystm
7. Adds support for datasetpropertylabels

#### created

Indicates the date and time when the portable software instance was created, in ISO 8601 format. For example, yyyy-mm-ddThh:mm:ssZ.

**gimpaflocation**

Indicates the relative path to the GIMPAF.XML file for this portable software instance, relative to the location of the portable software instance descriptor file.

**name**

Indicates the name for the originating software instance.

**description**

Indicates the description of the originating software instance. This is an optional property.

**globalzone**

Indicates the name of the CSI data set that contains the global zone. This is an optional property, specified only if the originating software instance describes SMP/E managed software.

**zones**

Indicates the list of SMP/E zones from the originating software instance. This is an optional property, specified only if the originating software instance describes SMP/E managed software.

**name**

Indicates the zone name.

**type**

Indicates the type for the zone, global, target or dlib.

**related**

Indicates the name of the zone's related zone, if any.

**csi**

Indicates the name for the CSI data set that contains the zone.

**datasets**

Indicates the list of data sets from the originating software instance.

**dsname**

Indicates the originating data set name.

**volumes**

Indicates the list of volume serials where the originating data set resided.

**storclas**

Indicates the name of the storage class where the originating data set resided. This is an optional property.

**dstype**

Indicates the type for the data set. Can be one of the following types:

- HFS — Hierarchical file system.
- PDS — Partitioned data set.
- PDSE — Partitioned data set extended.
- SEQ — Sequential data set.
- VSAM — VSAM data set.
- ZFS — zSeries file system.

**tracks**

Indicates the number of 3390-device equivalent tracks (56664 bytes/track) allocated to the data set.

**zonedddefs**

Indicates the list of SMP/E zones and DDDEF entries that reference the data set. This is an optional property, specified only if the originating software instance describes SMP/E managed software, and if the subject dataset is referenced in the SMP/E zones.

**zone**

Indicates the name of an SMP/E zone that contains one or more DDDEF entries for the data set.

**dddefs**

Indicates the list of DDDEF entries that identify the data set.

**dddef**

Indicates the name of the DDDEF entry.

**path**

Indicates the UNIX directory identified in the DDDEF entry. Null if the DDDEF entry identifies a data set.

**pathcontainedinds**

The fixed portion of the UNIX path that is contained in the subject data set. This value excludes the mount point and any intermediate symbolic links that are specified in a DDDEF entry.

**mountpoint**

Indicates the mount point for the originating UNIX file system data set. Null if the DDDEF entry identifies a data set instead of a UNIX directory. This is an optional property, specified only if the originating software instance describes SMP/E managed software, and if the subject dataset is referenced in the SMP/E zones by a DDDEF entry with a UNIX directory.

**unixdirs**

A list of UNIX directories that reside in the subject data set and contain one or more workflow definition files for the software instance.

**isextendedformat**

Indicates, true or false, if the data set is an extended format sequential data set.

**recfm**

Indicates the record format. The record format can be any valid combination of the following codes:

- A — ASA printer control characters.
- B — Blocked records.
- F — Fixed-length records.
- M — Machine code printer control characters.
- S — Standard (for F) or spanned (for V); used only with sequential data sets.
- T — Track-overflow feature.
- U — Undefined format records.
- V — Variable-length records.

**lrecl**

Indicates the logical record length.

**blksize**

Indicates the block size, in bytes.

**used**

Indicates the percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

**extents**

Indicates the number of extents allocated to the data set.

**dscategory**

List of categories for how the data set is used. Can be one or more of the following:

- DLIB — SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- GLOBAL — SMP/E control data set associated with the global zone.
- SMP — SMP/E control data set.
- SMPTLIB — SMPTLIB data sets associated with the global zone.
- TARGET — SMP/E managed target library, or SMP/E control data set associated with a target zone.
- WORKFLOW — Contains one or more workflow definition files for the workflows explicitly defined to the software instance.

- OTHER — None of the above.

**archid**

Indicates the archive ID value, produced by the GIMZIP service routine and specified in the GIMPAF.XML file, to identify the portable archive file for the data set.

**smpeproducts**

Indicates the list of software products installed in the originating software instance that are managed by SMP/E. This is an optional property.

**prodname**

Indicates the name for the product.

**prodid**

Indicates the identifier for the product.

**release**

Indicates the version, release, and modification level for the product, in this format: vv.rr.mm.

**vendor**

Indicates the name for the product's vendor. This is an optional property.

**url**

Indicates the URL that links to additional information about the product. This is an optional property.

**srels**

Indicates the system or subsystem releases on which the subject product can be installed.

**prodsups**

Indicates the list of products that are superseded by the subject product. This is an optional property.

**features**

Indicates the list of features for the subject product.

**featname**

Indicates the name for the feature.

**featid**

Indicates the identifier for the feature.

**fmids**

Indicates the list of FMIDs for the subject feature. It is also the list of FMIDs in the originating software instance. This list includes all FMIDs associated with one or more products and features, and all FMIDs associated with no products or features.

**products**

The list of products from the originating software instance that are not managed by SMP/E

**product-name**

Name of the product, but can be up to 64 characters.

**product-ID**

Identifier for the product, but can be up to 64 characters.

**product-level**

Release level for the product, but can be up to 64 characters.

**vendor-name**

Name of the vendor that provides the product, but can be up to 64 characters.

**product-URL**

A URL that links to additional information about the product, but can be up to 256 characters.

**feature-name**

List of names of features for the product, but can be up to 64 characters

**gadate**

The date when the product became generally available. May be null, or a date value, in ISO 8601 format.

**eosdate**

The last date on which the vendor will deliver standard support services for the product.

- null - The end of service date is unknown for the product.
- yyyy-mm-ddThh:mm:ssZ - The known end of service date, in ISO 8601 format.
- NotAnnounced - The end of service date has not yet been announced for the product.

#### **prodinfoversion**

The version for the most recent product information file that provided information for the subject product. The version value represents the date the file was created or updated. May be null, or a date value, in ISO 8601 format.

#### **workflows**

List of workflows for the software instance.

#### **workflow-name**

Name for the workflow.

#### **workflow-description**

Description for the workflow.

#### **location**

Location of the workflow definition file for the workflow.

#### **smp-type**

The SMP/E element type for a workflow definition file that is managed by SMP/E.

#### **smp-name**

The SMP/E element name for a workflow definition file that is managed by SMP/E.

#### **workflow-dsname**

The name of the data set that contains the workflow definition file.

#### **workflow-path**

The UNIX path for a workflow definition file that is a UNIX file.

#### **performonhostsystem**

Indicates whether the workflow steps may be performed on the host system or on another system in the same sysplex as the host system. This property is optional and the default value is true.

#### **true**

Indicates that the workflow steps may be performed on the z/OSMF host system on which the software instance resides.

#### **false**

Indicates that the workflow steps may be performed on a system in the sysplex other than the z/OSMF host system on which the software instance resides.

#### **datasetproperties**

A list of one or more properties for individual data sets. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,” on page 991](#) for more information.

#### **dddefname**

The name of the SMP/E DDDEF entry that describes an SMP/E managed data set.

#### **zone**

The zone name where the DDDEF entry resides. The zone is specified to identify a unique data set when there are more than one DDDEF entries with the same name in different zones, and each DDDEF entry identifies a different data set in the software instance.

#### **dsname**

The name of the subject data set. The data set name is specified to identify a non-SMP/E managed data set.

#### **volume**

The volume of the subject data set. The volume is specified to identify a non-SMP/E managed data set where the volume was specified to identify an uncatalogued data set.

**dstype**

The usage type of the subject data set. A value of DLIB indicates the data set is an SMP/E managed distribution library, or an SMP/E control data set associated with a distribution zone.

**properties**

A list of one or more properties for the subject data set, specified as key-value pairs.

**datasetpropertylabels**

A list of labels that each correspond to unique data set properties that a provider defines in datasetproperties. Label values are used for column headings to display provider defined data set property values on the Deployment Configuration Data Sets page. Not all provider defined data set properties must have corresponding defined labels, but only those with defined labels are eligible for display on the Deployment Configuration Data Sets page. A data set property can have only one associated label, and all labels must be unique.

**propertyname**

The name, or key, of the existing provider defined property.

**label**

The unique label that is displayed on the Deployment Configuration Data Sets page. Label values can contain up to 20 characters.

**productproperties**

A list of one or more properties for individual software products. These properties are made available to a workflow as workflow variable properties when Software Management creates a workflow instance for the software instance. See [Appendix D, “Software Management workflow variables,”](#) on page 991 for more information.

A prodid and release must be specified to uniquely identify an SMP/E product.

Prodname, prodid, and release may be specified to uniquely identify a non-SMP/E product.

**prodid**

The identifier for the subject product. This is required to identify an SMP/E managed product.

**release**

The version, release, modification level for the subject product. This is required to identify an SMP/E managed product.

**prodname**

The name of the subject product. This is required to identify a non-SMP/E managed product.

**properties**

A list of one or more properties for the subject product, specified as key-value pairs.



---

## Appendix D. Software Management workflow variables

You can use workflows in Software Management to perform setup and configuration tasks for a software instance. When a software instance is deployed, you can use workflows that are defined for the software instance to perform those setup and configuration tasks.

Software Management provides detailed properties about a software instance as well as the data sets and products that compose software instances. These properties are exposed to workflows as workflow variables.

**Note:** The support for the workflow variables is added with the PTFs for APAR PH09032.

The workflow variables that are created by Software Management for a software instance are shown as follows:

```
izud-varsversion = version
izud-created = yyyy-mm-ddThh:mm:ssZ
izud-createdby = user-id
izud-system = system-nickname
izud-fmids= ["fmid-name"]
izud-globalzone= csi-data-set-name
izud-zones= [{
  "izud-name":"zone-name",
  "izud-type":"zone-type",
  "izud-related":"related-zone-name",
  "izud-csi":"csi-data-set-name"
}]
izud-datasets = [{
  "izud-aliases":["alias"],
  "izud-cataloged":true | false,
  "izud-catalog":"catalog.data.set.name",
  "izud-blksize":"block-size",
  "izud-dddefs": [{
    "izud-dddef":"dddef-name",
    "izud-zone":"zone-name",
    "izud-path":"unix-directory"
  }],
  "izud-dscategories":["data-set-category"],
  "izud-dsname":"dataSetName",
  "izud-dstype":"dataSetType",
  "izud-extents":"allocated-extents",
  "izud-extendedformat":true | false,
  "izud-lrecl":"logical-record-length",
  "izud-mountpoint":"UNIX-path",
  "izud-recfm":"record-format",
  "izud-storclas":"storage-class",
  "izud-tempcatalias":"temporary-catalog-alias",
  "izud-tracks":"allocated-tracks",
  "izud-unixdirs":["unix-directory"],
  "izud-used":"used-tracks-percent",
  "izud-volumes":["volser"],
  "provider-dataset-property-key":"provider-dataset-property-value"
}]
izud-catalogs = [{
  "izud-catname":"catalog-name",
  "izud-cattype":"MASTER | USER",
  "izud-catdsnprefixes":["data-set-name-prefix"],
  "izud-tempcatalias":"temporary-catalog-alias",
  "izud-tgtsyscattype":"MASTER | USER"
}]
izud-products= [{
  "izud-eosdate":"yyyy-mm-ddThh:mm:ssZ" | "NotAnnounced" | "Unknown",
  "izud-features": [{
    "izud-featname":"feature-name",
    "izud-featid":"feature-id",
    "izud-fmids":["fmid-name"]
  }],
  "izud-gadate":"yyyy-mm-ddThh:mm:ssZ",
  "izud-prodname":"product-name",
  "izud-prodid":"product-id",
  "izud-prodinfoversion":"yyyy-mm-ddThh:mm:ssZ",
```

```

"izud-prodsups": [{
  "izud-prodid": "product-id",
  "izud-release": "vv.rr.mm"
}],
"izud-release": "product-level",
"izud-srels": ["srel"],
"izud-url": "product-url",
"izud-vendor": "vendor-name",
"provider-product-property-key": "provider-product-property-value"
}]

```

Where:

#### **izud-varsversion**

Indicates the version of the workflow variables that are defined by z/OSMF Software Management. izud-varsversion is an integer type variable. The version value can be:

1. The initial version of the variable set.

#### **izud-created**

Indicates the date and time when the workflow instance was created. izud-created is a string type variable.

#### **izud-createdby**

The user ID for the user that created the workflow instance. izud-createdby is a string type variable.

#### **izud-system**

The nickname of the z/OSMF host system where the software instance data sets reside. For Deployment Perform workflows, this is the target system of the deployment operation. izud-system is a string type variable.

#### **izud-fmids**

The list of FMIDs in the software instance. This includes all FMIDs associated with one or more products and features, as well as all FMIDs associated with no products or features. izud-fmids is an array type variable.

#### **izud-globalzone**

Indicates the name of the CSI data set that contains the global zone for the software instance. izud-globalzone is a string type variable.

#### **izud-zones**

Indicates the list of SMP/E zones for the software instance. izud-zones is an array type variable.

##### **izud-name**

Indicates the zone name.

##### **izud-type**

Indicates the type for the zone, global, target, or dlib.

##### **izud-related**

If the zone has a related zone, then this indicates the name of the zone's related zone.

##### **izud-csi**

Indicates the CSI data set name that contains the zone.

#### **izud-datasets**

The list of data sets in the software instance. izud-datasets is an array type variable.

##### **izud-aliases**

A list of alias names for the data set.

##### **izud-cataloged**

Indicates, true or false, whether the data set is cataloged.

##### **izud-catalog**

For cataloged data sets, the name of the catalog that includes the data set.

##### **izud-blksize**

The data set block size.

##### **izud-dsname**

The data set name.

**izud-dstype**

The type for the data set. Can be one of the following types:

- HFS — Hierarchical file system.
- PDS — Partitioned data set.
- PDSE — Partitioned data set extended.
- SEQ — Sequential data set.
- VSAM — VSAM data set.
- ZFS — zSeries file system.

**izud-dscategory**

List of categories for how the data set is used. Can be one or more of the following:

- DLIB — SMP/E managed distribution library, or SMP/E control data set associated with a distribution zone.
- GLOBAL — SMP/E control data set associated with the global zone.
- SMP — SMP/E control data set.
- SMPTLIB — SMPTLIB data sets associated with the global zone.
- TARGET — SMP/E managed target library, or SMP/E control data set associated with a target zone.
- WORKFLOW — Contains one or more workflow definition files for the workflows that are explicitly defined to the software instance.
- OTHER — None of the above.

**izud-volumes**

The list of volume serials where the data set resides.

**izud-storclas**

The name of the storage class where the data set resides.

**izud-tempcatalias**

The name of the temporary catalog alias used to uniquely identify the data set from the z/OS driving system catalog. This value is the temporary catalog alias that is used for the alternate master catalog and is a data set name prefix to reference the data set from the driving system. To reference the data set from the driving system catalog, use a name that is constructed like this: izud-tempcatalias.izud-dsname.

If a data set is cataloged in the new alternate master catalog indirectly (using a volume symbol instead of a specific volume serial) or not cataloged at all, then the izud-tempcatalias value is null. Such data sets cannot be referenced by name from the driving system catalog. They must be referenced using the volume instead.

If not creating an alternate master catalog izud-tempcatalias is null.

**izud-tracks**

The number of 3390-device equivalent tracks (56664 bytes/track) allocated to the data set.

**izud-dddefs**

Indicates the list of SMP/E zones and DDDEF entries that reference the data set.

**izud-dddef**

Indicates the name of the DDDEF entry.

**izud-zone**

Indicates the name of an SMP/E zone that contains the DDDEF entry.

**izud-path**

Indicates the UNIX directory that is identified in the DDDEF entry.

**izud-mountpoint**

Indicates the mount point for the UNIX file system data set.

**izud-unixdirs**

A list of UNIX directories that reside in the subject data set and contain one or more workflow definition files for the software instance.

**izud-extendedformat**

Indicates, true or false, whether the data set is an extended format sequential data set.

**izud-recfm**

Data set record format.

**izud-lrecl**

Data set logical record length.

**izud-used**

Indicates the percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1. If the data set is a PDSE, the percentage refers to the percentage of allocated pages used.

**izud-extents**

The number of extents that are allocated to the data set.

**provider-dataset-key and provider-dataset-property-value**

A property key and its value that is defined by the provider of the software instance for the subject data set. Each property that is defined for the data set is represented as a unique key-value pair. See [“Add a new software instance” on page 428](#) for more information about the datasetproperties.

**izud-catalogs**

The list of catalogs where data sets in the software instance are cataloged. izud-catalogs is an array type variable.

**izud-catname**

Name of the catalog.

**izud-cattype**

Indicates the catalog type, MASTER or USER, from the z/OS driving system perspective. **Note:** When creating a new alternate master catalog, from the z/OS driving system the new alternate master catalog is a user catalog. It is a master catalog only to the z/OS target system after that system is IPLed.

**izud-catdsnprefixes**

A list of data set name prefixes for data sets cataloged in the subject catalog.

**izud-tempcatalias**

When creating an alternate master catalog, izud-tempcatalias indicates the temporary catalog alias that is used to uniquely identify the data sets in the subject catalog from the z/OS driving system catalog.

If not creating an alternate master catalog, or the saved configuration information does not match the actuals, then z/OSMF might not be able to determine the temporary catalog alias. Therefore, izud-tempcatalias will be null.

**izud-tgtsyscattype**

When creating an alternate master catalog, izud-tgtsyscattype indicates the intended catalog type for the subject catalog, MASTER or USER, from the z/OS target system perspective.

If not creating an alternate master catalog, or the saved configuration information does not match the actuals, then z/OSMF might not be able to determine the intended catalog type. Therefore, izud-tgtsyscattype will be null.

**izud-products**

The list of software products in the software instance. This list includes both SMP/E and non-SMP/E managed products. izud-products is an array type variable.

**izud-prodname**

The name for the product.

**izud-prodid**

The identifier for the product.

**izud-release**

The release level for the product.

**izud-vendor**

The name of the vendor that supplied the product.

**izud-url**

The URL that links to additional information about the product.

**izud-srels**

The SMP/E system or subsystem releases on which the product can be installed.

**izud-prodsups**

The list of products that are superseded by the subject product.

**izud-prod-id**

The identifier for the superseded product.

**izud-release**

The release level for the superseded product.

**izud-features**

The list of features for the product.

**izud-featname**

The name of the feature.

**izud-featid**

The identifier for the feature.

**izud-fmids**

The list of FMIDs for the feature.

**izud-gadate**

The date when the product became generally available.

**izud-eosdate**

The last date on which the vendor delivers standard support services for the product.

**izud-prodinfoversion**

The version for the most recent product information file that provided information for the subject product.

**provider-product-key and provider-product-property-value**

A property key and its value defined by the provider of the software instance for the subject software product. Each property defined for the product is represented as a unique key-value pair. See [“Add a new software instance” on page 428](#) for more information about the productproperties.

## Provider Defined Properties

As a software provider you can define properties that are associated with individual data sets and products within a software instance. These properties are available to a workflow created by Software Management as workflow variable properties. These properties are merged with those defined by Software Management to describe the data sets and products within a software instance to create the workflow variables previously described.

Provider defined properties can be specified when you use the Software Management REST services to add or modify a software instance. Refer to the [“Add a new software instance” on page 428](#) and [Modify software instance](#) REST services to learn how to specify provider properties. Provider defined properties are merged with the Software Management defined properties to create workflow variables. These variables are made available to a workflow when Software Management creates a new workflow instance for the software instance.

**Note:** The support for provider defined properties is added with the PTFs for APAR PH11650.

## Example

If a data set in a software instance must be APF authorized for the software to run properly, as the provider of the software, you can create a workflow to help your users ensure that the data set is APF authorized when the software instance is deployed. Your workflow needs to know which data sets in the software instance must be APF authorized, therefore, you can define a property for the subject data sets and the workflow can reference this property in the workflow variables.

When you define the software instance by using the REST service to add a new software instance, as described in “Add a new software instance” on page 428, specify a data set property by using *datasetproperties* in the request content like this:

```
POST /zosmf/swmgmt/swi HTTP/1.1
Host: pev084.yourco.com
Content-Type: application/json
Accept-Language: en

{"swiname": "mySWI",
 "system": "PEV084",
 "globalzone": "JOHNDOE.GLOBAL.CSI",
 "targetzones": ["TGTZ"],
 "datasetproperties": [
   {"dddefname": "ABCMOD",
    "properties": [{"abc-ApfLst": "yes"}]}
 ]
}
```

*Figure 471. Using the Add software instance REST API to specify a data set property in the request content*

In this example the data set that must be APF authorized is managed by SMP/E and is identified by a DDDEF entry named ABCMOD. A property with a key "abc-ApfLst" and value "yes" are defined for the subject data set.

When Software Management creates a workflow instance for the software instance, workflow variables are generated, which provides the workflow programmatic access to the information about the data sets in the software instance. The workflow variable *izud-datasets* is a generated array variable containing one entry for each data set in the software instance. The entry in the *izud-datasets* array for the subject data set will contain properties like this:

```
izud-datasets = [
  ...
  {"izud-dsname": "ABC.ABCMOD",
   "izud-volumes": ["LV1234"],
   "izud-dstype": "PDS",
   ...
   "abc-ApfLst": "yes"
 },
  ...
]
```

*Figure 472. Sample entry in the list for the subject data set*

The data set property that you specified when you defined the software instance, abc-ApfLst, is set for the subject data set in the *izud-datasets* array entry. A template step in your workflow can analyze the entries in the *izud-datasets* array variable to determine which data sets in the software instance have this property, like this:

```

<template>
<inlineTemplate substitution="true">
APF FORMAT(DYNAMIC)
### For each data set in the software instance,
### find the data sets that must be APF authorized.
###
#foreach($dataset in ${instance-izud-datasets})
#if(${dataset.abc-ApfLst} == "yes")
APF ADD DSNAME(${dataset.izud-dsname}) VOLUME(${dataset.izud-volumes[0]})
#end
#end
</inlineTemplate>
<saveAsDataset substitution="true">
${instance-izud-createdby}.PARMLIB(PROGXX)</saveAsDataset>
</template>

```

*Figure 473. Sample template step to determine the specified property*

This example workflow template step creates a PROGxx parmlib member snippet for the data sets with the abc-ApfLst property set to yes. Workflow template steps such as this can be created for many other actions.

## Sample workflow definition

A sample workflow definition file named IZUDWFVR is provided in the SYS1.SAMPLIB data set. It demonstrates how to create a workflow that references the workflow variables that are defined by Software Management.



---

## Appendix E. Accessibility

Accessible publications for this product are offered through [IBM Documentation \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the [Contact the z/OS team web page \(www.ibm.com/systems/campaignmail/z/zos/contact\\_z\)](http://www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation  
Attention: MHVRCFS Reader Comments  
Department H6MA, Building 707  
2455 South Road  
Poughkeepsie, NY 12601-5400  
United States

---

### Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

---

### Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

---

### Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Vol I*

---

### Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Documentation with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1)

are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The \* symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element \*FILE with dotted decimal number 3 is given the format 3 \\* FILE. Format 3\* FILE indicates that syntax element FILE repeats. Format 3\* \\* FILE indicates that syntax element \* FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1\*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

#### **? indicates an optional syntax element**

The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

#### **! indicates a default syntax element**

The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

#### **\* indicates an optional syntax element that is repeatable**

The asterisk or glyph (\*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the \* symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1\* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3\* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

#### **Notes:**

1. If a dotted decimal number has an asterisk (\*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The \* symbol is equivalent to a loopback line in a railroad syntax diagram.

**+ indicates a syntax element that must be included**

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the \* symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the \* symbol, is equivalent to a loopback line in a railroad syntax diagram.



## Notices

---

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation  
Site Counsel  
2455 South Road*

Poughkeepsie, NY 12601-5400  
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

## Terms and conditions for product documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

### **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### **Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM Online Privacy Statement

---

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at [ibm.com/privacy](http://ibm.com/privacy) and IBM's Online Privacy Statement at [ibm.com/privacy/details](http://ibm.com/privacy/details) in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at [ibm.com/software/info/product-privacy](http://ibm.com/software/info/product-privacy).

## Policy for unsupported hardware

---

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

## Minimum supported hardware

---

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

## Trademarks

---

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle, its affiliates, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names might be trademarks or service marks of others.

---

# Index

## A

- About this document [xlix](#)
- accessibility
  - contact IBM [999](#)
  - features [999](#)
- actions
  - get responses for REST API [346](#)
  - instances REST API [337](#)
- adding plug-ins [963](#)
- Application Linking Manager interface
  - description [7](#)
  - event handlers [9](#)
  - event requestors [9](#)
  - event types [9](#)
  - obtaining a list of handlers for an event type [29](#)
  - obtaining a list of the tasks that are eligible to be handlers [28](#)
  - registering an event handler [25](#)
  - registering an event type [23](#)
  - unregistering an event handler [30](#)
  - unregistering an event type [31](#)
- application programming interface for z/OSMF [1](#)
- application server routing services
  - deleting data from an application server [42](#)
  - editing data for an application server [39](#)
  - modifying data for an application server [39](#)
  - removing data from an application server [42](#)
  - retrieving data from an application server [35](#)
  - updating data for an application server [39](#)
- Application server routing services
  - overview [32](#)
- approver element [852](#)
- approving
  - templates REST API [242](#), [243](#)
- archiving
  - templates REST API [233](#)
- assistive technologies [999](#)
- authenticate the user with a JSON web token [719](#)
- autoEnable attribute [846](#)
- automated step [846](#)
- automation-info object [743](#), [791](#)

## C

- caller scope variable [870](#)
- change a user password or passphrase [721](#)
- command
  - get detection results in unsolicited messages [551](#)
  - get response [545](#)
  - issue [528](#)
- condition attribute [840](#), [850](#)

- conditional step
  - description [804](#), [821](#)
  - using [840](#), [850](#)
- console
  - get a command response [545](#)
  - get detection results [551](#)
  - issue a command [528](#)
- contact
  - z/OS [999](#)
- Create z/OS UNIX zFS Filesystem
  - Create z/OS UNIX zFS Filesystem [623](#)
- creating
  - templates
    - new versions [178](#)
    - REST API [170](#)
- creating z/OSMF plug-ins
  - Application Linking Manager Javascript APIs
    - getEventFromUrl function [941](#)
    - getHandlers function [939](#)
    - hasLaunchContext function [940](#)
    - onLoadingComplete function [945](#)
    - sendEvent function [938](#)
    - subscribe function [943](#)
  - communicating with z/OSMF core
    - cleanupBeforeDestroy function [930](#)
    - cleanupBeforeDestroyComplete API [931](#)
    - getUserSessionId function [932](#)
    - isContentChanged function [926](#)
    - programmaticallyCloseTab function [929](#)
    - shouldClose function [928](#)
  - developing applications [924](#)
  - developing documentation [952](#)
  - security [969](#)

## D

- data persistence services
  - deleting persisted user or application data [372](#)
  - deleting z/OSMF settings data [707](#)
  - persisting user or application data [367](#)
  - retrieving persisted user or application data [369](#)
- Data persistence services
  - overview [366](#)
- data set
  - creating [585](#)
  - deleting [588](#), [590](#)
  - list [631](#), [632](#), [634](#), [637](#)
  - listing [569](#)
  - listing members [572](#)
  - reading members [575](#)
  - writing to member [580](#)
- data set member
  - list [572](#)
  - read [575](#)
  - write [580](#)
- defining plug-ins [963](#)
- Delete method

## Delete method (*continued*)

Delete z/OS UNIX zFS Filesystem [625](#)

## DELETE method

delete a PDS or PDSE [590](#)

delete a tenant [109](#)

delete a UNIX file or directory [613](#)

delete a workflow on a z/OS system [768](#)

delete an archived workflow on a z/OS system [801](#)

delete authentication tokens [724](#)

delete data from a remote z/OSMF instance [392](#)

delete data from a secondary z/OSMF instance [392](#)

delete data from an application server [42](#)

delete data from one system [392](#)

delete persisted user or application data [372](#)

delete portable software instances [464](#)

delete response for action for a software services instance [354](#)

delete software instances [452](#)

delete software services instances [333](#)

delete system variables [717](#)

delete z/OS data sets [588](#)

delete z/OSMF settings data [707](#)

deleting data from all the system in a central processor complex (CPC) [392](#)

deleting data from all the system in a group [392](#)

deleting data from all the system in a sysplex [392](#)

deleting data from multiple systems [392](#)

end a TSO/E address space [511](#)

place a TSO/E address space into dormant state [511](#)

purging a job from the JES spool [680](#)

remove data from an application server [42](#)

remove software instances [452](#), [464](#)

unregister an event handler [30](#)

unregister an event type [31](#)

delete the authentication tokens for this user [724](#)

## Delete z/OS UNIX zFS Filesystem

Delete z/OS UNIX zFS Filesystem [625](#)

## deleting

templates

REST API [194](#)

## DELTE method

get software services templates [246](#)

## E

### entity tag (ETag)

using [566](#)

### error report document

for z/OS data set and file REST interface [640](#)

### event handler

obtaining a list [28](#), [29](#)

registering [25](#)

unregistering [30](#)

### event requestor [14](#)

### event type

registering [23](#)

unregistering [31](#)

### Export

software instance [434](#)

### Export a defined software instance

Export a defined software instance [434](#)

## F

### feedback

collecting from user of a step [818](#)

### feedback attribute [818](#)

### file directory

Create [610](#)

### file system

mounting [626](#), [629](#)

### Filesystem

operations [623](#), [625](#)

### Filesystems

operations [621](#)

## G

### get approval

templates

REST API [236](#)

### Get method

Get the response to a command that was issued with the TSO/E REST API [501](#)

### GET method

get a command response [545](#)

get a domain [79](#), [84](#)

get a domain resource pool [138](#)

Get a notification [405](#)

get a published software services template [257](#)

get a published software services template history [269](#)

get a resource pool [132](#)

get a resource pool history [145](#)

get a software services instance [301](#)

get a tenant for cloud provisioning [94](#)

get a tenant history for cloud provisioning [100](#)

get data set attributes [74](#)

get detection results in unsolicited messages [551](#)

get prompt variables for a software services template [273](#)

get response for software services instance actions [351](#)

get responses for actions for a software services instance [346](#)

get security profile information [160](#)

get software services instances [312](#)

get source information for a software services template [218](#)

get the properties of a z/OSMF workflow [736](#)

get the properties of an archived workflow [787](#)

get the system variables [712](#)

get variables for a software services instance [320](#), [323](#)

get your metered data [522](#), [525](#)

getting an approval for a software services template [236](#)

list domain resource pools [152](#)

List MVS subsystems [402](#)

list resource pools [147](#)

list the archived workflows for a system [785](#)

list the domains for cloud provisioning [86](#)

list the jobs for an owner, prefix, or job ID [655](#)

list the spool files for a job [659](#)

list the tenants for cloud provisioning [103](#)

list the workflows for a system [759](#)

list z/OS data set members [572](#)

list z/OS data sets [569](#)

List z/OS UNIX Filesystems [621](#)

listing approvals for a software services template [239](#)

## GET method (*continued*)

- listing UNIX files and directories [598](#)
- obtain a list of groups [470](#)
- obtain a list of handlers for an event type [29](#)
- obtain a list of portable software instances [453](#)
- obtain a list of software instances [414](#)
- obtain a list of sysplexes [474](#)
- obtain a list of systems [467](#)
- obtain a list of systems in a central processor complex (CPC) [478](#)
- obtain a list of systems in a group [471](#)
- obtain a list of systems in a sysplex [476](#)
- obtain a list of the tasks that are eligible to be handlers [28](#)
- obtain the status of a job [652](#)
- published software services templates [278](#)
- read z/OS data set members [575](#)
- read z/OS UNIX file [603](#)
- receive messages from a TSO/E address space [509](#)
- receive messages from applications running in a TSO/E address space [510](#)
- retrieve a list of groups [470](#)
- retrieve a list of portable software instances [453](#)
- retrieve a list of software instances [414](#)
- retrieve a list of sysplexes [474](#)
- retrieve a list of systems [467](#)
- retrieve a list of systems in a central processor complex (CPC) [478](#)
- retrieve a list of systems in a group [471](#)
- retrieve a list of systems in a sysplex [476](#)
- retrieve a workflow definition [770](#)
- retrieve data from a remote z/OSMF instance [379](#)
- retrieve data from a secondary z/OSMF instance [379](#)
- retrieve data from an application server [35](#)
- retrieve data from one system [379](#)
- retrieve information about z/OSMF [700](#)
- retrieve messages from logs [558](#)
- retrieve persisted user or application data [369](#)
- retrieve persisted z/OSMF settings data [706](#)
- retrieve the contents of a job spool file [661](#)
- retrieve the properties of portable software instances [455](#)
- retrieve the properties of software instances [417](#)
- retrieving data from all the system in a central processor complex (CPC) [379](#)
- retrieving data from all the system in a group [379](#)
- retrieving data from all the system in a sysplex [379](#)
- retrieving data from multiple systems [379](#)
- software services instance names [360](#)
- software services templates [195](#), [211](#)
- view the properties of portable software instances [455](#)
- view the properties of software instances [417](#)

## get source information

- templates
- REST API [218](#)

## H

### handler

- obtaining a list [28](#), [29](#)
- registering [25](#)
- unregistering [30](#)

## I

### IBM z/OS Management Facility

- overview [xlix](#)
- programming interfaces [1](#)
- publications [xlix](#)
- web site [xlix](#)

### IDCAMS

- operations [596](#)

### IDCAMS Access Methods Services

- IDCAMS Access Methods Services [596](#)

### importing plug-ins [963](#)

### instances

- get response for actions
  - REST API [346](#)
- modifying
  - REST API [188](#)
- perform actions
  - REST API [337](#)
- update variables
  - REST API [330](#)

### isCallable attribute [815](#), [840](#)

## J

### job

- cancelling [677](#)
- changing the job class [675](#)
- holding [669](#)
- purging from the JES spool [680](#)
- releasing [672](#)
- submitting [665](#)

### job completed document [684](#)

### job document [682](#), [685](#)

### job file document [686](#)

### job spool file

- retrieving the contents [661](#)

### job spool files list

- obtaining [659](#)

### JOB statement

- variable substitution [833](#)

### job status

- obtaining [652](#)

### job step data document [687](#)

### jobfiles object [754](#)

### jobInfo object [752](#)

### jobs list

- obtaining [655](#)

### jobstatus object [753](#)

### JSON document

- data set list [631](#), [634](#), [637](#)

- data set list with attributes [632](#)

### error report document

- for z/OS data set and file REST interface [640](#)

### job completed document [684](#)

### job document [682](#), [685](#)

### job file document [686](#)

### job step data document [687](#)

### UNIX file list [638–640](#)

### JSON document specifications

- for z/OS data set and file REST interface [631](#)

- for z/OS jobs REST interface [682](#)

### JSON error report document

- error reporting categories

JSON error report document (*continued*)  
error reporting categories (*continued*)  
for z/OS data set and file REST interface [641](#)  
JSON web token [719](#), [721](#), [724](#)

## K

keyboard  
navigation [999](#)  
PF keys [999](#)  
shortcut keys [999](#)

## L

list approvals  
templates  
REST API [239](#)  
List MVS subsystems  
List MVS subsystems [402](#)  
List z/OS UNIX Filesystems  
List z/OS UNIX Filesystems [621](#)  
log in to the z/OSMF server [719](#)

## M

mainframe education [xlix](#)  
messages in logs  
retrieve [558](#)  
modifying  
instances  
REST API [188](#)  
multisystem routing services  
authenticating with a secondary z/OSMF instance [398](#)  
authenticating with an HTTP proxy server [400](#)  
deleting data from a secondary z/OSMF instance [392](#)  
deleting data from all the systems in a central processor complex (CPC) [392](#)  
deleting data from all the systems in a group [392](#)  
deleting data from all the systems in a sysplex [392](#)  
deleting data from multiple systems [392](#)  
deleting data from one system [392](#)  
logging into a remote system [398](#)  
logging into a secondary z/OSMF instance [398](#)  
logging into an HTTP proxy server [400](#)  
overview [374](#)  
retrieving data from a secondary z/OSMF instance [379](#)  
retrieving data from all the systems in a central processor complex (CPC) [379](#)  
retrieving data from all the systems in a group [379](#)  
retrieving data from all the systems in a sysplex [379](#)  
retrieving data from multiple systems [379](#)  
retrieving data from one system [379](#)  
updating data for a secondary z/OSMF instance [386](#)  
updating data for all the systems in a central processor complex (CPC) [386](#)  
updating data for all the systems in a group [386](#)  
updating data for all the systems in a sysplex [386](#)  
updating data for multiple systems [386](#)  
updating data for one system [386](#)  
MVS  
subsystems [402](#)

## N

navigation  
keyboard [999](#)  
Notification RESTful Services  
description [404](#)  
Get all of the notifications received by the current user [405](#)  
The notification is sent from a z/OSMF task or z/OSMF user [409](#)  
The notification is sent from a z/OSMF task, when the content is the message from the bundle file [407](#)

## O

output element [827](#)  
output file  
description [804](#)

## P

portable software instance [983](#)  
POST method  
add a classification rule [70](#)  
add portable software instances [460](#)  
add software instances [428](#)  
add tenant consumer [124](#)  
add tenant description [127](#)  
add tenant groups [129](#)  
approve a software services template [242](#)  
approve a software services template, batch [243](#)  
Archive a software services template [233](#)  
archive a workflow on a z/OS system [783](#)  
assign CPU properties to a tenant [110](#)  
assign memory capping properties to a tenant [112](#)  
assign solution ID [114](#)  
authenticate the user [719](#)  
authenticate with a secondary z/OSMF instance [398](#)  
authenticate with an HTTP proxy server [400](#)  
batch approve a software services template [243](#)  
create a new software services template based on an existing one [185](#)  
create a new version of a software services template [178](#)  
create a software services instance [288](#)  
create a software services template on a z/OS system [170](#)  
Create a UNIX file or directory [610](#)  
create a workflow on a z/OS system [728](#)  
create portable software instances [460](#)  
create software instances [428](#)  
create software service instance names [357](#)  
create system variables [710](#)  
create variable names [362](#)  
create z/OS data sets [585](#)  
Create z/OS UNIX zFS Filesystem [623](#)  
delete WLM resource pool [516](#)  
disable CPU capping [115](#)  
disable memory capping [117](#)  
disable metering [118](#)  
edit data for an application server [39](#)  
enable CPU capping [120](#)  
enable memory capping [121](#)

## POST method (*continued*)

- enable metering [123](#)
- Export a defined software instance [434](#)
- export system variables [715](#)
- import system variables [714](#)
- list software instance data sets [423](#)
- log into a remote system [398](#)
- log into a secondary z/OSMF instance [398](#)
- log into an HTTP proxy server [400](#)
- modify a software services instance [188](#)
- modify data for an application server [39](#)
- obtain a port from a resource pool [59](#)
- obtain a SNA application name from a resource pool [65](#)
- obtain an IP address from a resource pool [51](#)
- perform actions for a software services instance [330](#), [337](#)
- POST a notification [407](#), [409](#)
- prime WLM resource pool [514](#)
- publish a software services template [226](#)
- reconnect to a TSO/E address space [485](#)
- refresh files for software services templates [232](#)
- register an event handler [25](#)
- register an event type [23](#)
- rejecting an approval for a software services template [245](#)
- release a port from a resource pool [63](#)
- release a SNA application name from a resource pool [68](#)
- release an IP address from a resource pool [56](#)
- remove a classification rule [72](#)
- remove tenant consumer [126](#)
- remove tenant groups [130](#)
- start a TSO/E address space [485](#)
- start an application in a TSO/E address space [490](#)
- test a software services template [194](#), [228](#)
- test a software services template on a z/OS system [235](#)
- update data for a remote z/OSMF instance [386](#)
- update data for a secondary z/OSMF instance [386](#)
- update data for an application server [39](#)
- update data for one system [386](#)
- update security state [159](#)
- update system variables [710](#)
- updating data for all the system in a central processor complex (CPC) [386](#)
- updating data for all the system in a group [386](#)
- updating data for all the system in a sysplex [386](#)
- updating data for multiple systems [386](#)
- product information files [979](#)
- programming interface
  - for z/OSMF [1](#)
- property file [963](#)
- public objects
  - defining public objects [932](#)
  - deleting public objects [934](#)
  - retrieving public objects [934](#)
- publish
  - templates
    - REST API [226](#)

## PUT method

- Cancel a job [677](#)
- cancel a workflow on a z/OS system [766](#)
- change a user password or passphrase [721](#)
- change the class of a job [675](#)
- construct a WLM service definition [517](#), [519](#)
- create a tenant [91](#)

## PUT method (*continued*)

- edit data for an application server [39](#)
- edit software instances [440](#)
- holding a job [669](#)
- IDCAMS Access Methods Services [596](#)
- issue a command [528](#)
- modify data for an application server [39](#)
- modify software instances [440](#)
- mount file system [626](#), [629](#)
- persist user or application data [367](#)
- persist user or plug-in z/OSMF settings data [704](#)
- ping a TSO/E address space [507](#)
- releasing a job [672](#)
- retrieve products, features, and FMIDs [446](#)
- send messages to a TSO/E address space [504](#)
- send messages to an application running in a TSO/E address space [506](#)
- start a TSO/E address space [492](#)
- start a workflow on a z/OS system [761](#)
- submit a job [665](#)
- update data for a remote z/OSMF instance [386](#)
- update data for a secondary z/OSMF instance [386](#)
- update data for an application server [39](#)
- update data for one system [386](#)
- update variables for a software services instance [325](#)
- updating data for all the system in a central processor complex (CPC) [386](#)
- updating data for all the system in a group [386](#)
- updating data for all the system in a sysplex [386](#)
- updating data for multiple systems [386](#)
- write to a z/OS data set or member [580](#)
- write to a z/OS UNIX file [607](#)
- z/OS data set and PDS member utilities [592](#)
- z/OS UNIX file Utilities [615](#)

## R

- reject approval
  - templates
    - REST API [245](#)
- Representational State Transfer (REST) interfaces
  - information resources [1](#)
  - introduction [1](#)
- Resource Measurement Facility
  - overview [521](#)
- run as user element [852](#)

## S

- scope attribute [815](#), [840](#)
- sending to IBM
  - reader comments [li](#)
- ServerPac order
  - considerations [xlxi](#)
- shortcut keys [999](#)
- software management services
  - adding portable software instances [460](#)
  - adding software instances [428](#)
  - creating portable software instances [460](#)
  - creating software instances [428](#)
  - deleting portable software instances [464](#)
  - deleting software instances [452](#)
  - editing software instances [440](#)

- software management services (*continued*)
  - listing software instance data sets [423](#)
  - modifying software instances [440](#)
  - obtaining a list of portable software instances [453](#)
  - obtaining a list of software instances [414](#)
  - overview [412](#)
  - removing portable software instances [464](#)
  - removing software instances [452](#)
  - retrieving a list of portable software instances [453](#)
  - retrieving a list of software instances [414](#)
  - retrieving products, features, and FMIDs [446](#)
  - retrieving the properties of portable software instances [455](#)
  - retrieving the properties of software instances [417](#)
  - viewing the properties of portable software instances [455](#)
  - viewing the properties of software instances [417](#)
- software management workflows [991](#)
- software service instance name services [356](#)
- software services instance services
  - delete software services instances [333](#)
  - get [301](#)
  - get responses for actions [351](#)
  - get software services instances [312](#)
  - get variables [320](#), [323](#)
  - update variables [325](#)
- software services template services
  - delete approval [246](#)
  - refresh template files [232](#)
- step element
  - approver sub-element [852](#)
  - autoEnable attribute [846](#)
  - condition attribute [840](#), [850](#)
  - feedback attribute [818](#)
  - run as user sub-element [852](#)
  - suspend element [846](#)
- step-info object [744](#), [792](#)
- subsystem services
  - description [401](#)
- summary of changes
  - z/OSMF Programming Guide [liii](#)
- sysoutDD attribute [828](#)

## T

- templates
  - approving
    - REST API [242](#), [243](#)
  - archiving
    - REST API [233](#)
  - creating new versions
    - REST API [178](#)
  - deleting
    - REST API [194](#)
  - get approval
    - REST API [236](#)
  - get history of
    - REST API [211](#)
  - get properties of
    - REST API [195](#)
  - get source information
    - REST API [218](#)
  - list approvals
    - REST API [239](#)

- templates (*continued*)
  - publish
    - REST API [226](#)
  - reject approval
    - REST API [245](#)
  - test running
    - REST API [228](#)
- test running
  - templates
    - REST API [228](#)
- topology services
  - obtaining a list of groups [470](#)
  - obtaining a list of sysplexes [474](#)
  - obtaining a list of systems [467](#)
  - obtaining a list of systems in a central processor complex (CPC) [478](#)
  - obtaining a list of systems in a group [471](#)
  - obtaining a list of systems in a sysplex [476](#)
  - overview [465](#)
  - retrieving a list of groups [470](#)
  - retrieving a list of sysplexes [474](#)
  - retrieving a list of systems [467](#)
  - retrieving a list of systems in a central processor complex (CPC) [478](#)
  - retrieving a list of systems in a group [471](#)
  - retrieving a list of systems in a sysplex [476](#)
- trademarks [1006](#)
- TSO/E address space services
  - ending a TSO/E address space [511](#)
  - overview [481](#)
  - pinging a TSO/E address space [507](#)
  - placing a TSO/E address space into dormant state [511](#)
  - receiving messages from a TSO/E address space [509](#)
  - receiving messages from applications running in a TSO/E address space [510](#)
  - sending messages to a TSO/E address space [504](#)
  - sending messages to an application running in a TSO/E address space [506](#)
  - starting a TSO/E address space [485](#)
  - starting an application in a TSO/E address space [490](#)

## U

- UNIX file
  - listing [598](#)
  - read [603](#)
  - writing to [607](#)
- UNIX File
  - deleting [613](#)
- UNIX file list [638–640](#)
- UNIX System Services
  - file or directory [615](#)
- user interface
  - ISPF [999](#)
  - TSO/E [999](#)
- Utilities
  - operations [592](#), [615](#)

## V

- variable
  - input file [871](#)
  - internal [864](#)

- variable substitution in the JOB statement [833](#)
- variable value element [821](#)
- variable-info object [755](#), [800](#)
- variable-reference object [754](#), [799](#)
- variables
  - update for instances
    - REST API [330](#)
- versions
  - of templates
    - REST API [178](#)

## W

- WLM
  - resource pool [514](#), [516](#)
  - service definition [517](#), [519](#)
- WLM resource pool
  - delete [516](#)
  - prime [514](#)
- WLM service definition
  - construct [517](#), [519](#)
- workflow
  - terms [804](#)
  - upgradeable [817](#)
- workflow internal variable
  - description [864](#)
- workflow variable input file
  - creating [871](#)
  - description [871](#)
  - variable substitution [872](#)

## Z

- z/OS Basic Skills information center [xlix](#)
- z/OS console REST interface
  - description [513](#), [526](#)
- z/OS data set and PDS member utilities
  - z/OS data set and PDS member utilities [592](#)
- z/OS data sets and files REST interface
  - Create a UNIX file or directory [610](#)
  - creating z/OS data sets [585](#)
  - delete a PDS or PDSE [590](#)
  - delete a UNIX file or directory [613](#)
  - deleting z/OS data sets [588](#)
  - description [563](#)
  - listing UNIX files and directories [598](#)
  - listing z/OS data set members [572](#)
  - listing z/OS data sets [569](#)
  - mount a unix file system [629](#)
  - mount a UNIX file system [626](#)
  - reading z/OS UNIX files [603](#)
  - retrieving z/OS data set members [575](#)
  - writing to data set members [580](#)
  - writing to z/OS UNIX files [607](#)
  - z/OS UNIX file Utilities [615](#)
- z/OS jobs REST interface
  - cancelling a job on z/OS [677](#)
  - changing the class of a job [675](#)
  - description [647](#)
  - holding a job [669](#)
  - listing the jobs for an owner, prefix, or job ID [655](#)
  - listing the spool files for a job [659](#)
  - obtaining the status of a job [652](#)

- z/OS jobs REST interface (*continued*)
  - purging a job from the JES spool [680](#)
  - releasing a job [672](#)
  - retrieving messages from logs on the system [558](#)
  - retrieving the contents of a job spool file [661](#)
  - submitting a job to run on z/OS [665](#)
- z/OS UNIX file
  - write [607](#)
- z/OSMF authentication services
  - description [718](#)
- z/OSMF cloud provisioning resource management services
  - add tenant consumer [124](#)
  - add tenant description [127](#)
  - add tenant groups [129](#)
  - assign CPU properties to a tenant [110](#)
  - assign memory capping properties to a tenant [112](#)
  - assign solution ID [114](#)
  - delete a tenant [109](#)
  - disable CPU capping [115](#)
  - disable memory capping [117](#)
  - disable metering [118](#)
  - enable CPU capping [120](#)
  - enable memory capping [121](#)
  - enable metering [123](#)
  - remove tenant consumer [126](#)
  - remove tenant groups [130](#)
  - update security state [159](#)
- z/OSMF cloud provisioning resource pool services
  - add a classification rule [70](#)
  - obtain a port from a resource pool [59](#)
  - obtain a SNA application name from a resource pool [65](#)
  - obtain an IP address from a resource pool [51](#)
  - release a port from a resource pool [63](#)
  - release a SNA application name from a resource pool [68](#)
  - release an IP address from a resource pool [56](#)
  - remove a classification rule [72](#)
- z/OSMF Cloud Provisioning services [76](#)
- z/OSMF information retrieval service
  - overview [699](#)
  - retrieving information about z/OSMF [700](#)
- z/OSMF Programming Guide
  - content, changed [lvi](#)
  - content, new [liii](#)
  - content, no longer included [lvii](#)
  - summary of changes [liii](#)
- z/OSMF published software services template services
  - get [257](#), [269](#)
  - list [278](#)
- z/OSMF resource management services
  - creating a tenant [91](#)
  - get metered data [522](#)
  - getting a domain [79](#), [84](#)
  - getting a domain resource pool [138](#)
  - getting a resource pool [132](#)
  - getting a resource pool history [145](#)
  - getting a tenant for cloud provisioning [94](#)
  - getting a tenant history for cloud provisioning [100](#)
  - getting data set attributes [74](#)
  - getting security profile information [160](#)
  - getting z/OSMD tsoAPP information [501](#)
  - issue a TSO/E command with z/OSMF REST API [492](#)
  - list the domains for cloud provisioning [86](#)
  - listing domain resource pools [152](#)
  - listing resource pools [147](#)

- z/OSMF resource management services (*continued*)
  - listing the tenants for cloud provisioning [103](#)
- z/OSMF resource measurement services
  - get metered data [525](#)
- z/OSMF settings services
  - overview [702](#)
  - persisting user or plug-in z/OSMF settings data [704](#)
  - retrieving z/OSMF settings data [706](#)
- z/OSMF software services instance name services
  - list [360](#)
- z/OSMF software services instances name services
  - create variable names [362](#)
- z/OSMF software services instances services
  - create [288](#)
  - create names [357](#)
  - delete response for action [354](#)
  - get responses for actions [346](#)
  - modify [188](#)
  - perform actions [337](#)
  - update variables [330](#)
- z/OSMF software services template services
  - approve an approval record [242](#), [243](#)
  - Archiving a software services template [233](#)
  - creating a new software services template based on an existing one [185](#)
  - creating a new version of a software services template [178](#)
  - creating a software services template [226](#)
  - creating a software services template on a z/OS system [170](#)
  - deleting a software services template [194](#)
  - get an approval of a software services template [236](#)
  - get prompt variables [273](#)
  - get source information for software services template [218](#)
  - list approvals for a software services template [239](#)
  - rejecting an approval of a software services template [245](#)
  - testing a software services template [228](#)
  - testing a software services template on a z/OS system [235](#)
- z/OSMF software services templates services
  - get [195](#)
  - get history [211](#)
- z/OSMF system variable services
  - creating system variables [710](#)
  - deleting system variables [717](#)
  - exporting system variables [715](#)
  - getting the system variables [712](#)
  - importing system variables [714](#)
  - overview [709](#)
  - updating system variables [710](#)
- z/OSMF variable services
  - overview [709](#)
- z/OSMF workflow services
  - archiving a workflow on a z/OS system [783](#)
  - cancel a workflow on a z/OS system [766](#)
  - creating a workflow on a z/OS system [728](#)
  - delete a workflow on a z/OS system [768](#)
  - delete an archived workflow on a z/OS system [801](#)
  - get the properties of a z/OSMF workflow [736](#)
  - get the properties of an archived workflow [787](#)
  - listing the archived workflows on a z/OS system [785](#)
  - listing the workflows on a z/OS system [759](#)

- z/OSMF workflow services (*continued*)
  - overview [725](#)
  - retrieving a workflow definition [770](#)
  - starting a workflow on a z/OS system [761](#)





Product Number: 5650-ZOS

SC27-8420-40

