

z/OS



# Cryptographic Services PKI Services Guide and Reference



z/OS



# Cryptographic Services PKI Services Guide and Reference

**Note**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 405.

**Ninth edition, September 2006**

This edition applies to Version 1 Release 8 of z/OS (5694-A01), Version 1 Release 8 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation  
Department 55JA, Mail Station P384  
2455 South Road  
Poughkeepsie, NY 12601-5400  
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCS)

Internet e-mail: [mhvrdfs@us.ibm.com](mailto:mhvrdfs@us.ibm.com)

World Wide Web: [www.ibm.com/servers/eserver/zseries/zos/webqs.html](http://www.ibm.com/servers/eserver/zseries/zos/webqs.html)

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001, 2006. All rights reserved.

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

---

# Contents

<b>Tables</b> . . . . .	xi
<b>Figures</b> . . . . .	xiii
<b>About this document</b> . . . . .	xv
Who should use this document . . . . .	xv
How to use this document . . . . .	xv
Where to find more information . . . . .	xvii
Softcopy publications . . . . .	xvii
RACF courses . . . . .	xviii
Using LookAt to look up message explanations . . . . .	xviii
Using IBM Health Checker for z/OS . . . . .	xix
Other sources of information . . . . .	xix
IBM discussion area . . . . .	xx
Internet sources . . . . .	xx
To request copies of IBM publications . . . . .	xxi
<b>Summary of changes</b> . . . . .	xxiii
<hr/> <b>Part 1. Planning</b> . . . . .	<hr/> 1
<b>Chapter 1. Introducing PKI Services.</b> . . . .	3
What is PKI Services? . . . . .	3
What is a certificate authority?. . . . .	3
What is PKI? . . . . .	4
Basic components of PKI Services and related products . . . . .	4
Component diagram . . . . .	5
Supported standards . . . . .	6
Supported certificate types . . . . .	7
Supported certificate fields and extensions . . . . .	8
<b>Chapter 2. Planning your implementation</b> . . . . .	9
Installing PKI Services . . . . .	9
Requirements for sysplex support . . . . .	9
Determining prerequisite products . . . . .	10
z/OS HTTP Server . . . . .	10
LDAP directory server . . . . .	10
OCSF . . . . .	10
ICSF (optional) . . . . .	10
sendmail (optional) . . . . .	11
OCEP (optional) . . . . .	11
Identifying skill requirements . . . . .	11
Team members . . . . .	11
Skills for setting up prerequisite products . . . . .	12
Skills for setting up PKI Services . . . . .	13
Creating an implementation plan . . . . .	14
Task roadmap for implementing PKI Services. . . . .	14
<b>Chapter 3. Installing and configuring prerequisite products</b> . . . . .	17
Tasks to perform before setting up PKI Services . . . . .	17
Installing and configuring the z/OS HTTP Server . . . . .	17
Steps for installing and configuring the z/OS HTTP Server to work with PKI Services . . . . .	17

Installing and configuring OCSF . . . . .	19
Steps for installing and configuring OCSF to work with PKI Services . . . . .	19
Installing and configuring LDAP . . . . .	20
Steps for installing and configuring LDAP . . . . .	20
Installing and configuring ICSF (optional) . . . . .	22
Configuring sendmail (optional) . . . . .	23

---

## Part 2. Configuring your system for PKI Services . . . . . 25

<b>Chapter 4. Running IKYSETUP to perform RACF administration . . . . .</b>	<b>27</b>
Overview of IKYSETUP. . . . .	27
Before you begin . . . . .	28
Variables whose values must change. . . . .	29
Variables whose values may change depending on setup . . . . .	31
Variables you can optionally change . . . . .	36
Steps for performing RACF tasks using IKYSETUP . . . . .	39
Sample IKYSETUP log data set . . . . .	42

<b>Chapter 5. Configuring the UNIX runtime environment . . . . .</b>	<b>45</b>
Steps for copying files . . . . .	47
Optionally updating PKI Services environment variables. . . . .	47
(Optional) Steps for updating PKI Services environment variables . . . . .	49
Optionally updating the pkiserv.conf configuration file . . . . .	49
(Optional) Steps for updating the configuration file . . . . .	51
Steps for setting up the var directory . . . . .	63

<b>Chapter 6. Tailoring LDAP configuration for PKI Services . . . . .</b>	<b>65</b>
Steps for updating schema.user.ldif . . . . .	65

<b>Chapter 7. Updating z/OS HTTP Server configuration and starting the server . . . . .</b>	<b>67</b>
Steps for updating the z/OS HTTP Server configuration files . . . . .	67
Steps for starting the z/OS HTTP Server . . . . .	70

<b>Chapter 8. Tailoring the PKI Services configuration file for LDAP . . . . .</b>	<b>71</b>
Excerpt of LDAP section . . . . .	71
Storing information for encrypted passwords for your LDAP servers . . . . .	71
Steps for tailoring the LDAP section of the configuration file . . . . .	72

<b>Chapter 9. Creating VSAM data sets . . . . .</b>	<b>77</b>
Planning VSAM storage requirements . . . . .	77
Determining storage needs for ICL . . . . .	78
Determining storage needs for the object store . . . . .	78
(Optional) preliminary steps for establishing VSAM RLS. . . . .	78
Steps for creating the VSAM object store and ICL data sets and indexes . . . . .	79
Steps for creating additional alternate indexes . . . . .	80
(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS . . . . .	81
Tuning VSAM performance . . . . .	82
(Optional) steps for adding VSAM buffer space . . . . .	82

<b>Chapter 10. Starting and stopping PKI Services . . . . .</b>	<b>85</b>
Steps for starting the PKI Services daemon . . . . .	85
Stopping the PKI Services daemon . . . . .	86

---

## Part 3. Customizing PKI Services . . . . . 89

<b>Chapter 11. Customizing the end-user Web application.</b>	91
Contents of the pkiserv.tmpl certificates templates file	91
What are substitution variables?	91
What are named fields?	93
INSERT sections	93
The APPLICATION sections	99
Templates that PKI Services provides	101
TEMPLATE sections	103
Summary of fields in certificate templates.	110
Examining the pkiserv.tmpl file	114
Examining the APPLICATION section	114
Examining the TEMPLATE section	117
Examining the INSERT section	120
Relationship between CGIs and the pkiserv.tmpl file	123
Steps for performing minimal customization	124
Steps for additional first-time customization	126
Steps for retrofitting release changes into the PKI Services certificate templates	129
Locating code for customizing end-user Web pages	130
Steps for adding a new certificate template	132
Changing the runtime user ID	132
Steps for changing the runtime user ID for requesting certificates	133
Steps for changing the runtime user ID for retrieving certificates	134
Customizing e-mail notifications sent to users	135
Steps for customizing e-mail notification forms	137
Customizing the OtherName field.	137
Steps for customizing the sample AltOther_<OID>INSERTs	138
 <b>Chapter 12. Customizing the administration Web pages</b>	 141
CGIs for administration Web pages	141
Customizing the administration Web pages	143
Steps for customizing the administration Web pages	143
Changing the runtime behavior for accessing administration pages	143
Steps for changing control of access to administration pages	144
 <b>Chapter 13. Advanced customization</b>	 145
Scaling for high volume installations	145
Using certificate policies	146
Steps for creating the CertificatePolicies extension on a global basis	147
Steps for creating the CertificatePolicies extension on a template basis	148
Updating the signature algorithm	149
Steps for changing the signature algorithm	149
Customizing distribution point CRLs.	150
Specifying the URI format	151
Determining CRLDistURI <i>n</i>	151
Determining CRLDistDirPath	152
Steps for customizing distribution point CRLs	153
How distribution point CRLs work	155
Using the OCSP responder	156
Creating a distribution point ARL	156
Adding an application domain	158
Steps for creating multiple application sections in the PKI Services template file	159
Steps for adding application domains to the Web server configuration files	160

	Adding a new CA domain . . . . .	161
	Task overview . . . . .	162
	Task roadmap for adding CA domains . . . . .	163
	Recording your progress adding CA domains . . . . .	164
	Subtask 1: Steps for planning additional CA domains . . . . .	164
	Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains . . . . .	166
	Subtask 3: Steps for running the IKYSETUP exec . . . . .	168
	Subtask 4: Steps for configuring the UNIX environment . . . . .	170
	Subtask 5: Steps for updating the PKI Services template file . . . . .	171
	Subtask 6: Steps for updating the Web server configuration . . . . .	173
	Subtask 7: Steps for allocating VSAM data sets . . . . .	173
	Subtask 8: Steps for starting PKI Services . . . . .	174
	Enabling Simple Certificate Enrollment Protocol (SCEP) . . . . .	175
	Overview of SCEP preregistration . . . . .	175
	Overview of certificate request processing for preregistered SCEP clients . . . . .	176
	Checking certificate fingerprints . . . . .	178
	Steps for enabling Simple Certificate Enrollment Protocol (SCEP). . . . .	178
	Using the PKI exit . . . . .	180
	Steps for updating the exit code sample . . . . .	181
	Using the exit for pre- and post-processing . . . . .	181
	Scenarios for using the PKI exit . . . . .	191

---

## **Part 4. Using PKI Services . . . . . 195**

	<b>Chapter 14. Using the end-user Web pages . . . . .</b>	<b>197</b>
	Steps for accessing the end-user Web pages . . . . .	198
	Summary of fields . . . . .	200
	Steps for requesting a new certificate . . . . .	204
	Retrieving your certificate . . . . .	208
	Steps for retrieving your certificate from the bookmarked Web page . . . . .	209
	Steps for retrieving your certificate from the PKI Services home page . . . . .	210
	Steps for renewing a certificate . . . . .	210
	Steps for revoking or suspending a certificate . . . . .	213
	Steps for preregistering a SCEP client . . . . .	214
	 <b>Chapter 15. Using the administration Web pages. . . . .</b>	 <b>217</b>
	Steps for accessing the administration home page . . . . .	217
	Fields in the administration Web pages . . . . .	220
	Processing certificate requests . . . . .	220
	Status of certificate requests . . . . .	220
	Actions on certificate requests . . . . .	221
	Using the PKI Services administration home page . . . . .	221
	Processing certificates. . . . .	231
	Status of certificates . . . . .	231
	Actions for certificates . . . . .	232
	Steps for processing a single certificate . . . . .	232
	Steps for processing certificates by performing searches . . . . .	234
	Relationship between certificate requests and matching certificates . . . . .	238

---

## **Part 5. Administering security for PKI Services. . . . . 239**

	<b>Chapter 16. RACF administration for PKI Services . . . . .</b>	<b>241</b>
	Authorizing users for the PKI Services administration group . . . . .	241
	Connecting members to the group . . . . .	241



	Deleting members from groups . . . . .	241
	Authorizing users for inquiry access. . . . .	241
	Steps for authorizing users for inquiry access . . . . .	242
	Administering HostIdMappings extensions . . . . .	242
	Steps for administering HostIdMappings extensions . . . . .	243
I	Locating your PKI Services certificates and key ring. . . . .	244
I	Steps for locating the PKI Services certificates and key ring . . . . .	244
	Establishing PKI Services as an intermediate CA . . . . .	246
	Steps for establishing PKI Services as an intermediate CA . . . . .	247
I	Renewing your PKI Services CA and RA certificates. . . . .	248
	Steps for renewing your PKI Services CA certificate . . . . .	249
I	Steps for renewing your PKI Services RA certificate . . . . .	250
	Recovering a CA certificate profile . . . . .	251
	Steps for recovering a CA certificate profile . . . . .	251
	Retiring and replacing the PKI Services CA private key . . . . .	254
	Steps to retire and replace the PKI Services CA private key for the PKI templates . . . . .	254
	Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 1 . . . . .	255
	Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 2 . . . . .	256
	R_PKIServ (IRRSPX00) callable service . . . . .	257
	Authorizing end-user functions. . . . .	257
	Authorizing administrative functions . . . . .	259
	Using encrypted passwords for LDAP servers . . . . .	260
	Steps for using encrypted passwords . . . . .	261
<hr/> <b>Part 6. Using the certificate validation service . . . . .</b>		<b>263</b>
	<b>Chapter 17. PKI Services Trust Policy (PKITP) . . . . .</b>	<b>265</b>
	Overview of PKITP . . . . .	265
	Certificate policies . . . . .	267
	Checking certificate status with PKITP . . . . .	267
	Certificate extensions . . . . .	268
	CRL extensions and CRL entry extensions . . . . .	269
	Files for PKITP . . . . .	269
	Configuring and getting started with PKITP . . . . .	269
	Steps for configuring PKITP. . . . .	270
	Trust Policy API . . . . .	270
	CSSM_TP_PassThrough. . . . .	271
	Building the sample application to invoke the certificate validation service . . . . .	274
<hr/> <b>Part 7. Troubleshooting . . . . .</b>		<b>287</b>
	<b>Chapter 18. Using information from SYS1.LOGREC. . . . .</b>	<b>289</b>
	Sample LOGREC data . . . . .	292
	<b>Chapter 19. Using information from the PKI Services logs . . . . .</b>	<b>295</b>
	Viewing SYSOUT information . . . . .	295
	_PKISERV_MSG_LEVEL subcomponents and message levels. . . . .	299
	Changing logging options . . . . .	299
	Displaying log options settings. . . . .	300
	<b>Chapter 20. Using PKI Services utilities . . . . .</b>	<b>301</b>
I	Before you use a PKI Services utility . . . . .	301

Using the iclview utility . . . . .	302
Using the vosview utility . . . . .	305
Using the pkiprereg utility . . . . .	309

---

<b>Part 8. Reference information . . . . .</b>	<b>313</b>
--	------------

<b>Chapter 21. Messages . . . . .</b>	<b>315</b>
---------------------------------------	------------

<b>Chapter 22. File directory structure . . . . .</b>	<b>341</b>
Product libraries . . . . .	341
File system directory and subdirectories . . . . .	341

<b>Chapter 23. The pkiserv.conf configuration file . . . . .</b>	<b>343</b>
--	------------

<b>Chapter 24. Environment variables . . . . .</b>	<b>347</b>
Environment variables in the environment variables file . . . . .	347
The pkiserv.envars environment variables file . . . . .	349

<b>Chapter 25. The IKYSETUP REXX exec. . . . .</b>	<b>351</b>
Actions IKYSETUP performs by issuing RACF commands . . . . .	351
Setting up the PKI Services daemon user ID . . . . .	351
Setting up access control to protect PKI Services. . . . .	351
Establishing your CA and RA certificates . . . . .	353
Configuring the z/OS HTTP Server for SSL mode . . . . .	355
Using RACF to obtain a certificate for the Web server . . . . .	356
Enabling the z/OS HTTP Server for surrogate operation . . . . .	356
Enabling the PKI Services daemon to call OCSF functions . . . . .	356
IKYSETUP sample . . . . .	357

<b>Chapter 26. Other code samples . . . . .</b>	<b>371</b>
z/OS HTTP Server configuration directives . . . . .	371
IKYCVSAM. . . . .	373
IKYMVSAM. . . . .	377
IKYRVSAM. . . . .	380
PKISERVD sample procedure to start PKI Services daemon . . . . .	384

<b>Chapter 27. SMF recording . . . . .</b>	<b>385</b>
PKI Services event code . . . . .	385
Relocate section variable data. . . . .	385

<b>Appendix A. LDAP directory server requirements . . . . .</b>	<b>387</b>
---	------------

<b>Appendix B. Using a gskkyman key database for your certificate store . . . . .</b>	<b>389</b>
Steps for using a gskkyman key database for your certificate store . . . . .	389

<b>Appendix C. Configuring PKI Services as an IdenTrust certificate authority. . . . .</b>	<b>391</b>
Who should use this appendix. . . . .	391
Related information from IdenTrust . . . . .	391
Overview of configuring z/OS PKI Services as a CA. . . . .	391
System prerequisites . . . . .	392
Task overview. . . . .	392
Configuring z/OS PKI Services as a CA . . . . .	393
Steps to modify pkiserv.conf for different certificate types . . . . .	394
Steps to modify pkiserv.conf general settings . . . . .	395
Steps to create IdenTrust specific certificate templates. . . . .	395

Code samples. . . . .	396
Sample PKI Services configuration file directives for IdenTrust compliance	397
Sample browser certificate template for IdenTrust compliance . . . . .	397
Sample server certificate template for IdenTrust compliance . . . . .	400
<b>Appendix D. Accessibility . . . . .</b>	<b>403</b>
Using assistive technologies . . . . .	403
Keyboard navigation of the user interface. . . . .	403
z/OS information. . . . .	403
<b>Notices . . . . .</b>	<b>405</b>
Programming interface information . . . . .	406
Trademarks. . . . .	407
<b>Bibliography . . . . .</b>	<b>409</b>
<b>Index . . . . .</b>	<b>411</b>



# Tables

1.	Basic components of PKI Services and related products . . . . .	4
2.	Types of certificates you can request . . . . .	7
3.	File system directory variables . . . . .	9
4.	Tasks and skills needed for installing prerequisite products . . . . .	12
5.	Roles, tasks, and skills for setting up PKI Services . . . . .	13
6.	Task roadmap for implementing PKI Services . . . . .	14
7.	z/OS HTTP Server information you need to record . . . . .	19
8.	OCSF information you need to record . . . . .	20
9.	LDAP information you need to record . . . . .	21
10.	IKYSETUP—Structure and divisions. . . . .	28
11.	IKYSETUP variables whose values must change . . . . .	29
12.	Decision table for key_backup . . . . .	32
13.	Decision table for key_type . . . . .	32
14.	Decision table for restrict_surrog . . . . .	33
15.	Decision table for unix_sec . . . . .	34
16.	IKYSETUP variables you might want to change depending on setup . . . . .	34
17.	IKYSETUP variables you can optionally change . . . . .	36
18.	Deciding which files to copy and change . . . . .	45
19.	Information needed for updating the configuration file . . . . .	51
20.	LDAP information you need for tailoring LDAP configuration . . . . .	65
21.	Summary of configuration and usage of each Web server instance . . . . .	67
22.	LDAP information you need for tailoring z/OS HTTP Server configuration . . . . .	68
23.	Information needed for updating the LDAP section of the configuration file. . . . .	72
24.	VSAM RLS information you need to record . . . . .	79
25.	Structure and main divisions of the certificate template file (pkiserv.tmpl) . . . . .	91
26.	Substitution variables . . . . .	92
27.	Sample INSERTs. . . . .	94
28.	Named fields in INSERT sections. . . . .	95
29.	Subsections of the APPLICATION sections. . . . .	100
30.	Certificate templates PKI Services provides . . . . .	101
31.	Names, aliases, and nicknames of certificate templates . . . . .	104
32.	Summary of subsections in certificate templates . . . . .	109
33.	Summary of fields for PKI browser certificate templates . . . . .	110
34.	Summary of fields for PKI server certificate templates . . . . .	111
35.	Summary of fields for SAF and SCEP certificate templates . . . . .	113
36.	CGI actions for end-user Web pages . . . . .	123
37.	Location of code for various Web pages. . . . .	130
38.	Descriptions of variables for forms . . . . .	136
39.	Summary of substitution variables in forms. . . . .	137
40.	CGI actions for administrative Web pages . . . . .	141
41.	Task roadmap for creating multiple application domains . . . . .	159
42.	Task roadmap for adding a new CA domain . . . . .	164
43.	Multiple CA domains: Worksheet #1 for recording progress adding new CA domains . . . . .	164
44.	Multiple CA domains: Worksheet #2 for planning your domain names . . . . .	166
45.	Multiple CA domains: Worksheet #3 for planning your RACF identifiers, z/OS UNIX identifiers, and VSAM data set names . . . . .	166
46.	IKYSETUP log information you need to record . . . . .	169
47.	Information you need to enable Simple Certificate Enrollment Protocol (SCEP) . . . . .	178
48.	Summary of information about important files for the exit routine . . . . .	181
49.	Values of arguments for pre- and post-processing . . . . .	182
50.	Types of certificates you can request . . . . .	197
51.	Summary of fields in end-user Web pages . . . . .	201
52.	Summary of fields in the administration pages . . . . .	220

53.	Statuses of certificate requests . . . . .	220
54.	Summary of actions to perform on requests and required status . . . . .	221
55.	Searches to display certificate requests . . . . .	227
56.	Status of certificates . . . . .	231
57.	Summary of actions to perform and required status to do so . . . . .	232
58.	Searches to display certificates . . . . .	234
59.	Information you need for locating your PKI Services certificates and key ring . . . . .	244
60.	Information you need for establishing PKI Services as an intermediate CA . . . . .	247
61.	Information you need for renewing your PKI Services certificate authority certificate . . . . .	249
I 62.	Information you need for renewing your PKI Services RA certificate. . . . .	250
63.	Information you need for recovering a CA certificate profile . . . . .	251
64.	Summary of access authorities required for PKI Services requests . . . . .	258
65.	Sequence of validation stages for PKITP certificate revocation checking . . . . .	268
66.	Summary of information about important files for PKITP . . . . .	269
67.	PKI Services OCSF Trust Policy (PKITP) error codes . . . . .	273
68.	LOGREC data for PKI Services . . . . .	289
69.	Names, aliases, and nicknames of certificate templates . . . . .	304
70.	Names, aliases, and nicknames of certificate templates . . . . .	307
I 71.	List of valid field names for use in the preregistration record as input to the pkiprereg utility . . . . .	310
72.	Summary of information about important files . . . . .	313
73.	Meaning of fourth character in message number. . . . .	315
74.	Meaning of eighth character in message number . . . . .	315
75.	Files contained in subdirectories. . . . .	341
76.	Access required if you plan to have an administrator approve certificate requests . . . . .	352
77.	Access required if you plan to use auto-approval . . . . .	353
78.	FACILITY class access needed for protecting administrative functions . . . . .	353
I 79.	Access PKISRVD needs to use RACF certificates . . . . .	355
80.	SMF event code and event code qualifier for PKI Services . . . . .	385
81.	SMF data elements of the extended-length relocate section for PKI Services . . . . .	385
82.	LDAP objectclasses and attributes that PKI Services sets . . . . .	387
83.	Relationship of named fields to LDAP attributes and object identifiers . . . . .	388

# Figures

1.	Component diagram of a typical PKI Services system. . . . .	6
2.	Flowchart of the process of updating IKYSETUP . . . . .	40
3.	Partial listing of the AltOther_1_2_3_4_6 sample INSERT showing the lines you are most likely to customize . . . . .	140
4.	A sample CRLDistributionPoints extension for a certificate authority (CA) certificate . . . . .	158
5.	Illustration of two CA domains—one for employees and one for customers—administered by a single shared administrator who administers both domains . . . . .	162
6.	PKI Services end-user home page for certificate generation . . . . .	199
7.	The certificate popup window for installing the CA certificate . . . . .	200
8.	One-year SSL browser certificate request form . . . . .	205
9.	Supplying the PKCS #10 certificate request for a server or device certificate . . . . .	206
10.	Successful request displays transaction ID . . . . .	207
11.	Web page to retrieve your certificate . . . . .	208
12.	Browser certificate installation Web page . . . . .	209
13.	Server certificate installation Web page . . . . .	209
14.	Popup window listing certificates. . . . .	211
15.	Renew or revoke a certificate Web page. . . . .	212
16.	SCEP preregistration request form . . . . .	214
17.	Successful preregistration request . . . . .	215
18.	PKI Services administration start page . . . . .	218
19.	The certificate popup window for installing the CA certificate . . . . .	219
20.	Entering your user ID and password . . . . .	219
21.	PKI Services administration home page . . . . .	222
22.	Single request approval Web page. . . . .	223
23.	Processing successful Web page . . . . .	224
24.	Restriction note on the modify and approve request Web page . . . . .	224
25.	Modifying the request Web page . . . . .	225
26.	Processing requests after searching . . . . .	228
27.	Request processing was successful Web page . . . . .	230
28.	Request processing was not successful Web page . . . . .	230
29.	Request processing was partially successful Web page . . . . .	231
30.	Processing a certificate from the single certificate Web page . . . . .	233
31.	Processing certificates using searches . . . . .	235
32.	Processing of certificate was successful Web page. . . . .	237
33.	Request processing was not successful Web page . . . . .	237
34.	Request processing was partially successful Web page . . . . .	238
35.	Sample JCL data set for restoring the certificate serial number incrementer value . . . . .	253
36.	Examples of organizations, certificates, and chains . . . . .	266
37.	Sample LOGREC data . . . . .	293
38.	Separating the job files . . . . .	296
39.	Selecting a file to view . . . . .	297
40.	Messages contained in the file . . . . .	298





---

## About this document

This document supports z/OS® (5694-A01) and z/OS.e (5655-G52). This document contains information about planning, customizing, administering, and using the PKI Services component of the z/OS Cryptographic Services.

PKI Services provides a certificate authority for the z/OS environment and enables you to issue and administer digital certificates, so that you do not have to purchase them from an external certificate authority. This document provides you with the information you need to become productive with PKI Services. It discusses the following topics:

- Procedures for setting up PKI Services on the z/OS platform.
- Using the PKI Services administration and user Web pages, you can easily issue digital certificates to trusted parties and control whether or not a certificate is renewed or revoked.
- Guidelines to help you plan for PKI Services, such as how to integrate PKI Services components with other products installed at your site.

---

## Who should use this document

This document should be used by those who plan, install, customize, administer, and use PKI Services. It should also be used by those who install, configure, or provide support in the following areas:

- Lightweight Directory Access Protocol (LDAP)
- Open Cryptographic Services Facility (OCSF)
- Resource Access Control Facility (RACF)
- z/OS
- z/OS HTTP Server
- z/OS UNIX System Services
- (optional) Integrated Cryptographic Service Facility (ICSF)
- (optional) Open Cryptographic Enhanced Plug-ins (OCEP)
- (optional) z/OS Communications Server's sendmail utility

This document assumes that you have experience with installing and configuring products in a network environment. You should be knowledgeable about the following concepts and protocols:

- Hardware installation and configuration
- Internet communications protocols, in particular Transmission Control Protocol/Internet Protocol (TCP/IP) and Secure Sockets Layer (SSL)
- Public key infrastructure (PKI) technology, including directory schemas, the X.509 version 3 standard, and the Lightweight Directory Access Protocol (LDAP)

---

## How to use this document

This document contains several parts:

- Part 1, "Planning," on page 1 includes the following chapters:
  - Chapter 1, "Introducing PKI Services," on page 3 introduces PKI Services, describing its basic components and related products. It also describes supported standards, certificate types, fields and extensions.

- Chapter 2, “Planning your implementation,” on page 9 provides a planning overview for your implementation. It discusses the components that work with PKI Services and the team members you will need to implement PKI Services and the skills they will need.
- Chapter 3, “Installing and configuring prerequisite products,” on page 17 describes installing and configuring related products: the z/OS HTTP Server, OCSF, LDAP, and optionally ICSF.
- Part 2, “Configuring your system for PKI Services,” on page 25 describes the tasks your team members need to perform to configure PKI Services.
  - Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 describes how the RACF® administrator updates and runs IKYSETUP, a REXX exec to perform RACF administration tasks, such as setting up the daemon user ID and giving accesses.
  - Chapter 5, “Configuring the UNIX runtime environment,” on page 45 explains UNIX® programmer tasks including how to copy files, update environment variables, update the PKI Services configuration file, and set up the /var/pkiserv file system directory.
  - Chapter 6, “Tailoring LDAP configuration for PKI Services,” on page 65 explains how the LDAP programmer updates LDAP configuration for PKI Services.
  - Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 67 explains how the Web server programmer updates the z/OS HTTP Server configuration files and starts the z/OS HTTP Server.
  - Chapter 8, “Tailoring the PKI Services configuration file for LDAP,” on page 71 explains how the UNIX programmer updates the **LDAP** section of the PKI Services configuration file.
  - Chapter 9, “Creating VSAM data sets,” on page 77 explains how the MVS programmer creates VSAM data sets.
  - Chapter 10, “Starting and stopping PKI Services,” on page 85 explains how the MVS programmer starts and stops the PKI Services daemon.
- Part 3, “Customizing PKI Services,” on page 89 explains how to customize end-user and administration Web pages and advanced customization using an exit.
  - Chapter 11, “Customizing the end-user Web application,” on page 91 provides an overview of the pkiserv.tpl file, which contains the certificate templates, and explains how to customize the end-user Web pages.
  - Chapter 12, “Customizing the administration Web pages,” on page 141 provides an overview of the CGI scripts and explains how to customize the administration Web pages.
  - Chapter 13, “Advanced customization,” on page 145 explains how to use certificate policies, the signature algorithm, and the PKI exit.
- Part 4, “Using PKI Services,” on page 195 explains using the end-user and administration Web pages.
  - Chapter 14, “Using the end-user Web pages,” on page 197 shows the end-user Web pages and explains how to request a certificate, obtain the certificate, and renew or revoke a certificate.
  - Chapter 15, “Using the administration Web pages,” on page 217 shows the administration Web pages and explains how to process certificate requests and certificates.

- Part 5, “Administering security for PKI Services,” on page 239 explains how to perform many RACF administration tasks needed for PKI Services, such as authorizing users, administering extensions, locating your PKI Services certificate and key ring, and so on.
- Part 6, “Using the certificate validation service,” on page 263 describes how to setup and use the PKI Services Trust Policy (PKITP) plug-in for OCSF.
- Part 7, “Troubleshooting,” on page 287 explains using logs and utilities:
  - Chapter 18, “Using information from SYS1.LOGREC,” on page 289 describes SYS1.LOGREC — which is used to record unusual runtime events, such as an exception.
  - Chapter 19, “Using information from the PKI Services logs,” on page 295 discusses using the PKI Services logs to debug problems and explains how to change logging options and display log options settings.
  - Chapter 20, “Using PKI Services utilities,” on page 301 explains how to use PKI Services utilities: vosview displays the entries in the VSAM ObjectStore data set (request database), and iclview displays the entries in the issued certificate list (ICL).
- Part 8, “Reference information,” on page 313 provides reference information including messages and important code samples.
  - Chapter 21, “Messages,” on page 315 explains PKI Services messages.
  - Chapter 22, “File directory structure,” on page 341 describes product and file system directories for PKI Services and files contained in them.
  - Chapter 23, “The pkiserv.conf configuration file,” on page 343 provides a code sample of the pkiserv.conf configuration file.
  - Chapter 24, “Environment variables,” on page 347 explains the pkiserv.envars environment variables file and provides a code sample.
  - Chapter 25, “The IKYSETUP REXX exec,” on page 351 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
  - Chapter 26, “Other code samples,” on page 371 provides additional code samples.
- There are several appendixes, including the following:
  - Appendix A, “LDAP directory server requirements,” on page 387 explains using a non-z/OS LDAP server.
  - Appendix B, “Using a gskkyman key database for your certificate store,” on page 389 explains an alternative method for setting up your key database.
  - Appendix C, “Configuring PKI Services as an IdenTrust™ certificate authority,” on page 391 explains configuring PKI Services to operate as an IdenTrust compliant certificate authority (CA).

---

## Where to find more information

Where necessary, this document references information in other documents. For complete titles and order numbers for all elements of z/OS, see *z/OS Information Roadmap*.

## Softcopy publications

The PKI Services and RACF libraries are available on the following CD-ROMs. The CD-ROM online library collections include Softcopy Reader™, which is a program that enables you to view the softcopy documents.

**SK3T-4269**      *z/OS Version 1 Release 8 Collection*

## Preface

This collection contains the set of unlicensed documents for the current release of z/OS in both BookManager® and Portable Document Format (PDF) files. You can view or print the PDF files with an Adobe reader.

**SK3T-4272**     *z/OS Security Server RACF Collection*

This softcopy collection kit contains both BookManager and Portable Document Format (PDF) files. You can view or print the PDF files with an Adobe reader.

**SK2T-2180**     *Online Library OS/390 Security Server RACF Information Package*

This softcopy collection contains the Security Server library for OS/390. It also contains the RACF/MVS Version 2 product libraries, the RACF/VM 1.10 product library, product documents from the OS/390 and VM collections, International Technical Support Organization (ITSO) documents (known as Redbooks™), and Washington System Center (WSC) documents (known as orange books) that contain information related to RACF. The collection does not contain any licensed publications. By using this CD-ROM, you have access to RACF-related information from IBM products such as OS/390®, VM/ESA®, CICS TS®, and NetView®.

**SK3T-7876**     *IBM @server zSeries™ Redbooks Collection*

This softcopy collection contains a set of documents called Redbooks that pertain to zSeries subject areas ranging from e-business application development and enablement to hardware, networking, Linux™, solutions, security, Parallel Sysplex® and many others.

**SK2T-2177**     *IBM Redbooks S/390® Collection*

This softcopy collection contains a set of documents called Redbooks that pertain to S/390 subject areas ranging from application development and enablement to hardware, networking, security, Parallel Sysplex and many others.

## RACF courses

The following RACF classroom courses are available in the United States:

<b>H3917</b>	<i>Basics of z/OS RACF Administration</i>
<b>H3927</b>	<i>Effective RACF Administration</i>
<b>ES885</b>	<i>Exploiting the Advanced Features of RACF</i>
<b>ES840</b>	<i>Implementing RACF Security for CICS</i>

IBM provides a variety of educational offerings for RACF. For more information about classroom courses and other offerings, do any of the following:

- See your IBM representative
- Call 1-800-IBM-TEACH (1-800-426-8322)

## Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from these locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at [www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/](http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/).
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX System Services).
- Your Microsoft® Windows® workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T-4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from [www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html](http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html) with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T-4271).
- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

## Using IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework. This book refers to checks or messages associated with this component.

For additional information about checks and about IBM Health Checker for z/OS, see *IBM Health Checker for z/OS: User's Guide*. Starting with z/OS V1R4, z/OS users can obtain the IBM Health Checker for z/OS from the z/OS Downloads page at [www.ibm.com/servers/eserver/zseries/zos/downloads/](http://www.ibm.com/servers/eserver/zseries/zos/downloads/).

SDSF also provides functions to simplify the management of checks. See *z/OS SDSF Operation and Customization* for additional information.

---

## Other sources of information

IBM provides customer-accessible discussion areas where PKI Services and RACF may be discussed by customer and IBM participants. Other information is also available through the Internet.

## Preface

## IBM discussion area

IBM provides the *ibm.servers.mvs.racf* newsgroup for discussion of PKI Services and RACF-related topics. You can find this newsgroup on news (NNTP) server *news.software.ibm.com* using your favorite news reader client.

## Internet sources

The following resources are available through the Internet to provide additional information about PKI Services, RACF, and many other security-related topics:

- **Online library**

To view and print online versions of the z/OS publications, use this address:

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

- **Redbooks**

The Redbooks that are produced by the International Technical Support Organization (ITSO) are available at the following address:

<http://www.ibm.com/redbooks/>

- **Enterprise systems security**

For more information about security on the zSeries platform and z/OS, use this address:

<http://www.ibm.com/servers/eserver/zseries/zos/security/>

- **PKI Services home page**

You can visit the PKI Services home page on the World Wide Web using the following address. Check this site for updates regarding PKI Services.

<http://www.ibm.com/servers/eserver/zseries/zos/pki/>

- **RACF home page**

You can visit the RACF home page on the World Wide Web using the following address.

<http://www.ibm.com/servers/eserver/zseries/zos/racf/>

- **RACF-L discussion list**

Customers and IBM participants may also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.

To subscribe to the RACF-L discussion and receive postings, send a note to:

[listserv@listserv.uga.edu](mailto:listserv@listserv.uga.edu)

Include the following line in the body of the note, substituting your first name and last name as indicated:

`subscribe racf-l first_name last_name`

To post a question or response to RACF-L, send a note, including an appropriate Subject: line, to:

[racf-l@listserv.uga.edu](mailto:racf-l@listserv.uga.edu)

- **RACF sample code**

You can get sample code, internally developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.

To access this code from a Web browser, go to the RACF home page and select the “Downloads” topic from the navigation bar, or go to <ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/>.

The code is also available from [ftp.software.ibm.com](ftp://ftp.software.ibm.com) through anonymous FTP. To get access:



1. Log in as user **anonymous**.
2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:

```
cd eserver/zseries/zos/racf/
```

An announcement will be posted on RACF-L discussion list and on newsgroup *ibm.servers.mvs.racf* whenever something is added.

**Note:** Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using `ftp.software.ibm.com` because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:

- Try to get access by using a Web browser and the links from the RACF home page.
- Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.
- If your FTP client has configuration parameters for the type of remote system, configure it as UNIX instead of MVS.

#### Restrictions

Because the sample code and tools are not officially supported,

- There are no guaranteed enhancements.
- No APARs can be accepted.

---

## To request copies of IBM publications

Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 8:30 a.m. through 5:00 p.m. Eastern Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates





---

# Summary of changes

## Summary of changes for SA22-7693-08 z/OS Version 1 Release 8

This document contains information previously presented in SA22-7693-06 and SA22-7693-07, which support z/OS Version 1 Release 7.

### New information

- “Adding a new CA domain” on page 161
- “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175
- “Steps for preregistering a SCEP client” on page 214
- “Steps for renewing your PKI Services RA certificate” on page 250
- “Using the pkiprereg utility” on page 309
- Chapter 21, “Messages,” on page 315 contains the following new messages:
  - IKYC046I through IKYC060I, inclusively.
  - IKYP032I through IKYP033I, inclusively
  - IKYP034E
  - IKYU001I through IKYU017I, inclusively

### Updated information

- Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27
- Chapter 5, “Configuring the UNIX runtime environment,” on page 45
- Chapter 10, “Starting and stopping PKI Services,” on page 85
- Chapter 11, “Customizing the end-user Web application,” on page 91
- “Summary of fields in certificate templates” on page 110
- “Examining the pkiserv.tmpl file” on page 114
- Chapter 12, “Customizing the administration Web pages,” on page 141
- Chapter 15, “Using the administration Web pages,” on page 217
- “Processing certificate requests” on page 220
- Chapter 16, “RACF administration for PKI Services,” on page 241
- “Locating your PKI Services certificates and key ring” on page 244
- “Steps for renewing your PKI Services CA certificate” on page 249 is improved.
- Chapter 20, “Using PKI Services utilities,” on page 301
- Table 73 on page 315 is updated with information about the **U** character (UTILITY), a new value for the fourth character of PKI Services message identifiers.
- Chapter 21, “Messages,” on page 315 contains updated information for the following messages:
  - IKYI002I
  - IKYP001E
  - IKYP025I
  - IKYP026E
  - IKYS013I through IKYS017I, inclusively
- The following code samples were updated:
  - Chapter 23, “The pkiserv.conf configuration file,” on page 343
  - “The pkiserv.envvars environment variables file” on page 349
  - “IKYSETUP sample” on page 357

- “IKYCVSAM” on page 373
- “IKYMVSAM” on page 377
- “IKYRVSAM” on page 380
- Chapter 25, “The IKYSETUP REXX exec,” on page 351
- Appendix A, “LDAP directory server requirements,” on page 387 is updated to support Simple Certificate Enrollment Protocol (SCEP).

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

### **Summary of changes for SA22-7693-07 z/OS Version 1 Release 7**

This document contains information previously presented in SA22-7693-06, which supports z/OS Version 1 Release 7.

#### **New information**

- Appendix C, “Configuring PKI Services as an IdenTrust™ certificate authority,” on page 391

#### **Updated information**

- The information about the `PostInterval` parameter in the **LDAP** section of `pkiserv.conf` is updated in Table 23 on page 72 to support APAR OA12909.

This document includes terminology, maintenance, and editorial changes.

### **Summary of changes for SA22-7693-06 z/OS Version 1 Release 7**

This document contains information previously presented in SA22-7693-05, which supports z/OS Version 1 Release 6.

#### **New information**

- “Customizing the OtherName field” on page 137
- “Using the OCSP responder” on page 156
- “Creating a distribution point ARL” on page 156
- “Checking certificate status with PKITP” on page 267
- Chapter 21, “Messages,” on page 315 contains the following new messages:
  - IKYC040I
  - IKYC041I
  - IKYC042I
  - IKYC043I
  - IKYC044I
  - IKYC045I
  - IKYP031E

#### **Updated information**

- Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27
- Chapter 5, “Configuring the UNIX runtime environment,” on page 45
- Chapter 11, “Customizing the end-user Web application,” on page 91

- Chapter 13, “Advanced customization,” on page 145
  - “Updating the signature algorithm” on page 149
  - “Customizing distribution point CRLs” on page 150
  - “Using the PKI exit” on page 180
- “Code sample of the PKITP program (pkitsamp.c)” on page 275
- Chapter 21, “Messages,” on page 315 contains an updated system programmer response in support of APAR OA11109 for message IKYC005I, on page 316.
- Chapter 23, “The pkiserv.conf configuration file,” on page 343
- Chapter 25, “The IKYSETUP REXX exec,” on page 351
- Chapter 27, “SMF recording,” on page 385

This document includes terminology, maintenance, and editorial changes.

**Summary of changes  
for SA22-7693-05  
z/OS Version 1 Release 6**

This document contains information previously presented in SA22-7693-04, which supports z/OS Version 1 Release 5.

**New information**

- A new topic, “Retiring and replacing the PKI Services CA private key” on page 254, was added to Chapter 16, “RACF administration for PKI Services.”

**Updated information**

- Updated explanation and system programmer response for the following message:
  - IKYP028E, on page 333.

This document includes terminology, maintenance, and editorial changes.

**Summary of changes  
for SA22-7693-04  
z/OS Version 1 Release 5**

This document contains information previously presented in SA22-7693-03, which supports z/OS Version 1 Release 5.

**Changed information**

- Support for APAR OA06817

This document includes terminology, maintenance, and editorial changes.

**Summary of changes  
for SA22-7693-03  
z/OS Version 1 Release 5**

This document contains information previously presented in SA22-7693-02, which supports z/OS Version 1 Release 4.

**New information**

- Additional support for PKIX compliance, including:

- A new certificate template for two-year PKI Authenticode—code signing server certificates
- Support for X.509 certificate extension, ExtKeyUsage
- New values for the X.509 KeyUsage extension
- Support to specify which certificate extensions are marked critical
- Support to apply certificate policies on a certificate-template basis
- Support for third-party OCSP (Online Certificate Status Protocol) suppliers
- Support to temporarily suspend and resume certificates, including the use of a grace period
- Additional support for the 4758 coprocessor (PCICC) that includes the ability to generate 2048-bit private keys
- New VSAM support that includes:
  - “Steps for creating additional alternate indexes” on page 80
  - “Tuning VSAM performance” on page 82
- New techniques in Chapter 13, “Advanced customization,” on page 145 that include:
  - “Scaling for high volume installations” on page 145
  - “Customizing distribution point CRLs” on page 150
  - “Adding an application domain” on page 158
- New messages in Chapter 21, “Messages,” on page 315:
  - IKYC036I
  - IKYC037I
  - IKYC038I
  - IKYC039I
  - IKYP030I
  - IKYS016I
  - IKYS017I
  - IKYS018I
- New code sample in “IKYMVSA” on page 377.
- System Management Facilities (SMF) record information for PKI Services is provided in a new chapter, Chapter 27, “SMF recording,” on page 385.
- Support for the following APARs:
  - OW56346
  - OW56663
  - OW56664

### Updated information

- Updated VSAM information:
  - “Planning VSAM storage requirements” on page 77
  - “Steps for creating the VSAM object store and ICL data sets and indexes” on page 79
  - “(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS” on page 81
- Updated diagnostic information for messages IKYD001I, IKYI002I, IKYP013I, IKYS004I, and IKYS005I in Chapter 21, “Messages,” on page 315.

### Moved information

- The information presented in the chapter previously called “Migration considerations”, which appeared in Part 1, “Planning,” on page 1, is removed from this publication. Beginning with z/OS Version 1 Release 5, this information is restructured and divided across the following z/OS system-level publications:
  - *z/OS Introduction and Release Guide*
  - *z/OS Migration*

– *z/OS Summary of Message and Interface Changes*

- Chapter 17, “PKI Services Trust Policy (PKITP),” on page 265 is now presented in Part 6, “Using the certificate validation service.” It was previously presented in Part 8, “Reference information.”

**Deleted information**

- A sample of `pkiserv.tmpl`, the certificate template file, is no longer presented in Part 8, “Reference information” and is deleted from this publication. You can obtain a copy from directory `/usr/lpp/pkiserv/samples` on your PKI Services system.

This document includes terminology, maintenance, and editorial changes.



---

## Part 1. Planning

The Planning part includes the following:

- Chapter 1, “Introducing PKI Services,” on page 3 provides an overview of PKI Services, its components, and related concepts.
- Chapter 2, “Planning your implementation,” on page 9 provides a planning overview for your implementation, including a discussion of the components that work with PKI Services. It also discusses the team members you will need to implement PKI Services and the skills they will need.
- Chapter 3, “Installing and configuring prerequisite products,” on page 17 describes installing and configuring related products: the z/OS HTTP Server, OCSF, LDAP, and optionally ICSF.





---

## Chapter 1. Introducing PKI Services

This chapter provides an overview of PKI Services.

It covers the following topics:

- “What is PKI Services?”
- “What is a certificate authority?”
- “What is PKI?” on page 4
- “Basic components of PKI Services and related products” on page 4
- “Component diagram” on page 5
- “Supported standards” on page 6
- “Supported certificate types” on page 7
- “Supported certificate fields and extensions” on page 8

---

### What is PKI Services?

z/OS Cryptographic Services PKI Services allows you use z/OS to establish a PKI infrastructure and serve as a certificate authority for your internal and external users, issuing and administering digital certificates in accordance with your own organization's policies. Your users can use a PKI Services application to request and obtain certificates through their own Web browsers, while your authorized PKI administrators approve, modify, or reject these requests through their own Web browsers. The Web applications provided with PKI Services are highly customizable, and a programming exit is also included for advanced customization. You can allow automatic approval for certificate requests from certain users and, to provide additional authentication, add host IDs, such as RACF user IDs, to certificates you issue for certain users. You can also issue your own certificates for browsers, servers, and other purposes, such as virtual private network (VPN) devices, smart cards, and secure e-mail.

PKI Services supports Public Key Infrastructure for X.509 version 3 (PKIX) and Common Data Security Architecture (CDSA) cryptographic standards. It also supports the following:

- The delivery of certificates through the Secure Sockets Layer (SSL) for use with applications that are accessed from a Web browser or Web server.
- The delivery of certificates that support the Internet Protocol Security standard (IPSEC) for use with secure VPN applications or IPSEC-enabled devices.
- The delivery of certificates that support Secure Multipurpose Internet Mail Extensions (S/MIME), for use with secure e-mail applications.

z/OS is certified as IdenTrust™ compliant. This allows z/OS installations to participate in the IdenTrust infrastructure by configuring PKI Services to operate as an IdenTrust compliant certificate authority (CA). For details, see Appendix C, “Configuring PKI Services as an IdenTrust™ certificate authority,” on page 391.

### What is a certificate authority?

The certificate authority, commonly called a CA, acts as a trusted third party to ensure that users who engage in e-business can trust each other. A certificate authority vouches for the identity of each party through the certificates it issues. In addition to proving the identity of the user, each certificate includes a public key that enables the user to verify and encrypt communications.

## Introducing PKI Services

The trustworthiness of the parties depends on the trust that is placed in the CA that issued the certificates. To ensure the integrity of a certificate, the CA digitally signs the certificate as part of creating it, using its signing private key. Trying to alter a certificate invalidates the signature and renders it unusable.

Protecting the CA's signing private key is critical to the integrity of the CA. For this reason, you should consider using ICSF to securely store your PKI Services CA's private key.

As a CA using PKI Services, you can do the following:

- Track certificates you issue with an issued certificate list (ICL) that contains a copy of each certificate, indexed by serial number
- Track revoked certificates using certificate revocation lists (CRLs). When a certificate is revoked, PKI Services updates the CRL during the next periodic update. Just as it signs certificates, the CA digitally signs all CRLs to vouch for their integrity.

## What is PKI?

The public key infrastructure (PKI) provides applications with a framework for performing the following types of security-related activities:

- Authenticate all parties that engage in electronic transactions
- Authorize access to sensitive systems and repositories
- Verify the author of each message through its digital signature
- Encrypt the content of all communications.

The PKIX standard evolved from PKI to support the interoperability of applications that engage in e-business. Its main advantage is that it enables organizations to conduct secure electronic transactions without regard for operating platform or application software package.

The PKIX implementation in PKI Services is based on the Common Data Security Architecture (CDSA) from Intel® Corporation. CDSA supports multiple trust models, certificate formats, cryptographic algorithms, and certificate repositories. Its main advantage is that it enables organizations to write PKI-compliant applications that support their business policies.

---

## Basic components of PKI Services and related products

*Table 1. Basic components of PKI Services and related products*

Administration	Assists authorized administrators to review requests for certificates, approve or reject requests, renew certificates, or revoke certificates
Web application	through their own Web browsers. The application consists of sample screens that you can easily customize to display your organization's logo. It also supports the following tasks: <ul style="list-style-type: none"><li>• Reviewing pending certificate requests</li><li>• Querying pending requests to process those that meet certain criteria</li><li>• Displaying detailed information about a certificate or request</li><li>• Monitoring certificate information, such as validity period</li><li>• Annotating the reason for an administrative action</li></ul>

Table 1. Basic components of PKI Services and related products (continued)

End-user Web application	Guides your users to request, obtain, and renew certificates through their Web browsers. The application consists of sample screens that you can easily customize to meet your organization's needs for certificate content and standards for appearance. It offers several certificate templates that you can use to create requests for a variety of certificate types, based on the certificate's intended purpose and validity period, and supports certificate requests that are automatically approved.
Exit	Provides advanced customization for additional authorization checking, validating, and changing parameters on calls to the R_PKIServ callable service (IRRSPX00), and capturing certificates for further processing. You can call this exit from the PKIServ CGIs and use its IRRSPX00 preprocessing and post-processing functions. A code sample in C language code is included.
ICSF (optional)	Securely stores the PKI Services certificate authority's private signing key.
LDAP	The directory that maintains information about the valid and revoked certificates that PKI Services issues in an LDAP-compliant format. You can use an LDAP server such as z/OS Integrated Security Services LDAP server.
PKI Services daemon	The server daemon that acts as your certificate authority, confirming the identities of users and servers, verifying that they are entitled to certificates with the requested attributes, and approving and rejecting requests to issue and renew certificates. It includes support for: <ul style="list-style-type: none"> <li>• An issued certificate list (ICL) to track issued certificates</li> <li>• Certificate revocation lists (CRLs) to track revoked certificates</li> </ul>
R_PKIServ callable service (IRRSPX00)	The application programming interface (API) that allows authorized applications, such as servers, to programmatically request the functions of PKI Services to generate, retrieve and administer certificates.
RACF (or equivalent)	Controls who can use the functions of the R_PKIServ callable service and protects the components of your PKI Services system. RACF creates your certificate authority's certificate, key ring and private key. You can also use it to store the private key, if ICSF is not available.
z/OS HTTP Server	PKI Services uses the Web server to encrypt messages, authenticate requests, and transfer certificates to intended recipients.

## Component diagram

Figure 1 on page 6 shows a typical PKI Services system.

## Introducing PKI Services

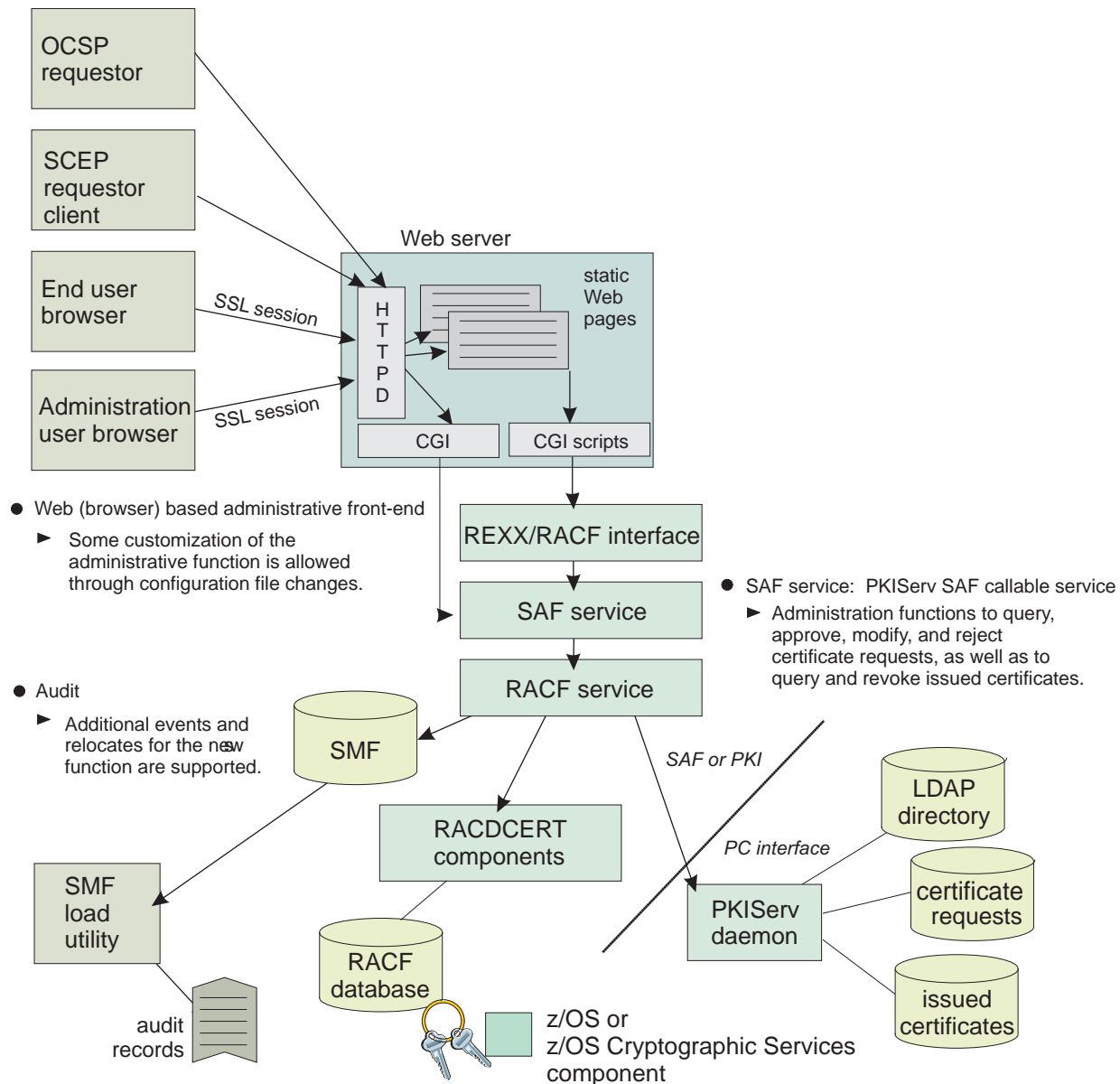


Figure 1. Component diagram of a typical PKI Services system

## Supported standards

PKI Services supports the following standards for public key cryptography:

- Secure Sockets Layer (SSL) version 2 and version 3, with client authentication
- PKCS #10 browser and server certificate format, with a base64-encoded response
- IPSEC certificate format
- S/MIME certificate format
- Browser certificates for:
  - Microsoft Internet Explorer version 5.x
  - Netscape Navigator and Netscape Communicator version 4.x
- Server certificates

- LDAP standard for communications with the directory
- X.509v3 certificates
- Certificate revocation lists (CRLv2)
- Key lengths up to 2048 bits for the RSA CA signing private keys and up to 1024 bits for DSA keys
- RSA algorithms for encryption and signing
- DSA algorithms for signing
- MD5 and SHA1 hash algorithms
- RFC 2560: *Online Certificate Status Protocol - OCSP*
- Cisco Systems' Simple Certificate Enrollment Protocol (SCEP) (Internet draft: draft-nourse-scep-11.txt)

The LDAP standard that PKI Services supports is LDAP Version 2. A directory using LDAP Version 3 (with RFC 1779 syntax) is acceptable if it is backwardly compatible with Version 2.

## Supported certificate types

Table 2 lists the types of certificates that you can request, based on the certificate templates that are included with PKI Services. Certificate templates are samples of the most commonly requested certificate types. You can add, modify, and remove certificate templates to customize the variety of certificate types you offer to your users.

Table 2. *Types of certificates you can request*

Type of certificate	Use
One-year PKI SSL browser certificate	End-user client authentication using SSL
One-year PKI S/MIME browser certificate	Browser-based e-mail encryption
Two-year PKI browser certificate for authenticating to z/OS	End-user client authorization using SSL when logging onto z/OS
Two-year PKI Authenticode—code signing server certificate	Software signing
Two-year PKI Windows logon certificate	End-user client authentication for an Active Directory user logging in to a Windows desktop using a smart card
Five-year PKI SSL server certificate	SSL Web server certification
Five-year PKI IPSEC server (firewall) certificate	Firewall server identification and key exchange
Five-year PKI intermediate CA server certificate	Subordinate (non-self-signed) certificate-authority certification
Five-year SCEP certificate	Creation of a preregistration record for certificate requestors. (Certificate requestors using Simple Certificate Enrollment Protocol (SCEP) must be preregistered.)
	Unlike other templates, this template is intended for administration use only.
<i>n</i> -year PKI browser certificate for extensions demonstration	Demonstration of all extensions supported by PKI Services

Table 2. Types of certificates you can request (continued)

Type of certificate	Use
One-year SAF browser certificate	End-user client authentication where the security product (RACF, not PKI Services) is the certificate provider
One-year SAF server certificate	Web server SSL certification where the security product (RACF, not PKI Services) is the certificate provider

**Note:** You can customize certificate templates to add, modify and remove certificate types.

---

## Supported certificate fields and extensions

PKI Services certificates support most of the fields and extensions defined in the X.509 version 3 (X.509v3) standard. This support lets you use these certificates for most cryptographic purposes, such as SSL, IPSEC, VPN, and S/MIME.

PKI Services certificates can include the following types of extensions:

### Standard extensions

The standard X.509v3 certificate extensions:

- authority information access
- authority key identifier
- basic constraints
- certificate policies
- certificate revocation list (CRL) distribution points
- extended key usage
- key usage
- subject alternate name
- subject key identifier

### Other extensions

Extensions that are unique to PKI Services, such as host identity mapping. This extension associates the subject of a certificate with a corresponding identity on a host system, such as with a RACF user ID.

To support your organization's policies, PKI Services also provides the means for you to customize and define certificate extensions. For example, you can change the extensions that are specified in the default certificate templates or create templates that return certificates with different extensions.

---

## Chapter 2. Planning your implementation

The implementation of PKI Services requires the interaction of several software products, each with its own required skills. Therefore, it is important to understand the tasks involved and to plan your implementation.

This chapter provides the information you need to understand the task of implementing PKI Services, determine which skills are required to complete your implementation team, and create your own implementation plan.

This chapter covers the following topics:

- “Installing PKI Services”
- “Determining prerequisite products” on page 10
- “Identifying skill requirements” on page 11
- “Creating an implementation plan” on page 14.

---

### Installing PKI Services

Your MVS programmer uses SMP/E to install PKI Services into a file system directory. By default, PKI Services is installed in the `/usr/lpp/pkiserv` directory but the MVS programmer can determine whether to change the default for this and other directories. Before your team begins installing and configuring prerequisite products and setting up PKI Services, you will need to know which file system directories were used so you can customize the install process.

Table 3 shows each file system variable with its description and default value. Your MVS programmer should review the rightmost column of this table, crossing out any defaults that have changed and recording the correct directory names.

*Table 3. File system directory variables*

Variable name	Description	Default value or customized value
<i>variables-dir</i>	The file system directory where PKI Services creates working files.	<b><code>/var/pkiserv</code></b>
<i>install-dir</i>	The file system directory where PKI Services is installed.	<b><code>/usr/lpp/pkiserv</code></b>
<i>runtime-dir</i>	The file system directory where PKI Services looks for configuration files.	<b><code>/etc/pkiserv</code></b>

---

### Requirements for sysplex support

If your installation plans to use sysplex support (running multiple independent instances of PKI Services, one per image, that work in unison):

- All systems in the sysplex that run PKI Services must be at z/OS V1R4 or later.
- All instances of PKI Services must share the same VSAM data sets. To do so, they use VSAM record-level sharing (RLS). This requires setting up a coupling facility for data sharing (lock and cache).

See “(Optional) preliminary steps for establishing VSAM RLS” on page 78 for information about creating VSAM data sets suitable for VSAM RLS. For information on establishing a Parallel Sysplex environment with a coupling facility, see *z/OS MVS Programming: Sysplex Services Guide*. For more information about

establishing data sharing for VSAM RLS, see *z/OS DFSMS Introduction* and *z/OS DFSMS Storage Administration Reference*.

---

### Determining prerequisite products

The installation and use of PKI Services requires the following products:

- z/OS HTTP Server
- LDAP directory server
- OCSF
- ICSF (optional)
- sendmail (optional)
- OCEP (optional).

The installation and use of RACF, or an equivalent security product, is required.

### z/OS HTTP Server

In a PKI Services system, the z/OS HTTP Server handles all requests that it receives from a Web browser. This includes requests for new certificates and requests to renew or revoke existing certificates. If needed, it performs authentication before allowing any exchange of information to take place.

z/OS HTTP Server must be installed on the same system where PKI Services is installed. SSL-enablement is required. If your HTTP server is SSL-enabled, your key file may be a RACF key ring, or a key file created by another product. For more information, see “Steps for installing and configuring the z/OS HTTP Server to work with PKI Services ” on page 17.

### LDAP directory server

Use of an LDAP server is required to maintain information about PKI Services certificates in a centralized location. The z/OS LDAP server is preferred but you can use a non-z/OS LDAP server if it can support the objectclasses and attributes that PKI Services uses. Typical PKI Services usage requires an LDAP directory server that supports the LDAP (Version 2) protocol (and the PKIX schema), such as the z/OS LDAP server. If you intend to use the z/OS LDAP server, you must configure it to use the TDBM backend.

Through the integration of the z/OS LDAP server with DB2®, the directory can support millions of directory entries. It also allows client applications, such as PKI Services, to perform database storage, update, and retrieval transactions. For more information, see “Steps for installing and configuring LDAP” on page 20.

### OCSF

PKI Services requires OCSF to be installed and configured so that the user ID under which the PKI Services daemon runs can use required services. For more information, see “Installing and configuring OCSF” on page 19.

### ICSF (optional)

ICSF is preferred but not required. You can begin using PKI Services without installing ICSF and install it later without reinstalling PKI Services. ICSF is strongly suggested to store and protect your certificate authority's private key. For more information, see “Installing and configuring ICSF (optional)” on page 22.



## sendmail (optional)

You need to configure sendmail if your installation plans to send e-mail notifications to users for certificate-related events, such as certificate expiration. For more information, see “Configuring sendmail (optional)” on page 23.

## OCEP (optional)

You need to install and configure OCEP if your installation plans to write an application to implement the use of PKI Trust Policy (PKITP). For more information, see “Configuring and getting started with PKITP” on page 269.

---

## Identifying skill requirements

The implementation of PKI Services requires the interaction of several software products, each with its own required skills. This means that your team may consist of people from several different disciplines, particularly if you work with a large organization.

This section provides the information you need to determine which skills are required to complete your implementation. These skills are presented in terms of job titles for people who specialize in those skills. For example, a task requiring MVS skills is referred to as a task for an MVS programmer. Therefore, if some of your team members have multiple skills, you may require fewer individuals to complete your team.

## Team members

Your team for installing and configuring prerequisite products and setting up PKI Services should include the following members:

- ICSF programmer (optional)
- LDAP programmer
- MVS programmer
- OCEP programmer (optional)
- OCSF programmer
- RACF administrator
- UNIX programmer
- Web server programmer

You may wish to include a Web page designer to customize your PKI Services Web applications. This task is listed in the chapter as a task for a Web server programmer.

One or more PKI administrators are needed to manage your ongoing operation as a certificate authority, once your PKI Services system is set up. The responsibilities of these administrators include approving, modifying and rejecting certificate requests and revoking certificates. It may be advisable to appoint a PKI administrator early, and involve this person in your planning.

**Important:** PKI administrators play a very powerful role in your organization. The decisions they make when managing certificates and certificate requests determine who will access your computer systems and what privileges they will have when doing so. Assign PKI administration duties to only highly trusted individuals.

## Planning your implementation

### Skills for setting up prerequisite products

The following table lists team members (alphabetically) and tasks and required skills needed for installing and configuring prerequisite products:

Table 4. Tasks and skills needed for installing prerequisite products

Role	Tasks	Required skills	Documented in:
ICSF programmer	(Optionally) installing and configuring ICSF (if not already done)	ICSF installation and configuration skills	<ul style="list-style-type: none"><li>• <i>z/OS Cryptographic Services ICSF Administrator's Guide</i></li><li>• <i>z/OS Cryptographic Services ICSF Application Programmer's Guide</i></li><li>• <i>z/OS Cryptographic Services ICSF System Programmer's Guide</i></li></ul>
LDAP programmer	Installing and configuring LDAP (if not already done) and recording information	LDAP installation and configuration skills	<ul style="list-style-type: none"><li>• <i>z/OS Integrated Security Services LDAP Server Administration and Use</i></li></ul>
OCEP programmer	(Optionally) Installing and configuring OCEP for use with PKITP	OCEP installation and configuration skills	<ul style="list-style-type: none"><li>• <i>z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming</i></li></ul>
OCSF programmer	Installing and configuring OCSF (if not already done) and recording information	OCSF installation and configuration skills	<ul style="list-style-type: none"><li>• <i>z/OS Open Cryptographic Services Facility Application Programming</i></li></ul>
UNIX programmer	(Optionally) Configuring sendmail if your installation is planning to send e-mail notifications to users about certificates	<ul style="list-style-type: none"><li>• Basic UNIX commands such as the <b>cp</b> (copy) command and <b>mkdir</b> (make directory) command</li><li>• sendmail configuration skills</li></ul>	<ul style="list-style-type: none"><li>• <i>z/OS Communications Server: IP Configuration Guide</i></li></ul>
Web server programmer	Installing and configuring the z/OS HTTP Server (if not already configured for at least non-SSL pages) and recording information	z/OS HTTP Server installation and configuration skills	<ul style="list-style-type: none"><li>• <i>z/OS HTTP Server Planning, Installing, and Using</i></li></ul>

Your team needs to install and configure prerequisite products before setting up PKI Services:

1. The Web server programmer installs and configures the z/OS HTTP Server.
2. The OCSF programmer installs and configures the OCSF.
3. The LDAP programmer installs and configures LDAP.
4. Optionally, the ICSF programmer install and configures ICSF.
5. Optionally, the OCEP programmer installs and configures the OCEP.

See Chapter 3, “Installing and configuring prerequisite products,” on page 17 for details about performing these tasks.

## Skills for setting up PKI Services

The following table lists team members (alphabetically) and the tasks and skills needed for setting up PKI Services:

Table 5. Roles, tasks, and skills for setting up PKI Services

Role	Tasks	Required skills	Documented in:
LDAP programmer	<ul style="list-style-type: none"> <li>Customizes LDAP configuration for PKI Services</li> </ul>	<ul style="list-style-type: none"> <li>LDAP customization skills</li> </ul>	<ul style="list-style-type: none"> <li><i>z/OS Integrated Security Services LDAP Server Administration and Use</i></li> </ul>
MVS programmer	<ul style="list-style-type: none"> <li>Creates VSAM object store and ICL data sets and indexes</li> <li>(Optionally) sets up VSAM RLS</li> <li>Starts the PKI Services daemon</li> </ul>	<ul style="list-style-type: none"> <li>Basic MVS skills               <ul style="list-style-type: none"> <li>Editing a data set</li> <li>ISPF COPY command</li> <li>MVS console START command</li> </ul> </li> <li>JCL knowledge to change job card</li> <li>Basic browser and Web skills</li> </ul>	<ul style="list-style-type: none"> <li><i>z/OS MVS System Commands</i></li> </ul>
RACF administrator	<ul style="list-style-type: none"> <li>Adds groups and user IDs</li> <li>Sets up access control</li> <li>Creates certificates</li> <li>Sets up daemon security</li> </ul>	<ul style="list-style-type: none"> <li>RACF administration</li> <li>REXX skills (for working with IKYSETUP REXX exec)</li> <li>RACF commands such as the following:               <ul style="list-style-type: none"> <li>ADDGROUP</li> <li>ADDSD</li> <li>ADDUSER</li> <li>RACDCERT</li> <li>RDEFINE</li> <li>PERMIT</li> <li>SETROPTS</li> </ul> </li> <li>TSO skills</li> </ul>	<ul style="list-style-type: none"> <li><i>z/OS TSO/E REXX Reference</i></li> <li><i>z/OS UNIX System Services Planning</i></li> <li><i>z/OS Security Server RACF Security Administrator's Guide</i></li> </ul>
UNIX programmer	<ul style="list-style-type: none"> <li>Copies files</li> <li>(Optionally) customizes environment variables</li> <li>(Optionally) customizes (non-<b>LDAP</b> sections of) <code>pkiserv.conf</code> configuration file</li> <li>Sets up <code>/var/pkiserv</code> directory</li> <li>Updates the <b>LDAP</b> section of the <code>pkiserv.conf</code> configuration file</li> </ul>	<ul style="list-style-type: none"> <li>Basic UNIX commands, such as the <b>cp</b> (copy) command</li> <li>Getting superuser authority</li> </ul>	<ul style="list-style-type: none"> <li><i>z/OS UNIX System Services Command Reference</i></li> <li><i>z/OS UNIX System Services Planning</i></li> </ul>

## Planning your implementation

Table 5. Roles, tasks, and skills for setting up PKI Services (continued)

Role	Tasks	Required skills	Documented in:
Web server programmer	<ul style="list-style-type: none"><li>Helps set up PKI Services<ul style="list-style-type: none"><li>Updates the z/OS HTTP Server configuration files</li><li>Starts the z/OS HTTP Server</li></ul></li><li>Customizes the PKI Services Web pages</li></ul>	<ul style="list-style-type: none"><li>z/OS HTTP Server customization skills</li><li>Editing configuration files</li><li>Customizing the Web pages</li></ul>	<ul style="list-style-type: none"><li><i>z/OS HTTP Server Planning, Installing, and Using</i></li></ul>

## Creating an implementation plan

Your implementation plan should include major subtasks, responsible parties, and a realistic estimate of time and effort required. The major tasks for implementing PKI Services are provided here as a basis for you to build your own plan.

## Task roadmap for implementing PKI Services

Table 6 shows the subtasks and associated procedures for implementing PKI Services. These tasks will comprise the major part of your implementation plan.

Table 6. Task roadmap for implementing PKI Services

Subtask	Associated procedure (See ...)
Installing and configuring prerequisite products: <ul style="list-style-type: none"><li>z/OS HTTP Server</li><li>OCSF</li><li>LDAP directory server</li><li>ICSF (optional)</li><li>OCEP (optional)</li><li>sendmail (optional)</li></ul>	Chapter 3, "Installing and configuring prerequisite products," on page 17 <ul style="list-style-type: none"><li>"Steps for installing and configuring the z/OS HTTP Server to work with PKI Services " on page 17</li><li>"Steps for installing and configuring OCSF to work with PKI Services" on page 19</li><li>"Steps for installing and configuring LDAP" on page 20</li><li>"Installing and configuring ICSF (optional)" on page 22</li><li>"Configuring and getting started with PKITP" on page 269</li><li>"Configuring sendmail (optional)" on page 23</li></ul>
Configuring your system for PKI Services: <ul style="list-style-type: none"><li>RACF</li><li>z/OS UNIX</li><li>LDAP configuration</li><li>z/OS HTTP Server</li></ul>	Part 2, "Configuring your system for PKI Services," on page 25 <ul style="list-style-type: none"><li>Chapter 4, "Running IKYSETUP to perform RACF administration," on page 27</li><li>Chapter 5, "Configuring the UNIX runtime environment," on page 45</li><li>Chapter 6, "Tailoring LDAP configuration for PKI Services," on page 65</li><li>Chapter 7, "Updating z/OS HTTP Server configuration and starting the server," on page 67</li></ul>

Table 6. Task roadmap for implementing PKI Services (continued)

Subtask	Associated procedure (See ...)
<ul style="list-style-type: none"> <li>• LDAP</li> <li>• VSAM</li> </ul>	<ul style="list-style-type: none"> <li>• Chapter 8, “Tailoring the PKI Services configuration file for LDAP,” on page 71</li> <li>• Chapter 9, “Creating VSAM data sets,” on page 77</li> <li>• Chapter 10, “Starting and stopping PKI Services,” on page 85</li> </ul>
Customizing PKI Services:	Part 3, “Customizing PKI Services,” on page 89
<ul style="list-style-type: none"> <li>• Customizing end-user Web pages</li> <li>• Customizing administration Web pages</li> <li>• Advanced customizing</li> </ul>	<ul style="list-style-type: none"> <li>• Chapter 11, “Customizing the end-user Web application,” on page 91</li> <li>• Chapter 12, “Customizing the administration Web pages,” on page 141</li> <li>• Chapter 13, “Advanced customization,” on page 145</li> </ul>
Testing PKI Services:	Part 4, “Using PKI Services,” on page 195
<ul style="list-style-type: none"> <li>• Using end-user Web pages</li> <li>• Using administration Web pages</li> </ul>	<ul style="list-style-type: none"> <li>• Chapter 14, “Using the end-user Web pages,” on page 197</li> <li>• Chapter 15, “Using the administration Web pages,” on page 217</li> </ul>
Administering PKI Services:	Part 5, “Administering security for PKI Services,” on page 239
<ul style="list-style-type: none"> <li>• RACF</li> </ul>	<ul style="list-style-type: none"> <li>• Chapter 16, “RACF administration for PKI Services,” on page 241</li> </ul>



---

## Chapter 3. Installing and configuring prerequisite products

After the MVS programmer installs PKI Services using SMP/E (but before team members set up PKI Services—see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 through Chapter 9, “Creating VSAM data sets,” on page 77), your team needs to set up prerequisite products:

- z/OS HTTP Server
- OCSF
- LDAP
- ICSF (optional)
- sendmail (optional)
- OCEP (optional)

You need to install and configure the z/OS HTTP Server, OCSF, and LDAP only if you are setting up prerequisite products for PKI Services for the first time. Installing ICSF is optional. You need to configure sendmail only if you are sending e-mail notifications to users (about rejected certificate requests or certificates that are ready for retrieval or about to expire). You need to configure OCEP only if you are developing your own application to use the PKI Trust Policy (PKITP).

---

### Tasks to perform before setting up PKI Services

Before you can set up PKI Services, your team needs to set up prerequisite software products by completing the following tasks, if not already done:

1. “Installing and configuring the z/OS HTTP Server”
2. “Installing and configuring OCSF” on page 19
3. “Installing and configuring LDAP” on page 20
4. “Installing and configuring ICSF (optional)” on page 22
5. “Configuring sendmail (optional)” on page 23
6. “Configuring and getting started with PKITP” on page 269 (optional)

This chapter explains these tasks in more detail.

---

### Installing and configuring the z/OS HTTP Server

You need perform this task only if you are setting up prerequisite products for PKI Services for the first time.

PKI Services requires that you have the z/OS HTTP Server installed and configured for at least non-SSL page retrieval. (Tasks of other team members, such as the RACF administrator and Web server programmer—see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 and Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 67 —assume that this is already done.)

### Steps for installing and configuring the z/OS HTTP Server to work with PKI Services

**Before you begin:**

1. You will need Web server programming skills to complete this procedure.
2. You may need to refer to the following document:

## Installing and configuring prerequisites

### *z/OS HTTP Server Planning, Installing, and Using*

Perform the following steps to install and configure the z/OS HTTP Server to work with PKI Services:

1. Use the following table to decide what you need to do:

If ...	Then ...	Notes®
The z/OS HTTP Server is not installed and configured ...	Install and configure z/OS HTTP Server by following the instructions in the installation section of <i>z/OS HTTP Server Planning, Installing, and Using</i> .	<b>Guideline:</b> For PKI Services, when you install the z/OS HTTP Server, do not use a password file.
The z/OS HTTP Server is installed but not configured for SSL ...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27.)	—
The z/OS HTTP Server is installed and configured for SSL using a RACF key ring ...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27.)	—
The z/OS HTTP Server is installed and configured for SSL using gskkyman ...	Fill in the missing values in the table in the next step. (The RACF programmer needs information for setting up PKI Services; see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27. The RACF programmer also needs to add your CA certificate to an existing keyfile; see Appendix B, “Using a gskkyman key database for your certificate store,” on page 389 for information about gskkyman steps.)	—

You can now perform the steps for the decision you have made.

2. Fill in the rightmost column of the following table with information from the configuration:



Table 7. z/OS HTTP Server information you need to record

z/OS HTTP Server information	Explanation	Value
z/OS HTTP Server fully qualified domain name	A fully qualified domain name is the name of a host system. It includes a series of subnames (each of which is a domain name). For example, ralvm7.vnet.ibm.com is a fully qualified domain name that includes the domain names ibm.com and vnet.ibm.com. (The RACF administrator needs to know the fully qualified domain name when setting up PKI Services.)	
The full UNIX pathname of your httpd.conf configuration file	(The Web server programmer needs to know the full UNIX pathname when updating the httpd.conf configuration file to support PKI Services.)	

## Installing and configuring OCSF

PKI Services requires OCSF to be installed and configured so that the user ID under which the PKI Services daemon runs can use required services. The OCSF programmer also needs to record some information.

### Steps for installing and configuring OCSF to work with PKI Services

You need perform this task only if you are setting up prerequisite products for PKI Services for the first time.

#### Before you begin:

1. Although the base feature of z/OS includes OCSF and ICSF, if you are in the United States or Canada, make sure you have ordered and installed the additional OCSF Security Level 3 feature. (There is no charge for this feature.)
2. You will need OCSF programming skills to complete this procedure.
3. You may need to refer to the configuration instructions in *z/OS Open Cryptographic Services Facility Application Programming* for the following information:
  - Instructions on how to set up the necessary security authorizations using RACF
  - Information on the RACF program control definitions necessary for OCSF
  - Instructions on how to run the installation scripts necessary to use OCSF.

Perform the following steps to install and configure OCSF to work with PKI Services:

1. If OCSF is not already installed and configured, follow the instructions for how to do so in *z/OS Open Cryptographic Services Facility Application Programming*.
2. If the value set for the registry directory differs from the default of '/var/ocsf', record the new value in the following table. (If it differs from the default, the UNIX programmer will need to update the OCSFREGDIR environment variable in the PKI Services environment variables file, pkiserv.envvars.)

## Installing and configuring prerequisites

Table 8. OCSF information you need to record

OCSF information	Explanation	Default value or customized value
Value set for the registry directory	This is the location of the OCSF registry. The default is '/var/ocsf'.	'/var/ocsf'

A later chapter, Chapter 17, “PKI Services Trust Policy (PKITP),” on page 265, provides information about the PKI Services OCSF Trust Policy, PKITP. For information about configuring this, see “Configuring and getting started with PKITP” on page 269.

---

## Installing and configuring LDAP

The LDAP programmer installs and configures LDAP for the TDBM DB2 backend and records entries that will be needed later.

### Steps for installing and configuring LDAP

You need perform this task only if you are setting up prerequisite products for PKI Services for the first time.

Although it may be configured otherwise, typical PKI Services usage requires access to an LDAP directory server. Install the LDAP directory server separately from PKI Services. After the installation is complete, LDAP needs to be configured for PKI Services. The directory stores issued certificates and certification revocation lists. The z/OS LDAP server is preferred but not required. The remainder of this chapter assumes you will use the z/OS LDAP server.

**Note:** The default name of the LDAP server configuration file is `slapd.conf` for the z/OS Integrated Security Services LDAP server.

You can use a non-z/OS LDAP server if it can support the object classes and attributes that PKI Services uses. For information about using a non-z/OS LDAP server, see Appendix A, “LDAP directory server requirements,” on page 387.

#### Before you begin:

1. You will need LDAP programming skills to complete this procedure.
2. You will need to refer to *z/OS Integrated Security Services LDAP Server Administration and Use*.

Perform the following steps to install and configure LDAP to work with PKI Services:

1. Use the following table to decide what you need to do:

If ...	Then...	Notes
You do not have LDAP installed and configured ...	Follow the instructions in the Administration section of <i>z/OS Integrated Security Services LDAP Server Administration and Use</i> .	<b>Note:</b> It is not necessary to set up the LDAP server for SSL because PKI Services does not use SSL to communicate with the LDAP server.

If ...	Then...	Notes
You have LDAP installed and configured but not for the TDBM backend ...	You need to migrate to the TDBM backend. See <i>z/OS Integrated Security Services LDAP Server Administration and Use</i> for details about how to do this.	—
You have LDAP installed and configured for the TDBM backend ...	Go to the next step.	—

You can now perform the steps for the decision you have made.

- Record the entries and values from the LDAP configuration step in the following table. (Your team will need this information when setting up PKI Services.)

Table 9. LDAP information you need to record

LDAP information	Explanation	Value
Administrator's distinguished name	<p>This is the distinguished name to use for LDAP binding. A distinguished name is the unique name of a data entry that identifies its position in the hierarchical structure of the directory. A distinguished name consists of the relative distinguished name (RDN) concatenated with the names of its ancestor entries. For example, an entry for Tim Jones could have an RDN of CN=Tim Jones and a DN of:</p> <p>CN=Tim Jones,O=IBM,C=US</p> <p>CAs typically have distinguished names in the following form:</p> <p>OU=<i>your-CA's-friendly-name</i>,O=<i>your-organization</i>,C=<i>your-country-abbreviation</i></p> <p>The LDAP administrator defines the administrator's distinguished name with the <b>adminDN</b> keyword in the LDAP server configuration file. For example, the value is "cn=Admin" in the following:</p> <p>adminDN "cn=Admin"</p>	
Administrator password	<p>This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example:</p> <ul style="list-style-type: none"> <li>By specifying the password as a TDBM entry by using the <b>userPassword</b> attribute in the <b>ldif2tdbm</b> load utility</li> <li>By using the <b>adminPW</b> keyword in the LDAP server configuration file configuration file (not suggested)</li> </ul>	
LDAP fully qualified domain name and port	<p>This is the domain name on which the LDAP server is listening. For example, for ldap.widgets.com:389, the fully qualified domain name is ldap.widgets.com and the port is 389. See Table 7 on page 19 for a definition of fully qualified domain name.</p>	

## Installing and configuring prerequisites

Table 9. LDAP information you need to record (continued)

LDAP information	Explanation	Value
Suffix	<p>A suffix in LDAP is the top-level name of the subtree. For example, for the following distinguished name:</p> <p><code>OU=your-CA's-friendly-name, O=your-organization, C=your-country-abbreviation</code></p> <p>the suffix could be either "<code>O=your-company, C=your-country-abbreviation</code>" or "<code>C=your-country-abbreviation</code>".</p> <p>The suffix value is specified after the suffix keyword in the LDAP server configuration file:</p> <p><code>suffix "O=your-company, C=your-country-abbreviation"</code></p> <p><b>Note:</b> If you have more than one suffix, record the suffix you intend to use as the root for storing the PKI Services CA certificate.</p>	

3. The chapters that follow require the LDAP server to be running. Follow the instructions in the chapter about running the LDAP server in *z/OS Integrated Security Services LDAP Server Administration and Use*.

## Installing and configuring ICSF (optional)

You can install and configure ICSF the first time you are setting up PKI Services or at a later time. Using ICSF is preferred but not required. RACF can use ICSF's public key data set (PKDS) to securely store the PKI Services CA signing key if directed to do so. For this to be successful, the ICSF programmer must install and configure ICSF for Public Key Algorithms (PKA), and ICSF must be running. (The RACF administrator uses the IKYSETUP REXX exec to set up any RACF profiles needed to control access to ICSF services and keys. For more information, see Chapter 4, "Running IKYSETUP to perform RACF administration," on page 27.)

**Note:** You do not have to choose whether or not to install ICSF and perform the installation and configuration at this point. You can do so later in the process.

### Before you begin:

- You will need ICSF programming skills to complete this procedure.
- You may need to refer to the following document:

*z/OS Cryptographic Services ICSF Administrator's Guide*

This document provides information about managing cryptographic keys, setting up and maintaining the PKDS, controlling who can use cryptographic keys and services, and general information about ICSF and cryptographic keys.

If ICSF is not already installed and configured for PKA, do this by following the instructions in *z/OS Cryptographic Services ICSF Administrator's Guide*.

---

## Configuring sendmail (optional)

If your installation plans to send e-mail notifications to users whose certificate request is rejected or whose certificate is ready for retrieval or about to expire, the UNIX programmer needs to configure sendmail.

**Before you begin:** You need the following document:

- *z/OS Communications Server: IP Configuration Guide*

Follow the instructions in *z/OS Communications Server: IP Configuration Guide* for configuring z/OS UNIX sendmail. In general, you need to perform the following steps:

1. Create an alias file to define the postmaster and MAILER-DAEMON user IDs and the nobody alias (/dev/null).
2. Create the sendmail configuration file using the m4 macro preprocessor.
3. Load this configuration file into sendmail.

**Note:** Because PKI Services always provides the return e-mail address, you do not need to configure sendmail to provide it. This simplifies your setup.

Perform the following steps to test your sendmail configuration:

1. From the UNIX command line, create a mail file with some information in it. The following example is called `mail.txt`. (You need this name in the next step.)

**Example:**

```
To:target-email@address.com
From:source-email@address.com
Subject:This is a test
```

2. Execute the following command:

```
sendmail -t <mail.txt
```



---

## Part 2. Configuring your system for PKI Services

After the MVS programmer installs PKI Services into the file system directory, your team needs to perform additional tasks to configure PKI Services, including the following:

- Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 describes how the RACF administrator updates and runs IKYSETUP, a REXX exec to perform RACF administration tasks, such as setting up the daemon user ID and giving accesses.
- Chapter 5, “Configuring the UNIX runtime environment,” on page 45 explains:
  - Copying files, such as the PKI Services configuration file
  - Updating environment variables
  - Updating the PKI Services configuration file
  - Setting up the /var/pkiserv file system directory.
- Chapter 6, “Tailoring LDAP configuration for PKI Services,” on page 65 explains how to update your LDAP configuration (performed earlier—see “Installing and configuring LDAP” on page 20) for PKI Services.
- Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 67 describes updating the z/OS HTTP Server configuration files and starting the z/OS HTTP Server.
- Chapter 8, “Tailoring the PKI Services configuration file for LDAP,” on page 71 explains how to update the **LDAP** section of the PKI Services configuration file.
- Chapter 9, “Creating VSAM data sets,” on page 77 explains how to create VSAM data sets.
- Chapter 10, “Starting and stopping PKI Services,” on page 85 explains how to start and stop the PKI Services daemon.





---

## Chapter 4. Running IKYSETUP to perform RACF administration

You need to perform this task only if you are configuring PKI Services for the first time.

PKI Services provides SYS1.SAMPLIB(IKYSETUP), a REXX exec, to perform RACF administration tasks for setting up PKI Services. The RACF administrator updates and runs this REXX exec, which issues RACF commands to perform the following tasks:

- Adding groups and user IDs
  - Setting up the PKI Services administration group
  - Creating the PKI Services daemon user ID
  - Giving appropriate access to the RACF group
  - Creating the surrogate user ID and giving the surrogate user ID authority to generate certificates

A surrogate user ID is the identity assigned to client processes when they are requesting certificate services. A surrogate user ID is required for external clients. **Guideline:** For simplicity, use surrogate user IDs for internal clients as well, rather than allowing them to access PKI Services under their own identities.
  - Associating the user ID with the PKI Services started procedure.
- Setting up access control to protect end-user and administrative functions of PKI Services:
  - Authorizing the PKI Services daemon user ID for CA functions
  - Giving administrators access to VSAM data sets
  - Optionally authorizing PKI Services for ICSF resources.
- Creating certificate authority (CA), registration authority (RA), and SSL certificates:
  - Creating a CA certificate and private key
  - Backing them up to a password-protected MVS data set
  - Optionally migrating the private key to ICSF
  - Optionally creating an RA certificate and private key for Simple Certificate Enrollment Protocol (SCEP)
  - Creating a SAF key ring and associating it with the certificate
  - Exporting the CA certificate to an MVS data set and file system file
  - Generating a server certificate signed by the new CA
  - Creating a key ring for the Web server
  - Associating the Web server and any trusted CA certificates to the key ring.
- Setting up the z/OS HTTP Server for surrogate operation.

---

### Overview of IKYSETUP

IKYSETUP consists of several parts:

- Configurable section—This section assigns values to variables.
- A section that issues RACF commands to perform RACF administration tasks. (See “Actions IKYSETUP performs by issuing RACF commands” on page 351 for details about the actions that various sections of code perform.)

## Running IKYSETUP

- A section that writes information (such as the name of the PKI Services administration group) to the log data set. The log itself consists of two parts: commands issued and other information. (See “Sample IKYSETUP log data set” on page 42.)

**Note:** By default, IKYSETUP creates the log. You can disable recording information to the log by changing the value of one of the variables in IKYSETUP (log\_dsn) to null.

The configurable section contains three parts:

- Values you must change (by making them specific to your company, such as your company’s name)
- Values you might change depending on how you want PKI Services set up (for example, whether your setup will include ICSF)
- Values you can optionally change (these defaults are acceptable without change, but you might want to change them to make them more specific to your company, for example the name of the PKI Services administration group, which by default is PKIGRP)

The following table illustrates the structure and divisions of IKYSETUP:

*Table 10. IKYSETUP—Structure and divisions*

Configurable section—assigns values to variables
<ul style="list-style-type: none"><li>• Values you must change to customize (See Table 11 on page 29.)</li><li>• Values you might change that are related to setup (See Table 16 on page 34.)</li><li>• Values you can optionally change (See Table 17 on page 36.)</li></ul>
Issues RACF commands
Records information in the log data set

---

## Before you begin

**Important:** Update and run IKYSETUP *only* if you have not done so previously for an earlier release (or if you are changing the value of one or more variables).

You need to collect the following documents:

- *z/OS Security Server RACF Command Language Reference*
- *z/OS Security Server RACF Security Administrator’s Guide*
- *z/OS TSO/E REXX Reference*

The RACF administrator needs to decide the values of variables in IKYSETUP and to record these values for future reference. Review and update as necessary the following three variables tables. There are three tables because there are three categories of variables:

- Variables whose values you are *required* to change, such as ones containing your company name
- Variables whose values you might want to change, depending based on how you are setting up PKI Services
- Variables whose values you can optionally change.

There is some overlap between the three types of variables, for example, if you are already using the RACF sample Web application, PKISERV.

**Guideline:** If you are running IKYSETUP for the first time, at a minimum, you need to complete the following:

- Table 11
- Table 15 on page 34
- The rows of Table 16 on page 34 concerning z/OS UNIX level security:
  - unix\_sec
  - bpx\_userid. and pgmctl\_dsn. (if z/OS level security is already set up)
- Review the default values in *all* the tables.

Several values described in this chapter, particularly for variables in Table 11 and Table 17 on page 36, can be qualified with a *ca\_domain* value based on whether you implement multiple CA domains. For information about implementing multiple CA domains, see “Adding a new CA domain” on page 161.

## Variables whose values must change

Fill in the blank lines in the rightmost column with your company's information (and cross out the defaults in these cells).

Table 11. IKYSETUP variables whose values must change

Variable name	Description	Referenced elsewhere	Default value and your company's information
ca_dn	<p>The CA's distinguished name. (For a definition of distinguished name, see Table 9 on page 21.)</p> <p>If you already have your CA certificate and private key set up in RACF, set ca_dn="", set ca_label (in the following row) to the value of your CA's label, and update ca_expires and web_expires (in Table 17 on page 36) to reflect the expiration date of your CA certificate.</p> <p>If you do not already have your CA certificate and private key set up in RACF, cross out the default in the rightmost cell of this row and record the information for your company-specific information for distinguished name on the blank line.</p>	<p>The suffix of the PKI Services CA's distinguished name must match the LDAP suffix. (The LDAP suffix is in the LDAP server configuration file. See Table 9 on page 21 for a definition of suffix.)</p> <p><b>Note:</b> However, do not specify a C('value') if it is not present in your LDAP suffix.</p>	<p>When you also set <i>ca_domain</i>: <b>OU('ca_domain Human Resources Certificate Authority')</b></p> <p>When you do not set <i>ca_domain</i>: <b>OU('Human Resources Certificate Authority')</b></p> <p><b>O('Your Company')</b></p> <p><b>C('Your Country 2 Letter Abbreviation')</b></p> <hr/>

## Running IKYSETUP

Table 11. IKYSETUP variables whose values must change (continued)

Variable name	Description	Referenced elsewhere	Default value and your company's information
ca_label	The CA certificate label. If you already have your CA certificate and private key set up in RACF (and your CA certificate's label differs from the default), you need to set ca_label to your CA certificate's label.	No	When you also set ca_domain: <b>ca_domain Local PKI CA</b>  When you do not set ca_domain: <b>Local PKI CA</b>  (Replace the default if you already have your CA certificate and private key set up in RACF.)
daemon_uid	The z/OS UNIX user identifier (UID) associated with the PKI Services daemon user ID.	No	<b>554</b>
pki_gid	The z/OS UNIX group identifier (GID) for the PKI Services administration group.	No	<b>655</b>
pkigroup_mem.	Members of the PKI administration group are responsible for administering PKI Services functions.  <b>Guideline:</b> Assign PKI administration duties to only highly trusted individuals.  pkigroup_mem. is a list in which pkigroup_mem.0 is the number of members in the list and the rest of the entries are their user IDs. You must change the pkigroup_mem.0 to at least 1, and change pkigroup_mem.1 through pkigroup_mem.n to the member user IDs.	No	<b>0</b> (default for pkigroup_mem.0, the number of member user IDs)  <b>Note:</b> You must change the default to at least 1.  (Record the member IDs:)
ra_dn	The RA's distinguished name for use with Simple Certificate Enrollment Protocol (SCEP). (For a definition of distinguished name, see Table 9 on page 21.)  This name should be similar but not identical to your CA's distinguished name. If you do not wish to have PKI Services operate with a separate RA certificate, set ra_dn="".	No	<b>CN('Registration Authority')</b>  <b>OU('Human Resources Certificate Authority')</b>  <b>O('Your Company')</b>  <b>C('Your Country 2 Letter Abbreviation')</b>

Table 11. IKYSETUP variables whose values must change (continued)

Variable name	Description	Referenced elsewhere	Default value and your company's information
ra_label	The certificate label of your RA certificate in RACF.	No	<b>Local PKI RA</b>
surrog_uid	The UID associated with the surrogate user ID.	No	<b>555</b>
web_dn	<p>Your Web server's distinguished name. (For a definition of distinguished name, see Table 9 on page 21.)</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The RACF administrator copies the fully qualified domain name from an earlier table: Table 7 on page 19.</li> <li>2. If you already have your Web server configured for SSL: <ul style="list-style-type: none"> <li>• Set web_dn=""</li> <li>• Update the web_ring row</li> </ul> </li> </ol> <p>(You need to connect your PKI Services CA certificate to your key ring. See the web_ring row for directions.)</p>	The value of the Web server's common name (CN), which is your server's symbol IP address. For example, <i>www.YourCompany.com</i> must match your Web server's fully qualified domain name.	<b>CN('www.YourCompany.com')</b> <b>O('Your Company')</b> <b>L('Your City')</b> <b>SP('Your Full State or Province Name')</b> <b>C('Your Country 2 Letter Abbreviation')</b>
web_ring	<p>The name of the Web server's SAF key ring.</p> <p>If your Web server is configured for SSL and you are using a RACF key ring, set web_ring to the value of the RACF key ring. If your Web server is configured for SSL and you are using gskkyman, set web_ring="" and see Appendix B, "Using a gskkyman key database for your certificate store," on page 389 for additional directions.</p>	httpd*.conf— KeyFile directive	<b>SSLring</b>

## Variables whose values may change depending on setup

To help in completing the next table of variables (see Table 16 on page 34) fill out the following four decision tables:

### Deciding the value of key\_backup

Use the following decision table to determine the value of key\_backup in Table 16 on page 34. The key\_backup variable determines whether the PKI Services CA certificate and private key should be backed up to an encrypted data set.

Table 12. Decision table for key\_backup

If ...	Then ...	Notes
You want to back up your CA's certificate and private key to a passphrase encrypted data set ...	Do not change the default key_backup=1	When you use IKYSETUP, you need to enter a passphrase whose display is not inhibited—it appears on the screen in the clear.  You cannot backup PCICC keys (key_type=2).
You do <i>not</i> want to back up your CA's certificate and private key to a passphrase encrypted data set ...	Set key_backup=0	—

### Deciding the value of key\_type

Use the following decision table to determine the value of key\_type in Table 16 on page 34. The key\_type variable determines whether you are using ICSF, PCICC, or DSA for private key protection.

By default, IKYSETUP does not use ICSF. **Guideline:** Do not change the default the first time you run IKYSETUP but change it before going into a production environment. (For information about installing and configuring ICSF, see “Installing and configuring ICSF (optional)” on page 22.)

Table 13. Decision table for key\_type

If ...	Then ...	Notes
You want to use software cryptography and you want a key generated using the RSA algorithm ...	Do not change the default key_type=0	—
You want to use ICSF for private key protection but do not want the key generated by the PCI cryptographic coprocessor (PCICC) ...	Set key_type=1	Review and possibly change the following additional variables in Table 16 on page 34: <ul style="list-style-type: none"> <li>csfkeys_profile</li> <li>csfserv_profile</li> <li>csfusers_grp</li> </ul>
You want to use ICSF for private key protection and you want the key generated by PCICC ...	Set key_type=2	PKI Services does not automatically backup the private key when you select the 2 value.  Review and possibly change the following additional variables in Table 16 on page 34: <ul style="list-style-type: none"> <li>csfkeys_profile</li> <li>csfserv_profile</li> <li>csfusers_grp</li> </ul>
You want to use software cryptography and you want a key generated using the DSA algorithm ...	Set key_type=3	The key cannot be saved in ICSF.

### Deciding the value of restrict\_surrog

Use the following decision table to determine the value of restrict\_surrog in Table 16 on page 34. The restrict\_surrog variable determines if the RESTRICTED attribute is assigned to the surrogate user ID. The RESTRICTED attribute limits the resources available to this user ID.

By default, IKYSETUP does not assign the RESTRICTED attribute to the surrogate user ID. **Guideline:** Do not change the default the first time you run IKYSETUP but change it before going into a production environment. For more information, see the chapter about defining groups and users in *z/OS Security Server RACF Security Administrator's Guide*.

Table 14. Decision table for restrict\_surrog

If ...	Then ...
You want to assign the RESTRICTED attribute to the surrogate user ID ...	Set restrict_surrog=1
You do <i>not</i> want to assign the RESTRICTED attribute to the surrogate user ID ...	Do not change the default restrict_surrog=0

### Deciding the value of unix\_sec

Use the following decision table to determine the value of unix\_sec in Table 16 on page 34. The unix\_sec variable determines whether you want to use z/OS UNIX security, which is a higher level of security. z/OS UNIX provides two levels of security:

#### UNIX level security

This is a less stringent level of security than z/OS UNIX level security. It is for installations where system programmers have been granted superuser authority. Programs that run with superuser authority have daemon level authority and can issue MVS identity-changing services without entering a \_passwd() for the target user ID. With this level of security, the BPX.DAEMON profile in the FACILITY class is not defined.

#### z/OS UNIX level security

This is a higher level of security than z/OS UNIX level security. It lets your system exercise more control over superusers. With this level of security, the BPX.DAEMON profile in the FACILITY class is defined.

## Running IKYSETUP

Table 15. Decision table for `unix_sec`

If ...	Then ...	Notes
You already have z/OS UNIX security set up ...	Set <code>unix_sec=1</code>	—
You do not have z/OS UNIX security set up and you do not want to set it up ...	Do not change the default of <code>unix_sec=0</code>	—
You do not have z/OS UNIX security set up and you want to set it up for the first time ...	Set <code>unix_sec=2</code>	<ol style="list-style-type: none"> <li>For information about additional manual configuration, see the section about establishing z/OS UNIX security in the <i>z/OS UNIX System Services Planning</i>.</li> <li>If you are setting <code>unix_sec=2</code>, you must update the following variables: <ul style="list-style-type: none"> <li><code>bpx_userid</code>.</li> <li><code>pgmctl1_dsn</code>.</li> </ul> </li> </ol>

Update the following table based on your answers in the preceding decision tables. If you have decided to change any of the defaults in the rightmost column, cross out the defaults and enter your company's information:

Table 16. *IKYSETUP* variables you might want to change depending on setup

Variable name	Description	Referenced elsewhere	Default value or your company's information
<code>bpx_userid</code> .	A list of user IDs with daemon and server authority. The <code>bpx_userid.0</code> is the number of items in the list and the rest of the entries are the z/OS UNIX user IDs. (This is non-applicable if <code>unix_sec</code> $\neq$ 2.)	No	1 (default for number of items)  <b>OMVSKERN</b>
<code>ca_keysize</code>	The size in bits of the certificate-authority's private key.  The minimum key size is 512 bits. The maximum key size is 1024 bits for <code>key_type=0</code> and <code>key_type=1</code> , and 2048 bits for <code>key_type=2</code> and <code>key_type=3</code> .	No	<b>1024</b>
<code>csfkeys_profile</code>	A profile to protect the PKI Services key in ICSF. (This is non-applicable if <code>key_type=0</code> or <code>key_type=3</code> .) If you do not want IKYSETUP to create the profile, set <code>csfkeys_profile=""</code> . <b>Note:</b> When RACF stores the private key in the PKDS, it generates the label as: <code>'IRR.DIGTCERT.CERTIFAUTH. unique-time-stamp'</code>	No	<b>IRR.DIGTCERT.CERTIFAUTH.*</b>
<code>csfserv_profile</code>	A profile to protect ICSF services. (This is non-applicable if <code>key_type=0</code> or <code>key_type=3</code> .)	No	<b>CSF*</b>



Table 16. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
csfusers_grp	A group of authorized ICSF service users. (This is non-applicable if key_type=0 or key_type=3.)	No	
key_backup	Specifies whether the PKI Services CA certificate and private key should be backed up to an encrypted data set. The value can be: <ul style="list-style-type: none"> <li>1 (yes— the default)</li> <li>0 (no).</li> </ul> <b>Note:</b> This value is ignored when key_type=2 is also specified.  When you use IKYSETUP with key_backup=1, you need to enter a passphrase whose display is not inhibited—it appears on the screen in the clear.	No	1 (yes)
key_type	Specifies whether PKI Services should use ICSF, PCICC, or DSA for private key operations. The value can be: <ul style="list-style-type: none"> <li>0 (use software cryptography with the RSA algorithm—the default)</li> <li>1 (use ICSF but not PCICC)</li> <li>2 (use ICSF and PCICC)</li> <li>3 (use software cryptography with the DSA algorithm).</li> </ul> If you are changing key_type to 1 or 2, see also the csfkeys_profile, csfserv_profile, and csfusers_grp rows.  <b>Guideline:</b> Do not change the default the first time you run IKYSETUP, but change it before going into a production environment.	If choosing option 1 or 2, ICSF must be configured for RSA (PKA) operations and running.	0
pgmctl_dsn.	A list in which pgmctl_dsn.0 is the number of items in the list and the rest of the entries are a list of load libraries to be program controlled.  <b>Rule:</b> If you set unix_sec=2, you must update the list of data sets.	No	8 (default for number of items) <ul style="list-style-type: none"> <li>'CEE.SCEERUN'</li> <li>'CBC.SCLBDLL'</li> <li>'SYS1.SIEALNKE'</li> <li>'SYS1.CSSLIB'</li> <li>'TCPIP.SEZALOAD'</li> <li>'SYS1.LINKLIB'</li> <li>'CSF.SCSFMOD0'</li> <li>'CSF.SCSFMOD1'</li> </ul>

## Running IKYSETUP

Table 16. IKYSETUP variables you might want to change depending on setup (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
restrict_surrog	Specifies whether the surrogate user ID should be marked restricted. The value can be: <ul style="list-style-type: none"><li>• 0 (no—the default)</li><li>• 1 (yes)</li></ul> <b>Guideline:</b> Do not change the default the first time you run IKYSETUP, but change it before going into a production environment.	No	0 (no)
unix_sec	Specifies whether to set up z/OS UNIX level security. (See page 33 for a definition of z/OS UNIX level security.) The value can be: <ul style="list-style-type: none"><li>• 0 (do not set up — the default)</li><li>• 1 (is already set up)</li><li>• 2 (add this level of security)</li></ul> <b>Rule:</b> If you are changing unix_sec to 1 or 2, you must update the bpx_userid. and pgmctl_dsn. rows. <b>Guideline:</b> Do not set unix_sec=2 the first time you are running IKYSETUP.	For unix_sec=2, the names of the load libraries need to change.	0 (no)

## Variables you can optionally change

Review the values of the following variables to determine if you want to change any of the defaults in the rightmost column. (You should probably change at least the values for ca\_expires and web\_expires.) If you decide to change any value, cross out the default in the rightmost column and record your company's information.

Table 17. IKYSETUP variables you can optionally change

Variable name	Description	Referenced elsewhere	Default value or your company's information
backup_dsn	The data set that will contain a backup copy of the PKI Services certificate and private key.	No	When you also set ca_domain: 'daemon.ca_domain.PRIVATE.KEY.BACKUP.P12BIN'  When you do not set ca_domain: 'daemon.PRIVATE.KEY.BACKUP.P12BIN'  <b>Note:</b> The daemon refers to the daemon variable in this table.

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
ca_domain	<p>The unique name for the CA when you establish multiple PKI Services CAs.</p> <p>If specified, the first eight characters must uniquely identify the CA. The characters of the CA_domain value are limited to the following character set: alphanumeric characters (a–z, A–Z, 0–9) and the hyphen (-). In addition, the first character must not be a number or hyphen.</p>	No	<p>""</p> <p><b>Guideline:</b> Do not change the default (null) value until you perform advanced customization. (See "Adding a new CA domain" on page 161.)</p>
ca_expires	The date the PKI Services CA certificate expires.	No	<p><b>2020/01/01</b></p> <p>The date format is <i>yyyy/mm/dd</i>.</p> <p>You should update this value to the expiration date of your CA certificate.</p>
ca_ring	The name of the PKI Services SAF key ring.	pkiserv.conf— <b>SAF</b> KeyRing value	<p>When you also set <i>ca_domain</i>: <b>CAring.ca_domain</b></p> <p>When you do not set <i>ca_domain</i>: <b>CAring</b></p>
daemon	<p>The PKI Services daemon user ID.</p> <p>If you also set <i>ca_domain</i>, you may choose to assign a unique user ID to the daemon for each CA domain. <b>Example:</b> For a <i>ca_domain</i> called BankA, you might choose user ID PKISRVDA.</p>	pkiserv.conf— <b>SAF</b> KeyRing value	<b>PKISRVDA</b>
export_dsn	The data set that will contain the PKI Services certificate for copying to file system.	No	<p>When you also set <i>ca_domain</i>: <b>'daemon.ca_domain.CACERT.DERBIN'</b></p> <p>When you do not set <i>ca_domain</i>: <b>'daemon.CACERT.DERBIN'</b></p> <p><b>Note:</b> The <i>daemon</i> refers to the daemon variable in this table.</p>

## Running IKYSETUP

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
log_dsn	The log data set name.	No	<p>When you also set <i>ca_domain</i>: '<i>your-id.ca_domain</i>.IKYSETUP.LOG'</p> <p>When you do not set <i>ca_domain</i>: '<i>your-id</i>.IKYSETUP.LOG'</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The <i>your-id</i> refers to the RACF ID of the person running IKYSETUP. (You do not need to add this; MVS adds this for you.)</li> <li>2. Changing the default is not suggested.</li> </ol>
pkigroup	<p>The PKI Services administration group. This is a RACF group containing the list of user IDs that are authorized to use PKI Services administration functions.</p> <p>If you also set <i>ca_domain</i>, you may choose to assign a unique group name to the administration group for each CA domain. <b>Example:</b> For a <i>ca_domain</i> called BankA, you might choose group name PKIGRPA.</p>	No	<b>PKIGRP</b>
ra_backup_dsn	<p>The data set that will contain a backup copy of the PKI Services RA certificate and private key.</p> <p>This name should be similar but not identical to the backup_dsn value.</p>	No	<p>When you also set <i>ca_domain</i>: '<i>daemon.ca_domain</i>.PRIVATE..RAKEY.BACKUP.P12BIN'</p> <p>When you do not set <i>ca_domain</i>: '<i>daemon</i>.PRIVATE.RAKEY.BACKUP.P12BIN'</p> <p><b>Note:</b> The <i>daemon</i> refers to the daemon variable in this table.</p>
signing_ca_label	The label of the CA certificate that is the superior (signer) of the PKI Services CA. If specified, the value must match the label of an existing CERTAUTH certificate in RACF that has a private key. Use this value to create a CA hierarchy when you establish multiple PKI Services CAs.	No	""

Table 17. IKYSETUP variables you can optionally change (continued)

Variable name	Description	Referenced elsewhere	Default value or your company's information
surrog	<p>The surrogate user ID for PKI Services.</p> <p>If you also set <i>ca_domain</i>, you may choose to assign a unique user ID as the surrogate user ID for each CA domain. <b>Example:</b> For a <i>ca_domain</i> called BankA, you might choose user ID PKISERVA.</p> <p><b>Note:</b> This cannot be an existing user ID (because IKYSETUP creates the user ID with the NOPASSWORD attribute).</p>	Surrogate user ID in httpd*.conf	<b>PKISERV</b>
vsamhlq	<p>The high-level qualifier of the VSAM data sets for PKI Services.</p> <p><b>Note:</b> The RACF administrator gets this information from the MVS programmer.</p>	<ul style="list-style-type: none"> <li>• <b>ObjectStore</b> *DSN values in pkiserv.conf</li> <li>• Data sets names in IKYCVSAM</li> </ul>	Same as the daemon variable earlier in this table.
web_expires	The date the Web server certificate expires.	No	<p><b>2020/01/01</b></p> <p>The date format is <i>yyyy/mm/dd</i>.</p> <p>You should update this value to the expiration date of your CA certificate.</p>
web_label	The label for the Web server's certificate.	No	<b>SSL Cert</b>
webserver	The Web server's daemon user ID.	See Web server documentation.	<b>WEBSRV</b>

## Steps for performing RACF tasks using IKYSETUP

Use the following directions to run IKYSETUP only if you have not done so for a previous release (or if you are changing values).

You can use the following directions to run IKYSETUP with minimal changes or to extensively customize it.

**Guideline:** If this is your first attempt to use IKYSETUP, change only the IKYSETUP variables in the section Things you must change. You can refine IKYSETUP later, after you are familiar with the process of updating and running it.

The following flowchart illustrates the iterative nature of the process of updating IKYSETUP:

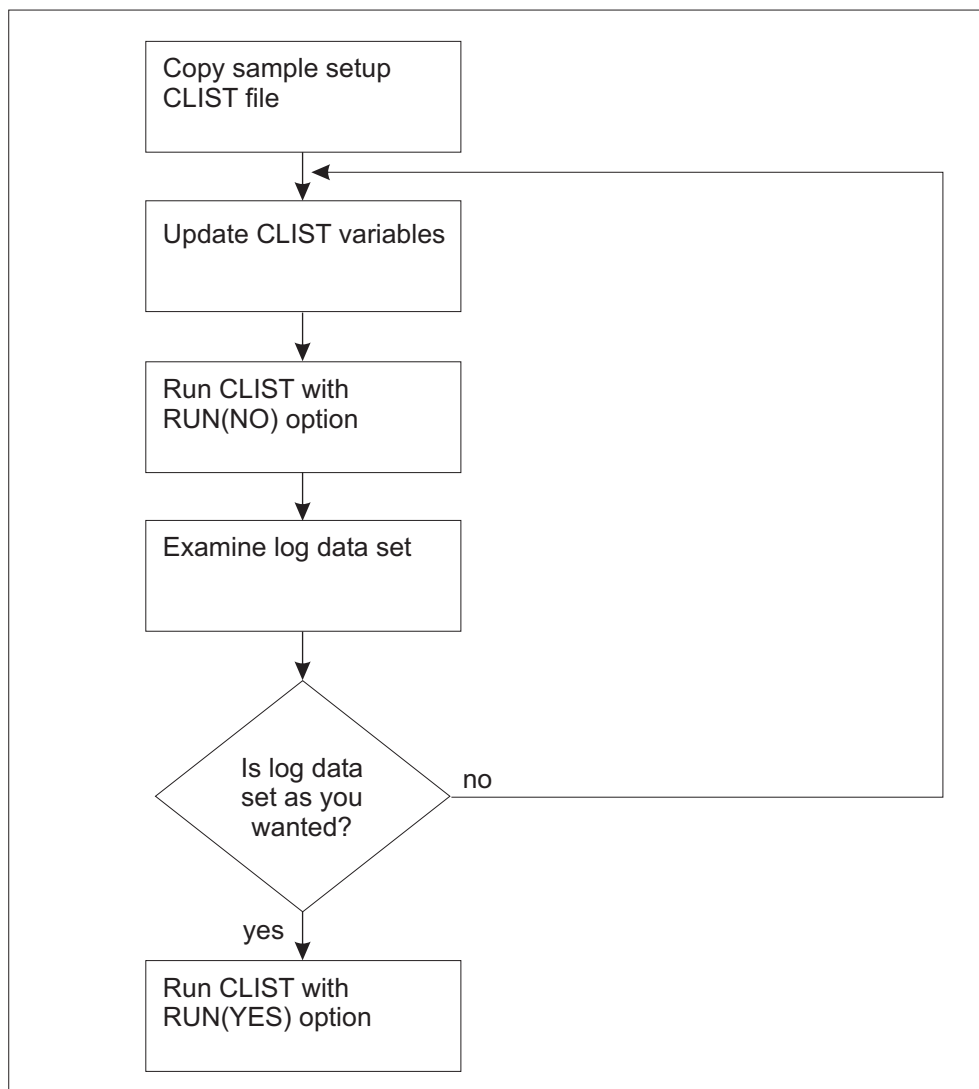


Figure 2. Flowchart of the process of updating IKYSETUP

Perform the following steps to use IKYSETUP to perform RACF administration tasks:

1. Copy SYS1.SAMPLIB(IKYSETUP) to a data set you are permitted to edit.

2. Edit the IKYSETUP code to update the values of variables you changed in Table 11 on page 29.

The following example shows how to change the `pkigroup_mem.` variables. (Remember that for `pkigroup_mem.`, you set `pkigroup_mem.0` to the number of items in the list and `pkigroup_mem.1` through `pkigroup_mem.n` to the PKI Services administration group member IDs.)

**Example:**

```
pkigroup_mem.0=3      /* Number of pkigroup members to connect */
pkigroup_mem.1="TOM"
pkigroup_mem.2="DICK"
pkigroup_mem.3="HARRY"
```

3. If necessary, update the values of variables you changed in Table 16 on page 34.

The following example shows how to change the `key_type` variable.

**Example:**

```
key_type=1
```

---

4. Optionally update any variables you changed in Table 17 on page 36.

The following example shows how to change the `log_dsn` variable.

**Example:**

```
log_dsn="PRIVATE.IKYSETUP.LOG"
```

---

5. Run IKYSETUP by entering the following command:

```
EX 'data-set-name(IKYSETUP)' 'RUN(NO)'
```

**Notes:**

- a. The user ID that runs IKYSETUP must be a RACF SPECIAL user ID.
- b. When IKYSETUP runs, it prompts you to enter your secret passphrase. (This is for encrypting the backup copy of your CA certificate and private key.) Be aware that asterisks do not replace the secret passphrase; it appears on the screen in the clear.

**Important:** Make a note of this passphrase. If you forget it, your backup will be useless.

- c. The NO option in the command specifies displaying the commands only. (This creates a log data set listing the commands and other information. Alternative parameters are: YES, which indicates running IKYSETUP as is, and PROMPT, indicates prompting the user before running each command.)
- 

6. Review the log data set. (See “Sample IKYSETUP log data set” on page 42 for an example of the data that appears on your display when you are running IKYSETUP; this is similar to the contents of the log data set.) The top part identifies the tasks and shows the commands that run to perform those tasks. Review this to ensure that the issued commands match your expectations. (For more information about these commands, see “Actions IKYSETUP performs by issuing RACF commands” on page 351.) The bottom part provides a record of important information that you will need for later steps, such the name of your daemon user ID. Review this information to ensure that the values are the ones you want.

If you want to change any of the commands or information in the log data set, you need to change additional values in IKYSETUP. Remember to record any additional changes in Table 11 on page 29, Table 16 on page 34, and Table 17 on page 36. Then go back to Step 3.

---

7. If the log data set includes the commands and information you want, rerun the IKYSETUP code by entering the following command:

```
EX 'data-set-name(IKYSETUP)' 'RUN(YES)'
```

---

8. After running IKYSETUP with RUN(YES), examine the results recorded in the log data set. Investigate and rerun (potentially by hand) any failing

## Running IKYSETUP

commands. Investigate informational messages and make any necessary corrections. (Informational messages usually indicate a setup problem that may affect operations later. For example, any informational message from the RACDCERT commands that indicate that the certificate has been marked NO TRUST is an error.)

- 
9. For the PKI Services procedure to start, the PKI Services user ID (by default, PKISRV) needs read access to the OCSF services. Provide this access by entering the following RACF commands:

```
PERMIT CDS.CSSM CLASS(FACILITY) ID(PKISRV) ACC(READ)
PERMIT CDS.CSSM.CRYPTO CLASS(FACILITY) ID(PKISRV) ACC(READ)
PERMIT CDS.CSSM.DATALIB CLASS(FACILITY) ID(PKISRV) ACC(READ)
SETROPTS RACLIST(FACILITY)
```

- 
10. If you intend to use encrypted LDAP passwords, you need to perform additional RACF administration tasks; see “Using encrypted passwords for LDAP servers” on page 260.
- 

## Sample IKYSETUP log data set

Here is an example of the data that appears when you run IKYSETUP.

```
Creating users and groups ...
ADDUSER PKISRV name('PKI Srvs Daemon') nopassword omvs(uid(554) assize(256000000) threads(512))
ADDUSER PKISRV nopassword omvs(uid(555)) name('PKI Srvs Surrogate')
SETROPTS EGN GENERIC(DATASET)
ADDSD 'PKISRV.*' UACC(NONE)
ADDGROUP PKIGRP OMVS(GID(655))
Allowing administrators to access PKI databases ...
PERMIT 'PKISRV.*' ID(PKIGRP) ACCESS(CONTROL)
SETROPTS GENERIC(DATASET) REFRESH
Creating the CA certificate ...
RACDCERT GENCERT CERTAUTH SUBJECTSDN(OU('Human Resources Certificate Authority')
O('Your Company') C('Your Country 2 Letter Abbreviation'))
WITHLABEL('Local PKI CA') NOTAFTER(DATE(2020/01/01))
Backing up the CA certificate ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA')) DSN('PKISRV.PRIVATE.KEY.BACKUP.P12BIN')
FORMAT(PKCS12DER) PASSWORD('*****')
Marking CA certificate as HIGHTRUST ...
RACDCERT CERTAUTH ALTER(LABEL('Local PKI CA')) HIGHTRUST
Saving the CA certificate to a data set for OPUT ...
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA')) DSN('PKISRV.PRIVATE.CACERT.DERBIN') FORMAT(CERTDER)
Creating the PKI Services keyring ...
RACDCERT ADDRING(Caring) ID(PKISRV)
RACDCERT ID(PKISRV) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(Caring) USAGE
(PERSONAL) DEFAULT)
Creating the Webserver SSL certificate and keyring ...
RACDCERT GENCERT ID(WEBRV) SIGNWITH(CERTAUTH LABEL('Local PKI CA')) WITHLABEL
('SSL Cert') SUBJECTSDN(CN('www.YourCompany.com') O('Your Company') L('Your City')
SP('Your Full State or Province Name') C('Your Country 2 Letter Abbreviation'))
NOTAFTER(DATE(2020/01/01))
RACDCERT ADDRING(SSLring) ID(WEBRV)
RACDCERT ID(WEBRV) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(SSLring))
RACDCERT ID(WEBRV) CONNECT(ID(WEBRV) LABEL('SSL Cert') RING(SSLring)
USAGE(PERSONAL) DEFAULT)
Giving PKISRV access to BPX.SERVER ...
RDEFINE FACILITY BPX.SERVER
PERMIT BPX.SERVER CLASS(FACILITY) ID(PKISRV) ACCESS(READ)
Allowing the PKI Services daemon to act as a CA ...
RDEFINE FACILITY IRR.DIGTCERT.GENCERT
RDEFINE FACILITY IRR.DIGTCERT.LISTRING
RDEFINE FACILITY IRR.DIGTCERT.LIST
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(PKISRV) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(PKISRV) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(PKISRV) ACCESS(READ)
Allowing the Webserver to access its keyring ...
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(WEBRV) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(WEBRV) ACCESS(READ)
```



```

Allowing the Webserver to switch identity to PKISERV ...
SETOPTS CLASSACT(SURROGAT)
RDEFINE SURROGAT BPX.SRV.PKISERV
PERMIT BPX.SRV.PKISERV CLASS(SURROGAT) ID(WEBSRV) ACCESS(READ)
SETOPTS RACLIST(SURROGAT) REFRESH
Creating the STARTED class profile for the daemon ...
RDEFINE STARTED PKISRVD.* STDATA(USER(PKISRVD))
SETOPTS CLASSACT(STARTED) RACLIST(STARTED)
SETOPTS RACLIST(STARTED) REFRESH
Allowing PKISERV to request certificate functions ...
SETR GENERIC(FACILITY)
RDEFINE FACILITY IRR.RPKISERV.**
PERMIT IRR.RPKISERV.** CLASS(FACILITY) ID(PKISERV) ACCESS(CONTROL)
Creating the profile to protect PKI Admin functions ...
RDEFINE FACILITY IRR.RPKISERV.PKIADMIN
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY) ID(PKIGRP) ACCESS(UPDATE)
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY) ID(PKISERV) ACCESS(NONE)
SETOPTS RACLIST(FACILITY) REFRESH

```

```

-----
Information needed for PKI Services UNIX set up:
-----

```

The daemon user ID is:  
PKISRVD

The VSAM high level qualifier is:  
PKISRVD  
This is needed for the [ObjectStore] section in pkiserv.conf

The PKI Services' DER encoded certificate is in data set:  
'PKISRVD.PRIVATE.CACERT.DERBIN'  
This must be OPUT to /var/pkiserv/cacert.der with the BINARY option

The fully qualified PKI Services' SAF keyring is:  
PKISRVD/CAring  
This is needed for the [SAF] section in pkiserv.conf

The PKI Services CA DN is:  
OU=Human Resources Certificate Authority,O=Your Company,C=Your Country 2 Letter Abbreviation  
The suffix must match the LDAP suffix in slapd.conf

The webserver's SAF keyring is:  
SSLring  
This is needed for the KeyFile directive in httpd\*.conf files

The Webserver's DN is:  
CN=www.YourCompany.com,O=Your Company,L=Your City,ST=Your Full State or  
Province Name,C=Your Country 2 Letter Abbreviation  
The left most RDN must be the webserver's fully qualified domain name



## Chapter 5. Configuring the UNIX runtime environment

You need to perform all of the tasks in this chapter if you are configuring PKI Services for the first time. If you have already configured PKI Services for an earlier release, you may need to perform some of the tasks in this chapter if you are:

- Using a sysplex for PKI Services daemons
- Adding an additional CA domain
- Planning to use a PKI Services utility
- Enabling Simple Certificate Enrollment Protocol (SCEP)
- Sending e-mail notification for certificates ready for retrieval or expiration or rejected certificate requests
- Customizing certificate revocation list (CRL) distribution point processing

After the RACF administrator performs the tasks necessary to set up PKI Services, the UNIX programmer needs to perform the following tasks:

- If necessary, copy files.
- If necessary, update the environment variables file.
- If necessary, update the configuration file.
- If configuring PKI Services for the first time or adding a new CA domain, set up the `/var/pkiserv` directory.

The following table summarizes information about copying and updating files. To view the contents of any of these files, see Chapter 26, “Other code samples,” on page 371.

Table 18. Deciding which files to copy and change

File	Purpose	Need to copy?	Need to change?
<code>expiringmsg.form</code>	The form for an e-mail sent to a user when a certificate is going to expire.	Only if your company sends an e-mail notification to a user about a certificate that is going to expire	<b>Guideline:</b> Make no changes to this file until later. See Chapter 12, “Customizing the administration Web pages,” on page 141 for details about making changes.

## Configuring the UNIX runtime environment

Table 18. Deciding which files to copy and change (continued)

File	Purpose	Need to copy?	Need to change?
pkiserv.conf	Configuration file. Contains various settings and values PKI Services needs.	Only if you are configuring PKI Services for the first time.	<p>The UNIX programmer may need to change the <b>LDAP</b> section of this file. <b>Guideline:</b> Do not change it now but change it later when you perform “Steps for tailoring the LDAP section of the configuration file” on page 72.</p> <p>The UNIX programmer needs to update the non-LDAP section of the pkiserv.conf configuration file if any of the following is true:</p> <ul style="list-style-type: none"> <li>• You intend to run multiple instances of PKI Services in a sysplex.</li> <li>• You intend to run multiple CA domains on a single z/OS image. (See “Adding a new CA domain” on page 161.)</li> <li>• You are customizing certificate revocation list (CRL) distribution point processing.</li> <li>• You are migrating from z/OS V1R3 and you intend to send e-mail notifications for certificate-related events.</li> <li>• You are configuring PKI Services for the first time and do not intend to send e-mail notifications for certificate-related events.</li> </ul>
pkiserv.envars	The environment variables file.	Only if you are configuring PKI Services for the first time and the file needs changes.	UNIX programmer may have to update this file. See “Optionally updating PKI Services environment variables” on page 47.
pkiserv.tpl	Certificate templates file. Contains HTML-style code that builds the Web pages underlying certificate requests.	Only if you are configuring PKI Services for the first time.	<b>Guideline:</b> Make no changes to this file until later. See Chapter 12, “Customizing the administration Web pages,” on page 141 for details about making changes.
readymsg.form	The form for an e-mail sent to a user when the PKI Services administrator has approved a certificate request and the certificate is ready for retrieval.	Only if your company sends an e-mail notification to a user after the PKI Services administrator has approved a certificate request and the certificate is ready for retrieval.	<b>Guideline:</b> Make no changes to this file until later. See Chapter 12, “Customizing the administration Web pages,” on page 141 for details about making changes.

Table 18. Deciding which files to copy and change (continued)

File	Purpose	Need to copy?	Need to change?
rejectmsg.form	The form for an e-mail sent to a user when the PKI Services administrator has rejected a certificate request.	Only if your company sends an e-mail notification to a user after the PKI Services administrator has rejected a certificate request	<b>Guideline:</b> Make no changes to this file until later. See Chapter 12, “Customizing the administration Web pages,” on page 141 for details about making changes.

## Steps for copying files

### Before you begin:

- You need to obtain the following document:  
*z/OS UNIX System Services Planning*
- You need to know the file system directory where the MVS programmer installed PKI Services and the runtime directory, *install-dir* and *runtime-dir* in the commands that follow. The defaults are `/usr/lpp/pkiserv/` and `/etc/pkiserv` respectively. The MVS programmer was asked to record any changes to these defaults; see Table 3 on page 9.
- The user ID you use for copying files must have superuser authority.

Perform the following steps to copy the files:

1. If you are configuring PKI Services for the first time, copy the configuration and template files by entering the following commands from the UNIX command line.

**Note:** To use these commands, your user ID must have super user authority.

```
cp -p /install-dir/samples/pkiserv.conf runtime-dir
cp -p /install-dir/samples/pkiserv.tmpl runtime-dir
```

2. If your company is sending e-mail notifications to users (when certificate requests are rejected or when certificates are ready for retrieval or expiring), copy the appropriate notification files from the samples directory to the runtime directory by entering commands such as the following:

```
cp -p /install-dir/samples/rejectmsg.form runtime-dir
cp -p /install-dir/samples/readymsg.form runtime-dir
cp -p /install-dir/samples/expiringmsg.form runtime-dir
```

3. If you are configuring PKI Services for the first time, examine the values in the environment variables file (by default, `pkiserv.envars`). If any values need to change (such as the `OCSFREGDIR`, the environment variable for the OCSF registry directory—see Step 2 on page 19), copy this file by entering the following command:

```
cp -p /install-dir/samples/pkiserv.envars runtime-dir
```

## Optionally updating PKI Services environment variables

You need to perform this task only if any one of the following conditions is true:

- You are configuring PKI Services for the first time.
- You are adding an additional CA domain.

## Configuring the UNIX runtime environment

- You want to send e-mail notifications (for rejected certificate requests or certificates that are ready for retrieval or expiring) and you did not use the default location for sendmail (/usr/sbin/sendmail).
- You are migrating from z/OS V1R3 where you are running with customized environment variables and you want to send e-mail notifications.

You need to define certain environment variables (such as LIBPATH) for the PKI Services daemon to run. There are two files related to environment variables.

- A sample environment variables file, pkiserv.envs (by default in /usr/lpp/pkiserv/samples/)
- SYS1.PROCLIB member PKISERVD (You can use the ENVAR parameter to point to the environment variables file.)

You can use pkiserv.envs to set environment variables for the PKI Services daemon. This file contains most of the environment variables needed to run the daemon.

You need to change the file if you did not use the default for any of the following:

- The install directory for PKI Service (/usr/lpp/pkiserv)
- The message level
- The location of the OCSF registry directory (/var/ocsf)
- The location for sendmail (/usr/sbin/sendmail)

**Guideline:** If you need to make changes to the pkiserv.envs file, copy the file another directory (such as /etc/pkiserv) and make changes only to the copy.

PKISERVD is the sample procedure to start PKI Services. (For sample code, see “PKISERVD sample procedure to start PKI Services daemon” on page 384.) PKISERVD sets the **TZ** (time zone) environment variable because it is very likely that the value of this variable needs to change. PKISERVD also includes parameters specifying the directory containing the environment variables file (DIR) and the file name of the environment variables file (FN). If you make a copy of pkiserv.envs as suggested, you also need to change the name of the directory in PKISERVD (for example, DIR="/etc/pkiserv") and possibly the file name (for example, FN="pki.env").

**Note:** You can change all of the following on the START command:

- environment variables directory
- file name
- job output class
- region size
- standard output
- standard error
- time zone.

See “Steps for starting the PKI Services daemon” on page 85.

Because of the limitation of the number of characters allowed in the PARM=*operand* on the JCL EXEC card, take care to ensure that the total length of the environment variables directory and file name, **TZ** value, and stdout and stderr redirection values do not exceed the 100 character maximum.

You must specify any environment variables that PKI Services requires either in the PKISERVD procedure or in the environment variables file (`pkiserv.envars`).

**Guideline:** Make your additions and changes to the environment variables file, rather than to the PKISERVD procedure.

### (Optional) Steps for updating PKI Services environment variables

**Before you begin:** See page 47 to determine if you need to update environment variables.

Perform the following steps to update PKI Services environment variables:

1. Examine the values in the environment variables file (by default, `pkiserv.envars`) and update the file as necessary. (See “Environment variables in the environment variables file” on page 347 for a description of the environment variables and “The `pkiserv.envars` environment variables file” on page 349 for a code sample of the environment variables file.)

**Notes:**

- a. If the value set for the OCSF registry directory differs from the default value of `'/var/ocsf'`, you need to update the `OCSFREGDIR` environment variable.
- b. If you did not install sendmail in its default location (`/usr/sbin`), you need to update the `PATH` environment variable.

- 
2. Make any needed changes to PKISERVD, such as updating the pathname of the environment variables file (FN and DIR parameters). (See “PKISERVD sample procedure to start PKI Services daemon” on page 384 for a code sample of the PKISERVD procedure.)
- 
3. If you are migrating from z/OS V1R3 where you are already running with a customized environment variables file and you want to send e-mail notifications, you need to add a `PATH` statement to your environment variables file. If you installed sendmail in its default location (`/usr/sbin`), then you can copy the `PATH` statement from the sample file shipped with PKI Services (`/usr/lpp/samples/pkiserv.envars`). Otherwise add a `PATH` statement such as the following:

```
PATH=/directory-where-sendmail-resides
```

---

### Optionally updating the `pkiserv.conf` configuration file

You need to update the `pkiserv.conf` configuration file if you meet any of the following conditions:

- You are configuring PKI Services for the first time
- You are adding support for:
  - Running a sysplex for PKI Services daemon
  - Sending e-mail notifications to users if the PKI Services administrator rejects certificate requests or certificates are ready for retrieval or expiring
  - Customizing certificate revocation list (CRL) distribution point processing. (See “Customizing distribution point CRLs” on page 150 for details.)

You can also optionally update the file if you want to change certain default values.

## Configuring the UNIX runtime environment

The `pkiserv.conf` configuration file for the PKI Services daemon consists of sections of name-value pairs. **Important:** Everything in the `pkiserv.conf` file—including section names, keys, and values—is case-sensitive.

Each section of the `pkiserv.conf` configuration file has a title enclosed in square brackets. The configuration file includes the following sections:

**[OIDs]** The OIDs section specifies the object identifiers for various nicknames PKI Services uses internally. The OIDs are specified in the following form:  
*name=dotted-decimal*

The following excerpt is from the OIDs section:

```
[OIDs]
:
MyPolicy=1.2.3.4
```

**[ObjectStore]** The ObjectStore section specifies operational information for various files and data sets.

The following excerpt is from the ObjectStore section:

```
[ObjectStore]
ObjectDSN='pkisrzd.vsam.ost'
:
```

**[CertPolicy]** The CertPolicy section is for CA policy information.

The following excerpt is from the CertPolicy section:

```
[CertPolicy]
SigAlg1=sha-1WithRSAEncryption
:
```

**[General]** The General section is for general information.

The following excerpt is from the General section:

```
[General]
InitialThreadCount=10
:
```

**[SAF]** The SAF section is for information about the SAF (RACF) key ring that is used for CA certificate and private key storage.

The following excerpt is from the SAF section:

```
[SAF]
KeyRing=PKISRVD/CARing
```

**[LDAP]** The LDAP section contains information about the LDAP server for posting certificates and CRLs.

The following excerpt is from the LDAP section:

```
[LDAP]
NumServers=1
:
```

The UNIX programmer needs to update the **LDAP** section of this file. **Guideline:** Do not change it now but change it later when you perform “Steps for tailoring the LDAP section of the configuration file” on page 72.



## (Optional) Steps for updating the configuration file

**Before you begin:** You need to update the `pkiserv.conf` configuration file if you meet any of the following conditions:

- You are configuring PKI Services for the first time.
- You are adding support for:
  - Running a sysplex for PKI Services daemon
  - Sending e-mail notifications to users if the PKI Services administrator rejects certificate requests or certificates are ready for retrieval or expiring
  - Certificate revocation list (CRL) distribution point processing.
- You can also optionally update the file if you want to change certain default values.

The following table provides information about parameters in the `pkiserv.conf` configuration file. (It omits parameters for the **LDAP** section. For information about these parameters, see Table 23 on page 72.) Read the parameter descriptions, and examine the values provided in the sample configuration file—shown in the rightmost column—to ensure that the values meet your company’s requirements. As necessary, cross out the sample values and enter the information appropriate to your own organization’s needs and policies.

Table 19. Information needed for updating the configuration file

Parameter	Information needed	Where to get this information	Sample value or your customized value
<b>OIDs section</b>			
MyPolicy=	A registered Object ID identifying your organization’s usage policy, for example: 1.2.3.4	If you are creating your own certificate policy, see “Using certificate policies” on page 146 for information on creating certificate policies. Otherwise, do not change this information.	<b>1.2.3.4</b>  If you need to use the CertificatePolicies extension, replace 1.2.3.4 with the value of your Object ID:  _____
<b>ObjectStore section</b>			
ObjectDSN=	VSAM data set name for the ObjectStore base cluster. This is the request database. Each VSAM request record consists of a fixed header followed by a variable-length section.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	<b>'pkisrvd.vsam.ost'</b>  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrvd.employee.vsam.ost'		

## Configuring the UNIX runtime environment

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ObjectTidDSN=	VSAM data set name for the ObjectStore transaction ID (TID) alternate index.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrzd.vsam.ost.path'  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrzd.employee.vsam.ost.path'		
ObjectStatusDSN=	VSAM data set name for the ObjectStore status alternate index.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrzd.vsam.ost.status'  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrzd.employee.vsam.ost.status'		
ObjectRequestorDSN=	VSAM data set name for the ObjectStore requestor alternate index.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrzd.vsam.ost.requestr'  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrzd.employee.vsam.ost.requestr'		
ICLDSN=	VSAM data set name for the ICL base cluster.  This contains the certificates that have been issued. Each VSAM ICL record consists of a fixed header followed by a variable-length section containing the BER-encoded certificates.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrzd.vsam.icl'  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrzd.employee.vsam.icl'		

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ICLStatusDSN=	VSAM data set name for ICL status alternate index.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrwd.vsam.icl.status'  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrwd.employee.vsam.icl.status'		
ICLRequestorDSN=	VSAM data set name for ICL requestor alternate index.	For the high-level qualifier before the period, see the <i>vsamhlq</i> variable in Table 17 on page 36. The name of the file (after the period) can change; the MVS programmer who creates the VSAM data sets usually decides these names.	'pkisrwd.vsam.icl.requestr'  Note that this begins with the VSAM high-level qualifier.
	<b>Guideline:</b> If you are adding a new CA domain, insert the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the data set name. <b>Example:</b> 'pkisrwd.employee.vsam.icl.requestr'		
RemoveCompletedReqs=	Time period that completed certificate requests remain in the ObjectStore before automatic deletion. This is a number followed by d (days) or w (weeks).	UNIX programmer decides this value.	1w
RemoveInactiveReqs=	Time period that incomplete, inactive certificate requests remain in the ObjectStore before automatic deletion. This is a number followed by d (days) or w (weeks).	UNIX programmer decides this value.	4w
RemoveExpiredCerts=	Time period that expired certificates remain in the ICL before automatic deletion. This is a number followed by d (days) or w (weeks). If you do not specify this parameter, or you set the value to 0d, expired certificates will not be removed.	UNIX programmer decides this value.	0d

## Configuring the UNIX runtime environment

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
SharedVSAM=	Indicates whether you intend to share a single copy of the PKI Services VSAM data sets among multiple images in a sysplex. This is T (True) or F (False).	UNIX programmer decides this value.	<b>F</b>
<b>CertPolicy section</b>			
ARLDist=	Indicates whether an authority revocation list (ARL) distribution point will be created. F (the default) indicates no ARL distribution point will be created. T indicates that an ARL distribution point will be created if CRLDistSize is greater than zero.	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Creating a distribution point ARL” on page 156 for more information.	<b>F</b>
CPS1=	The Uniform Resource Identifier (URI) where your organization’s Certification Practice Statement (CPS) is located. This is in the form:  <a href="http://www.mycompany.com/cps.html">http://www.mycompany.com/cps.html</a>	Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<b><a href="http://www.mycompany.com/cps.html">http://www.mycompany.com/cps.html</a></b>  If you changed PolicyRequired=F to PolicyRequired=T, you need to replace the variable <i>mycompany</i> with your own value for this:  <b><a href="http://www._____.com/cps.html">http://www._____.com/cps.html</a></b>
CreateInterval=	How often the certificate creation thread scans the database for approved requests. This is a number followed by w (weeks), d (days), h (hours), m (minutes), or s (seconds).	UNIX programmer decides this value.	<b>3m</b>
CRLDistDirPath=	The full path for the file system directory where PKI Services is to save each DP CRL, as specified by the HTTP URI in the CRLDistributionPoints extension. This value is ignored if you do not create a CRLDistributionPoints extension or if the URI protocol is ldap. This value can be specified with or without the trailing slash.  The default value is <code>/var/pkiserv/</code> .	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Customizing distribution point CRLs” on page 150 for more information.	<b><code>/var/pkiserv/</code></b>

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
CRLDistName=	Constant portion of the (leaf-node) relative distinguished name for a distribution point (DP) CRL, if DP CRL processing is being performed.  The default value is CRL.	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Customizing distribution point CRLs” on page 150 for more information.	<b>CRL</b>
CRLDistSize=	An integer value that represents the maximum number of certificates that may appear on one DP CRL.  If you do not specify this parameter, or you set the value to 0, DP CRLs will not be created.	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Customizing distribution point CRLs” on page 150 for more information.	<b>500</b>
CRLDistURI=	<i>Optional:</i> Specifies a URI format name for the DP CRL. You can specify multiple names using parameters CRLDistURI1, CRLDistURI2, and so forth. This value is ignored if you do not create DP CRLs by specifying CRLDistSize with a value greater than zero. Specify this only if you want a URI-format name, in addition to the distinguished name format, built in the CRLDistributionPoints extension.	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Customizing distribution point CRLs” on page 150 for more information.	—
CRLDuration=	The amount of time that a certificate revocation list is valid. This is a number followed by w (weeks), d (days), h (hours), m (minutes), or s (seconds).	UNIX programmer decides this value.	<b>2d</b>
EnableSCEP=	Specifies whether Simple Certificate Enrollment Protocol (SCEP) is allowed. This is T (True) or F (False).	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175 for more information.	<b>F</b>

## Configuring the UNIX runtime environment

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ExpireWarningTime=	<p><b>Note:</b> You need a value for this parameter only if you are sending e-mail notifications to users when certificates are expiring.</p> <p>This parameter indicates how soon before certificate expiration to send a warning message (that is, the number of days or weeks before the day and time the certificate expires).</p> <p>This name-value pair is optional. Its absence indicates no expiration checking is performed. Also, if the name-value pair is present but has an incorrect value or if PKI Services is configured to operate without LDAP, no expiration checking is done.</p>	UNIX programmer decides this value.	<b>4w</b>
MaxSuspendDuration=	The length of certificate suspension grace period in weeks or days. This is a number followed by w (weeks) or d (days). Certificates that remain suspended for longer than this period are automatically revoked. If you do not specify this parameter, or you set it to 0d, the grace period is unlimited.	UNIX programmer decides this value.	<b>120d</b>
OCSPTYPE=	<p>The type of OCSP responder support desired:</p> <ul style="list-style-type: none"> <li>• none (the default)</li> <li>• basic</li> </ul> <p>If you do not specify this parameter, or you set the value to none, the responder is not enabled.</p>	Change to basic if you want to enable the responder.	<b>none</b>
PolicyCritical=	Indicates whether the CertificatePolicies extension created on a global basis (if PolicyRequired=T was specified) should be marked critical. This is T (True) or F (False). This field is ignored if PolicyRequired=F is specified.	<p>UNIX programmer decides this value.</p> <p>Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.</p>	<b>F</b>

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
PolicyRequired=	Indicates whether the CertificatePolicies extension should be created on a global basis. This is T (True) or F (False). PolicyRequired=T indicates that the CertificatePolicies extension will be created with the same value for all certificate templates based on the keywords specified in the <b>CertPolicy</b> section of the configuration file. Any policies specified through the CONSTANT subsection in the template file will be ignored. PolicyRequired=F indicates that the policies specified through the CONSTANT subsection, if any, will be used.	UNIX programmer decides this value.  Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<b>F</b>
PolicyName1=	The name of the policy. (This is the same policy name used with the MyPolicy parameter of the <b>OIDs</b> section.)	Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<i>MyPolicy</i>  If you changed PolicyRequired=F to PolicyRequired=T, replace the name <i>MyPolicy</i> with the same policy name used in the <b>OIDs</b> section.  _____
Policy1Org=	This is the organization name for the CertificatePolicies extension. For example: International Business Machines, Inc.	Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<b>My Company, Inc.</b>  If you changed PolicyRequired=F to PolicyRequired=T, you need to specify your own value for this:  _____
Policy1Notice1=	The first company notice number.	Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<b>1</b>  If you changed PolicyRequired=F to PolicyRequired=T, you need to specify your own value for this:  _____
Policy1Notice2=	The second company notice number.	Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<b>17</b>  If you changed PolicyRequired=F to PolicyRequired=T, you need to specify your own value for this:  _____

## Configuring the UNIX runtime environment

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
SigAlg1=	<p>The Object ID for the signature algorithm.</p> <p>If the certificate key type is RSA, the SigAlg1 algorithm value must be one of the following:</p> <ul style="list-style-type: none"> <li>• sha-1WithRSAEncryption (the default)</li> <li>• md-5WithRSAEncryption</li> <li>• md-2WithRSAEncryption</li> </ul> <p>If the certificate key type is DSA, the SigAlg1 algorithm value must be as follows:</p> <ul style="list-style-type: none"> <li>• id-dsa-with-sha1</li> </ul> <p><b>Note:</b> Changing the default also requires adding a line in the <b>OIDs</b> section. See “Updating the signature algorithm” on page 149.</p>	Do not change this information until you perform advanced customization. See “Updating the signature algorithm” on page 149 for more information.	<b>sha-1WithRSAEncryption</b>
TimeBetweenCRLs=	<p>How often a certificate revocation list should be created. This is a number followed by w (weeks), d (days), h (hours), m (minutes), or s (seconds).</p> <p><b>Note:</b> If you change this value after PKI Services has been in operation and then restart PKI Services, the change does not take effect until after the next CRL is created.</p>	UNIX programmer decides this value.	<b>1d</b>
UserNoticeText1=	A legal statement about certificate issuance and use. For example: Certificate for IBM internal use only	Do not change this information until you perform advanced customization. See “Using certificate policies” on page 146 for more information.	<p><i>statement</i></p> <p>If you changed PolicyRequired=F to PolicyRequired=T, you need to replace the variable <i>statement</i> with your own value for this:</p> <p>_____</p> <p>_____</p>
<b>General section</b>			
InitialThreadCount=	Number of threads (at least 2 and no more than 100) the PKI Services daemon should create at program initialization.	UNIX programmer decides this value.	<b>10</b>



Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
ReadyMessageForm=	The full pathname or data set name containing the 'Your certificate is ready' message form. Using this name-value pair is optional. If you do not specify this name-value pair, no message is sent.	UNIX programmer decides this value.	<b>/etc/pkiserv/readymsg.form</b>
	<b>Guideline:</b> If you are adding a new CA domain, use the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the pathname. <b>Example:</b> /etc/pkiserv/employees/readymsg.form		
RejectMessageForm=	The full pathname or data set name containing the 'Your certificate request has been rejected' message form. By default, no message is issued. Using this name-value pair is optional.	UNIX programmer decides this value.	<b>/etc/pkiserv/rejectmsg.form</b>
	<b>Guideline:</b> If you are adding a new CA domain, use the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the pathname. <b>Example:</b> /etc/pkiserv/employees/rejectmsg.form		
ExpiringMessageForm=	The full pathname or data set name containing the 'Your certificate is about to expire' message form. By default, no message is issued. If your team has specified a value for ExpireWarningTime (see the ExpireWarningTime row in this table), then ExpiringMessageForm is required. Otherwise an error is logged and no expiring message processing is performed.	UNIX programmer decides this value.	<b>/etc/pkiserv/expiringmsg.form</b>
	<b>Guideline:</b> If you are adding a new CA domain, use the <i>ca_domain</i> value from Table 17 on page 36 as the second qualifier in the pathname. <b>Example:</b> /etc/pkiserv/employees/expiringmsg.form		
<b>SAF section</b>			
KeyRing=	The fully qualified name of the SAF key ring for PKI Services to use. (This must consist of an uppercase user ID and a case-sensitive ring name separated by a slash (/).)	See the <i>ca_ring</i> and <i>daemon</i> values in Table 17 on page 36.	<b>PKISRVD/CARing</b>
RA_label=	The label of your PKI Services registration authority (RA) certificate.	See the <i>ra_label</i> value in Table 11 on page 29.	<b>Local PKI RA</b>
<b>LDAP section</b>			

## Configuring the UNIX runtime environment

Table 19. Information needed for updating the configuration file (continued)

Parameter	Information needed	Where to get this information	Sample value or your customized value
For information about the <b>LDAP</b> section, see Table 23 on page 72.			

Perform the following steps to update the pkiserv.conf configuration file:

**Note:** Keep in mind that everything in the pkiserv.conf file—including section names, keys, and values—is case-sensitive.

1. If necessary, update the **ObjectStore** section:

- a. If you are configuring PKI Services for the first time, you can omit the following change. (The pkiserv.conf configuration file that is shipped with the product starting in z/OS V1R4 does not contain the Name= line. This line was included in z/OS V1R3.)

If you are migrating from z/OS V1R3 for PKI Services, you may have updated the value pkica on the following line. Leave this line in the pkiserv.conf file until migration is completed. You can tell when migration has completed because the PKI Services daemon renames the file system files, appending .MIGRATED to the file names. After migration is completed, you can delete the Name= line if you wish:

Name=**pkica**

If you are configuring PKI Services for the first time, you can omit the following change. (The pkiserv.conf configuration file that is shipped with the product starting in z/OS V1R4 does not contain the Path= line. This line was included in z/OS V1R3.)

If you are migrating from z/OS V1R3, you may have updated the value /var/pkiserv on the following line. Leave this line in the pkiserv.conf file until migration is completed. After this, you can delete the Path= line if you wish:

Path=**/var/pkiserv**

- b. If necessary, change the data set names in the following lines to the names you chose in the ObjectDSN=, ObjectTidDSN=, ObjectStatusDSN=, ObjectRequestorDSN=, ICLDSN=, ICLStatusDSN=, and ICLRequestorDSN= rows in Table 19 on page 51:

```
ObjectDSN='pkisrvd.vsam.ost'
ObjectTidDSN='pkisrvd.vsam.ost.path'
ObjectStatusDSN='pkisrvd.vsam.ost.status'
ObjectRequestorDSN='pkisrvd.vsam.ost.requestr'
ICLDSN='pkisrvd.vsam.icl'
ICLStatusDSN='pkisrvd.vsam.icl.status'
ICLRequestorDSN='pkisrvd.vsam.icl.requestr'
```

If you are configuring PKI Services for the first time be aware that the high-level qualifier of the VSAM data set names must match the name of the RACF user ID assigned to the PKI Services daemon (by default, PKISRVD). If you change from the default to another user ID, you need to change the high-level qualifier in the pkiserv.conf configuration file as well. If the MVS programmer changes the data set names (see Step 2d on page 80), you must make equivalent changes in pkiserv.conf.

If you are migrating from z/OS Version 1 Release 4 or lower and you want to use a sysplex, you need to set up VSAM record-level sharing (RLS). This involves running IKYRVSAM to reallocate your VSAM data sets. It may also involve changing the names of the source and destination data

sets in IKYRVSAM. If you change the names of the destination data sets in IKYRVSAM to different names than you currently have in the pkiserv.conf configuration file, then you need to update pkiserv.conf as well.

- C. If necessary, change **1w** in the following line to the value in the RemoveCompletedReqs= row in Table 19 on page 51:  
RemoveCompletedReqs=**1w**
- d. If necessary, change **4w** in the following line to the value in the RemoveInactiveReqs= row in Table 19 on page 51:  
RemoveInactiveReqs=**4w**
- e. If necessary, uncomment the following line and, optionally, change **26w** to the value in the RemoveExpiredCerts= row in Table 19 on page 51:  
RemoveExpiredCerts=**26w**
- f. If necessary, update the SharedVSAM lines:
  - If you intend to use sysplex and you are migrating from z/OS V1R3 for PKI Services, locate the following lines in the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) and copy them to the bottom of the **ObjectStore** section; then change F in the last line to T.  
# Are the VSAM data sets shared in a sysplex with other instances  
# of PKI Services. True (T) or False (F)  
SharedVSAM=F
  - If you intend to use a sysplex and you are configuring PKI Services for the first time, change F in the following line to T:  
SharedVSAM=F
  - If you are not using a sysplex (regardless of whether you are migrating from z/OS V1R3 or configuring PKI Services for the first time), you do not need to do anything.

## 2. If necessary, update the **CertPolicy** section.

- a. If necessary, change **3m** in the following line to the value in the CreateInterval= row in Table 19 on page 51:  
CreateInterval=**3m**
- b. If necessary, update the ExpireWarningTime line(s):
  - If you are sending e-mail notifications (about rejected certificate requests or certificates ready for retrieval or expiring) and you are migrating from z/OS V1R3, locate the following lines in the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) and copy them into the **CertPolicy** section (after the CreateInterval parameter). If necessary, change the value 4w to the value in the ExpireWarningTime row of Table 19 on page 51.  
# when the warning message should be issued. (i.e. the number of days  
# or weeks before the certificate expiration date/time). Defaults to never  
ExpireWarningTime=4w
  - If you are sending e-mail notifications and you are configuring PKI Services for the first time, if necessary change the value 4w in the following line to the value in the ExpireWarningTime row of Table 19 on page 51.  
ExpireWarningTime=4w

## Configuring the UNIX runtime environment

- If you are not using e-mail notifications and you are configuring PKI Services for the first time, remove the `ExpireWarningTime=4` line from the `pkiserv.conf` file.
  - If you are not using e-mail notifications and you are migrating from z/OS V1R3, you do not need to do anything.
- C. If necessary, change **1d** in the following line to the value in the `TimeBetweenCRLs=` row in Table 19 on page 51:
- `TimeBetweenCRLs=1d`
- d. If necessary, change **2d** in the following line to the value in the `CRLDuration=` row in Table 19 on page 51:
- `CRLDuration=2d`
- e. If necessary, change **F** in the following line to the value in the `PolicyRequired=` row in Table 19 on page 51:
- `PolicyRequired=F`
- For more information on this parameter, see “Using certificate policies” on page 146.
- f. If necessary, change **F** in the following line to the value in the `PolicyCritical=` row in Table 19 on page 51:
- `PolicyCritical=F`
- For more information on this parameter, see “Using certificate policies” on page 146.
- g. If necessary, change **120d** in the following line to the value in the `MaxSuspendDuration=` row in Table 19 on page 51:
- `MaxSuspendDuration=120d`
- h. If necessary, establish distribution point (DP) certificate revocation lists (CRLs) and a DP authority revocation list (ARL). Follow the procedure shown in “Steps for customizing distribution point CRLs” on page 153 to determine the values for Table 19 on page 51.
- i. If you wish to enable the OCSP responder, change `OCSPType=none` to `OCSPType=basic`.
- j. If you wish to enable Simple Certificate Enrollment Protocol (SCEP), change `EnableSCEP=F` to `EnableSCEP=T`. (See “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.)

---

3. If necessary, update the **General** section:

- a. If necessary, change **10** in the following line to the value in the `InitialThreadCount` row in Table 19 on page 51:
- `InitialThreadCount=10`
- b. If necessary update the `ReadyMessageForm`, `RejectMessageForm`, and `ExpiringMessageForm` lines:
- If you are sending e-mail notifications (about rejected certificate requests or certificates that are ready for retrieval or expiring) and you are migrating from z/OS V1R3, copy the following lines from the sample configuration file (`/usr/lpp/pkiserv/samples/pkiserv.conf`) into your `pkiserv.conf` configuration file (at the bottom of the **General** section). If

necessary, change the values of the pathname in the uncommented lines to the corresponding values in Table 19 on page 51.

```
# full pathname or data set name containing the 'your certificate is ready'  
# message form. Defaults to no message issued  
ReadyMessageForm=/etc/pkiserv/readymsg.form
```

```
# full pathname or data set name containing the 'your certificate request  
# has been rejected' message form. Defaults to no message issued  
RejectMessageForm=/etc/pkiserv/rejectmsg.form
```

```
# full pathname or data set name containing the 'your certificate is about  
# to expire' message form. Defaults to no message issued  
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form
```

- If you are sending e-mail notifications and you are configuring PKI Services for the first time, if necessary, change the values of the pathname in following three lines to the corresponding values in Table 19 on page 51:

```
ReadyMessageForm=/etc/pkiserv/readymsg.form
```

```
RejectMessageForm=/etc/pkiserv/rejectmsg.form
```

```
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form
```

- If you are not sending e-mail notifications and you are configuring PKI Services for the first time, delete all of the following lines in the pkiserv.conf configuration file:

```
# full pathname or data set name containing the 'your certificate is ready'  
# message form. Defaults to no message issued  
ReadyMessageForm=/etc/pkiserv/readymsg.form
```

```
# full pathname or data set name containing the 'your certificate request  
# has been rejected' message form. Defaults to no message issued  
RejectMessageForm=/etc/pkiserv/rejectmsg.form
```

```
# full pathname or data set name containing the 'your certificate is about  
# to expire' message form. Defaults to no message issued  
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form
```

- If you are not sending e-mail notification and you are migrating from z/OS V1R3, you do not need to do anything.

---

#### 4. If necessary, update the **SAF** section:

- a. If necessary, change **PKISRVD/CAring** in the following line to the value in the KeyRing= row in Table 19 on page 51:

```
KeyRing=PKISRVD/CAring
```

- b. If you specified EnableSCEP=T in Step 2j on page 62, change **Local PKI RA** in the following line to the value in the ra\_label= row in Table 11 on page 29:

```
RALabel=Local PKI RA
```

---

## Steps for setting up the var directory

You need to perform this task only if you are configuring PKI Services for the first time or adding a new CA domain.

# Configuring the UNIX runtime environment

**Before you begin:** Replace the following default values (used in the command examples) with values appropriate for your configuration:

Default value	Your value
PKISRVD	Use your daemon value in Table 17 on page 36.
'pkisrvd.cacert.derbin'	Use your export_dsn value in Table 17 on page 36.
/var/pkiserv	<b>Guideline:</b> Use your ca_domain value from Table 17 on page 36 to qualify the directory location if you are adding a new CA domain. For example, /var/pkiserv/employees.

Perform the following steps to set up a UNIX directory and copy certain files that PKI Services needs into that directory:

1. Change ownership of the directory to the user ID of the PKI Services daemon by entering the following command from the UNIX command line:

**Example:**

```
chown PKISRVD /var/pkiserv
```

2. Copy the CA certificate from its MVS data set to cacert.der in the /var/pkiserv directory by entering the following command from the UNIX command line:

**Example:**

```
cp "'pkisrvd.cacert.derbin'" /var/pkiserv/cacert.der
```

3. Change the permission settings of the file by entering the following command from the UNIX command line:

**Example:**

```
chmod 644 /var/pkiserv/cacert.der
```

4. Change the ownership of the file by entering the following command from the UNIX command line:

**Example:**

```
chown pkisrvd /var/pkiserv/*
```

## Chapter 6. Tailoring LDAP configuration for PKI Services

You need to tailor LDAP configuration only if you are configuring PKI Services for the first time.

The directions in this section are for using the z/OS Integrated Security Services LDAP server for PKI Services. If you intend to use a different LDAP product, you need to refer to the documentation for this product. See Appendix A, “LDAP directory server requirements,” on page 387 for information about installing a non-z/OS LDAP.

The LDAP programmer needs to update the `schema.user.ldif` file so that the LDAP server understands the format of entries that will be stored in the directory.

### Steps for updating `schema.user.ldif`

#### Before you begin:

- You will need LDAP programming skills to complete this procedure.
- Make sure that the LDAP server is started before beginning these steps. If you are unsure about this, see “Steps for installing and configuring LDAP” on page 20.
- You need to know the following information from LDAP installation. Copy the information into the following table from (completed) Table 9 on page 21:

Table 20. LDAP information you need for tailoring LDAP configuration

LDAP information	Explanation	Value
Administrator's distinguished name	This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 9 on page 21. The LDAP administrator defines the administrator's distinguished name with the <b>adminDN</b> keyword in the LDAP server configuration file. For example, the value is “cn=Admin” in the following: <code>adminDN "cn=Admin"</code> )	
Administrator password	This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example: <ul style="list-style-type: none"><li>• By specifying the password as a TDBM entry by using the <b>userPassword</b> attribute in the <b>ldif2tdbm</b> load utility</li><li>• By using the <b>adminPW</b> keyword in the LDAP server configuration file (not suggested)</li></ul>	
LDAP fully qualified domain name and port	This is the IP address and port on which the LDAP server is listening. For example, for <code>ldap.widgets.com:389</code> , the fully qualified domain name is <code>ldap.widgets.com</code> and the port is 389. See Table 7 on page 19 for a definition of fully qualified domain name.	
Suffix	(For a definition of suffix, see Table 9 on page 21.) The suffix value is specified after the suffix keyword in the LDAP server configuration file. <code>suffix "o=your-company,c=your-country-abbreviation"</code>	

You need to update the `schema.user.ldif` file only if you are configuring PKI Services for the first time.

## Tailoring LDAP configuration for PKI Services

If you have already configured your LDAP schema using a `schema.user.ldif` file that is from before z/OS V1R2 (or if you are using schema files other than `schema.user.ldif`), see *z/OS Integrated Security Services LDAP Server Administration and Use* for instructions on how to change the schema. Refer to the sections about the LDAP directory schema for TDBM and the minimum schema for TDBM.

### PKI Services schema requirements:

1. All portions of the minimum schema
2. RFC2587.ldif and its prerequisites

If you are configuring your LDAP schema for the first time, perform the following steps:

1. Copy the `/usr/lpp/ldap/etc/schema.user.ldif` file to the directory from which you are working by entering the following z/OS UNIX shell command:  

```
cp /usr/lpp/ldap/etc/schema.user.ldif .
```

---
2. Edit the `schema.user.ldif` file in the current directory, ensuring that the “dn:” line (the first line in the file) has the following form and replacing *Your Company Suffix* with the suffix from Table 20 on page 65:  

```
dn: cn=schema, Your Company Suffix
```

---
3. Load the schema defined in the `schema.user.ldif` file into the directory by entering the following command. Replace *admin* and *passwd* with the adminDN and adminPW values from Table 20 on page 65.  

```
ldapmodify -D admin -w passwd -V 3 -f schema.user.ldif
```

---



---

## Chapter 7. Updating z/OS HTTP Server configuration and starting the server

You need to perform the tasks in this chapter only if you are configuring PKI Services for the first time.

Starting the Web server requires having a configuration file for it. This chapter describes how the Web server programmer performs the following tasks:

- Updating the z/OS HTTP Server configuration files by cutting and pasting directives from the PKI Services samples directory into them
- Starting the z/OS HTTP Server.

### Before you begin:

- The z/OS HTTP Server must have already been configured.
- It would be helpful to have available a copy of *z/OS HTTP Server Planning, Installing, and Using*.

---

## Steps for updating the z/OS HTTP Server configuration files

PKI Services uses two modes of SSL, and these two modes require running two instances of the z/OS HTTP Server. Although the two instances share a single server certificate and private key, they use two different configuration files.

- The first configuration file is your existing configuration file (created earlier—see “Steps for installing and configuring the z/OS HTTP Server to work with PKI Services” on page 17). It specifies port 80 for normal HTTP traffic and port 443 for the SSL traffic port.
- The second configuration file, `/etc/httpd1443.conf`, specifies SSL traffic only on port 1443, with client authentication. (If this file does not exist, you create it by copying the first file.)

The following table summarizes the configuration and usage of each Web server:

*Table 21. Summary of configuration and usage of each Web server instance*

Server instance	Protocol	SSL	Server authentication	Client authentication	Port number
First instance	HTTP	No	No	No	80
First instance	HTTPS	Yes	Yes	No	443
Second instance	HTTPS	Yes	Yes	Yes	1443

### Before you begin:

- **Important:** You need to perform these steps only if you are configuring PKI Services for the first time.
- You need to know the file system install directory (the file system directory where the MVS programmer installed PKI Services), called *install-dir* in the commands that follow. The default is `/usr/lpp/pkiserv/`. The MVS programmer was asked to record any changes to the defaults; see Table 3 on page 9.

## Updating z/OS HTTP Server configuration and starting the server

- You need to know the following LDAP information. Record the information in the rightmost row of Table 22. Note that the default name of the LDAP server configuration file is `slapd.conf` for the z/OS Integrated Security Services LDAP server.

Table 22. LDAP information you need for tailoring z/OS HTTP Server configuration

LDAP information	Explanation	Value
Administrator's distinguished name	This is the distinguished name to use for LDAP binding. (For a definition of distinguished name, see Table 9 on page 21.) The LDAP administrator defines the administrator's distinguished name with the <b>adminDN</b> keyword in the LDAP server configuration file. For example, the value is "cn=Admin" in the following: <code>adminDN "cn=Admin"</code>	
Administrator password	This is the password to use for LDAP binding. The LDAP programmer can set this in several ways, for example: <ul style="list-style-type: none"><li>• By specifying the password as a TDBM entry by using the <b>userPassword</b> attribute in the <b>ldif2tdbm</b> load utility</li><li>• By using the <b>adminPW</b> keyword in the LDAP server configuration file (not suggested)</li></ul>	
LDAP fully qualified domain name	This is the IP address on which the LDAP server is listening, for example, for <code>ldap.widgets.com</code> . See Table 7 on page 19 for a definition of fully qualified domain name.	
LDAP port	This is the port for LDAP, for example, 389 in <code>ldap.widgets.com:389</code>	

Perform the following steps to update the z/OS HTTP Server configuration files (if you are configuring PKI Services for the first time):

1. If the second configuration file does not yet exist, create it by copying the first configuration file with the following command:

```
cp -p /etc/httpd.conf /etc/httpd1443.conf
```

2. Copy the first set of sample z/OS HTTP Server configuration directives (from the PKI Services samples directory, `/install-dir/samples/httpd.conf` file) into the default configuration file, `/etc/httpd.conf`.

**Note:** The *install-dir*, your file system installation directory, by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 9.)

- a. Copy the `keyfile`, `sslmode`, `sslport`, and `normalmode` directives as is, replacing any existing values.
- b. If your organization customized the value of `web_ring` (see Table 11 on page 29), change `SSLring` in the `keyfile` directive in the following line to the customized value:

```
keyfile SSLring SAF
```

- c. Optionally, copy the `userId` directive as is, replacing any existing value.

**Guideline:** Copy the `userId` directive (as shown in the following) into your file as is. However, if you already have a value in your file for this, you are not required to change it.

```
UserId %%CLIENT%%
```

## Updating z/OS HTTP Server configuration and starting the server

- d. Copy the protection and protect directives after any protection and protect directives you already have. Do not change the order in which these directives appear.
- e. Copy the redirect directives after any redirect directives you already have. Do not change the order in which these directives appear.
- f. Copy the pass and exec directives before any pass and exec directives you already have.
- g. Add the addtype directives to your list of addtypes if they don't already exist.
- h. Change all instances of *server-domain-name* to your Web server's fully qualified domain name, for example, `www.ibm.com`. (For information about your Web server's fully qualified domain name, see Table 7 on page 19.)
- i. Change all instances of *application-root* to your file system installation directory, which is `usr/lpp/pkiserv` by default.

**Note:** Your file system installation directory by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 9.)

- 
3. Copy the second set of z/OS HTTP Server configuration directives (from the PKI Services samples directory, `/install-dir/samples/httpd2.conf`) into the `/etc/httpd1443.conf` file.

**Note:** The *install-dir*, your file system installation directory, by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 9.)

- a. If you created this file by copying the first `httpd.conf` file, delete all existing protection, protect, redirect, pass, exec, and FastCGI directives.
- b. Copy the `userId`, `keyfile`, `sslmode`, `sslport`, `sslclientauth`, `normalmode`, and `SSLX500CARoots` directives as is, replacing any existing values.
- c. If your organization customized the value of `web_ring` (see Table 11 on page 29), change `SSLring` in the `keyfile` directive in the following line to the customized value:

`keyfile SSLring SAF`

- d. Add the following directives after the `SSLX500CARoots` directive:

- `SSLX500Host`
- `SSLX500Port`
- `SSLX500UserID`
- `SSLX500Password`

Replace the `<>` placeholders with the actual values from Table 22 on page 68.

- e. Copy the protection and protect directives after any protection and protect directives you already have. Do not change the order in which these directives appear.
- f. Copy the redirect directives after any redirect directives you already have. Do not change the order in which these directives appear.
- g. Copy the exec directives before any pass and exec directives you already have.

## Updating z/OS HTTP Server configuration and starting the server

- h. Change all instances of *server-domain-name* to your Web server's fully qualified domain name, for example, `www.ibm.com`. (For information about your Web server's fully qualified domain name, see Table 7 on page 19.)
- i. Change all instances of *application-root* to your file system installation directory.

**Note:** Your file system installation directory by default is `/usr/lpp/pkiserv`. The MVS programmer determines whether to change this default. (See Table 3 on page 9.)

- j. If you created `httpd1443.conf` by copying `httpd.conf`, optionally change the directories in `httpd1443.conf` for the report, log, and pid files.

**Guideline:** Execute this step to ensure the two servers are not using the same files at the same time.

- 1) Create a new directory for the `httpd1443` files by using the following command:  

```
mkdir /etc/internet/logs1443
```
- 2) Assign ownership to `WEBSRV` with the following command:  

```
chown websrv /etc/internet/logs1443
```
- 3) Edit the `*Log` directives in the new `httpd1443.conf` file to provide unique pathnames.

For example, if the first `httpd.conf` file has the following:

```
AccessLog      /etc/internet/logs/httpd-log
AgentLog       /etc/internet/logs/agent-log
RefererLog     /etc/internet/logs/referer-log
ErrorLog       /etc/internet/logs/httpd-errors
CgiErrorLog    /etc/internet/logs/cgi-errors
```

change the `httpd1443.conf` `*Logs` to the following:

```
AccessLog      /etc/internet/logs1443/httpd-log
AgentLog       /etc/internet/logs1443/agent-log
RefererLog     /etc/internet/logs1443/referer-log
ErrorLog       /etc/internet/logs1443/httpd-errors
CgiErrorLog    /etc/internet/logs1443/cgi-errors
```

---

## Steps for starting the z/OS HTTP Server

Perform the following steps to start the z/OS HTTP Server (if you are configuring PKI Services for the first time):

1. Make sure that the LDAP server is started. (If you are unsure about this, see "Steps for installing and configuring LDAP" on page 20.)

2. Execute the following commands from the UNIX command line:

```
httpd
httpd -r /etc/httpd1443.conf
```

Alternately, if you are using the `IMWEBSRV` started procedure as shipped with the Web server, you can start the two instances by entering the following MVS console commands:

```
S IMWEBSRV
S IMWEBSRV,ICSPARM='-r /etc/httpd1443.conf'
```

---

---

## Chapter 8. Tailoring the PKI Services configuration file for LDAP

You need to tailor the **LDAP** section of the `pkiserv.conf` configuration file only if you meet one of the following conditions:

- You are configuring PKI Services for the first time
- You intend to use encrypted passwords for your LDAP servers

Chapter 5, “Configuring the UNIX runtime environment,” on page 45 describes tasks the UNIX programmer performs. The other team members perform additional tasks before the UNIX programmer updates the **LDAP** section of the `pkiserv.conf` configuration file (described in this chapter) and starts the PKI Services daemon (described in Chapter 10, “Starting and stopping PKI Services,” on page 85).

---

### Excerpt of LDAP section

The following excerpt shows the **LDAP** section of the `pkiserv.conf` configuration file as it is shipped:

```
[LDAP]
NumServers=1
PostInterval=5m
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPw1=root
CreateOUValue= Created by PKI Services
RetryMissingSuffix=T
# Name of the LDAPBIND Class profile containing the bind information for LDAP
# server 1. This key is optional. Used in place of keys Server1, AuthName1.
# and AuthPw1
#BindProfile1=LOCALPKI.BINDINFO.LDAP1
```

You use the **LDAP** section of the `pkiserv.conf` file to provide information for one or more LDAP servers. The `NumServers` line specifies the number of servers.

---

### Storing information for encrypted passwords for your LDAP servers

You store information about passwords for binding to LDAP directories in the `pkiserv.conf` configuration file. Passwords can be in clear text or encrypted. By default, the `pkiserv.conf` configuration file contains `Server1`, `AuthName1`, and `AuthPw1` parameters; these lines are for specifying your LDAP bind information, including passwords, in clear text: (For more than one LDAP server, you add additional lines, `Server2`, `AuthName2`, `AuthPw2`, `Server3`, `AuthName3`, `AuthPw3`, and so forth.) If you want to use encrypted passwords for your LDAP servers, you delete all these lines, uncomment (remove the `#`) from the `BindProfile1` line at the bottom of the file, and correct the profile value specified if necessary. (See “Using encrypted passwords for LDAP servers” on page 260 for information about setting up this bind profile in RACF). For more than one LDAP server, you add additional lines: `BindProfile2`, `BindProfile3`, and so forth.

PKI Services performs the following processing when locating LDAP bind information:

1. The `Server $n$`  line specifies the fully qualified domain and port of your LDAP server. If your file contains a `Server $n$`  line, PKI Services looks for the matching `AuthName $n$`  and `AuthPw $n$`  lines and uses these values.

## Tailoring the PKI Services configuration file for LDAP

2. The `BindProfilen` parameter specifies the name of the LDAPBIND class profile. If your file does not contain a `Servern` line but does contain a `BindProfilen` line, PKI Services looks for the bind information in the LDAPBIND class profile. (If `Servern` is present, PKI Services does not look for bind information in `BindProfilen`, even if the value in `Servern` is incorrect.)
3. If neither is present for a specific server, then PKI Services uses the default from IRR.PROXY.DEFAULTS in the FACILITY class.

---

## Steps for tailoring the LDAP section of the configuration file

### Before you begin:

- **Important:** You need to update the **LDAP** section of the `pkiserv.conf` configuration file only if you are configuring PKI Services for the first time or your company is using encrypted passwords for your LDAP servers.
- You will need UNIX programming skills to complete this procedure.
- Table 23 lists some parameters that are in the **LDAP** section of the `pkiserv.conf` configuration file. The rightmost column lists the default values. You need to change some of these values. Fill in the blank lines with your company's information (and cross out these defaults). If you decide to change any of the other defaults, cross out these values and record your company's information.

Table 23. Information needed for updating the LDAP section of the configuration file

Parameter	Information needed	Where to get this information	Default value and your company's information
<code>NumServers=</code>	The number of available LDAP servers. These are replicas that can post certificates and CRLs.	From LDAP programmer	<b>1</b>
<code>PostInterval=</code>	<p>How often the posting thread will scan the request database for certificates and CRLs to post to the LDAP server in weeks (w), days (d), hours (h), minutes (m), or seconds (s).</p> <p><b>Note:</b> If the post is unsuccessful for a given certificate, the post is retried at the next post interval. If the post continues to be unsuccessful after 3 attempts, the post frequency for this certificate is reduced to no more than once per hour. After 26 unsuccessful attempts, it is further reduced to no more than once per day. After 33 unsuccessful attempts, the post request for this certificate is deleted from the request database.</p>	<p>UNIX programmer decides this. Specify a number followed by h (hours), m (minutes) or s (seconds).</p> <p>Example: 6m</p>	<b>5m</b>

## Tailoring the PKI Services configuration file for LDAP

Table 23. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
Server1=	<p>You use this parameter only if you are storing LDAP passwords in the clear.</p> <p>This parameter's value is the fully qualified domain name (domain name or IP address and port) for the first LDAP server.</p>	Copy this information from the earlier (completed) table, Table 20 on page 65.	<p><i>myldapserver.mycompany.com:389</i></p> <p><b>Note:</b> If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for <i>each</i> server.</p>
AuthName1=	<p>You use this parameter only if you are storing LDAP passwords in the clear.</p> <p>This parameter's value is the distinguished name to use for LDAP binding.</p> <p><b>Note:</b> You must specify the OID qualifiers in uppercase and without any spaces surrounding the equal signs or commas that separate the attribute value assertions (AVAs).</p> <p>(See Table 9 on page 21 for a definition of distinguished name.)</p> <p>The LDAP administrator defines the administrator's distinguished name with the <b>adminDN</b> keyword in the LDAP server configuration file. For example, the value is "CN=Admin" in the following:</p> <pre>adminDN "CN=Admin"</pre>	Copy this information from the earlier (completed) table, Table 20 on page 65.	<p><i>CN=root</i></p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for <i>each</i> server.</li> <li>The default name of the LDAP server configuration file is <i>slapd.conf</i> for the z/OS Integrated Security Services LDAP server.</li> </ul> <p><b>Note:</b></p>

## Tailoring the PKI Services configuration file for LDAP

Table 23. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
AuthPwd1=	<p>You use this parameter only if you are storing LDAP passwords in the clear.</p> <p>This parameter's value is the password to use for LDAP binding. The LDAP programmer sets this.</p> <p><b>Note:</b> Include this parameter, Server1, and AuthName1 only if you are storing the LDAP password in the clear. Alternately, if you encrypting the password for an LDAP server, use the BindProfile1 parameter. Omitting BindProfile1 and Server1 specifies using the PROXY segment information from the IRR.PROXY.DEFAULTS profile in the FACILITY class. (For more information, see "Using encrypted passwords for LDAP servers" on page 260.)</p>	Copy this information from the earlier (completed) table, Table 20 on page 65.	<p><i>root</i></p> <p><b>Note:</b> If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for <i>each</i> server.</p>
CreateOUValue=	<p>Value to use for the OU attribute when creating LDAP entries under the objectclass organizationalUnit. (See Table 82 on page 387.) This is used only when no OU value is specified in the relative distinguished name.</p>	UNIX programmer decides this (after consulting with LDAP programmer)	<b>Created by PKI Services</b>
RetryMissingSuffix=	<p>True (T) or False (F) setting that indicates whether LDAP post requests should be retried later if the distinguished name suffix does not exist. When set to F, LDAP post requests that fail because of a missing suffix are discarded.</p>	UNIX programmer decides this (after consulting with LDAP programmer)	<b>T</b>



## Tailoring the PKI Services configuration file for LDAP

Table 23. Information needed for updating the LDAP section of the configuration file (continued)

Parameter	Information needed	Where to get this information	Default value and your company's information
BindProfile1=	<p>You use this parameter only if you intend to use an encrypted password for your LDAP server.</p> <p>This parameter's value is the name of the LDAPBIND class profile containing the bind information for the LDAP server. (For more information, see "Using encrypted passwords for LDAP servers" on page 260.)</p>	Get the profile name from the RACF administrator who creates the profile. See "Using encrypted passwords for LDAP servers" on page 260 for more information.	<p>LOCALPKI.BINDINFO.LDAP1</p> <p><b>Note:</b> If the number of servers (the value in the row containing NumServers=) is greater than one, you need one value for <i>each</i> server.</p>

Perform the following steps to update the **LDAP** section of the pkiserv.conf configuration file (if you are configuring PKI Services for the first time or using encrypted passwords for your LDAP servers):

1. If necessary, change 1 (the default) in the following line to the number of available LDAP servers listed in Table 23 on page 72:

```
NumServers=1
```

2. Optionally change 5m in the following line to the posting interval in Table 23 on page 72:

```
PostInterval=5m
```

3. If necessary, update the BindProfile1 line or the Server1, AuthName1, and AuthPwd1 lines:
  - If you intend to use encrypted passwords for your LDAP servers and you are migrating from z/OS V1R3:

- If you intend to use an LDAPBIND class profile, perform the following steps:

- a. Copy the following lines in the sample configuration file (/usr/lpp/pkiserv/samples/pkiserv.conf) into your pkiserv.conf configuration file (at the bottom of the **LDAP** section).

```
# Name of the LDAPBIND Class profile containing the bind information for LDAP
# server 1. This key is optional. Used in place of keys Server1, AuthName1,
# and AuthPwd1
# BindProfile1=LOCALPKI.BINDINFO.LDAP1
```

- b. Remove the comment delimiter (#) from the start of the fourth line and change LOCALPKI.BINDINFO.LDAP1 to the name of the LDAPBIND class profile. (See Step 3 on page 261).

- c. Delete the following three lines in the **LDAP** section:

```
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPwd1=root
```

## Tailoring the PKI Services configuration file for LDAP

- If you intend to use the FACILITY class profile IRR.PROXY.DEFAULTS, delete the following three lines in the **LDAP** section:  

```
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPwd1=root
```
  - If you intend to use an encrypted password for your LDAP server and you are configuring PKI Services for the first time, perform the following steps:
    - a. If you are using an LDAPBIND class profile, remove the comment delimiter (#) from the start of the following line and change LOCALPKI.BINDINFO.LDAP1 to the name of the LDAPBIND class profile. (See Step 3 on page 261).  

```
# BindProfile1=LOCALPKI.BINDINFO.LDAP1
```
    - b. Delete the following three lines in the **LDAP** section:  

```
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPwd1=root
```
  - If you are not using an encrypted password for your LDAP server and are configuring PKI Services for the first time, perform the following steps:
    - a. Change *your-ldap-server-address:port* to your fully qualified domain name and port as listed in Table 23 on page 72:  

```
Server1=your-ldap-server-address:port
```
    - b. Change *CN=root* in the following line to the value of the administrator distinguished name in Table 23 on page 72:  

```
AuthName1=CN=root
```
    - c. Change *root* in the following line to the value of the administrator password in Table 23 on page 72:  

```
AuthPwd1=root
```
  - If you are not using encrypted passwords for your LDAP servers and you are migrating from z/OS V1R3, you do not need to do anything.
- 
4. If the value of NumServers= is greater than 1, repeat Step 3 on page 75 for each additional server. (You will need to increment the number in the parameter names for each additional server, for example Server2, AuthName2, AuthPwd2).
- 
5. If necessary, change Created by PKI Services in the following line to the OU attribute value in Table 23 on page 72:  

```
CreateOUValue=Created by PKI Services
```
- 
6. If necessary, change T in the following line to the RetryMissingSuffix value in Table 23 on page 72:  

```
RetryMissingSuffix=T
```
-

---

## Chapter 9. Creating VSAM data sets

This chapter includes the following procedures:

- “Planning VSAM storage requirements”
- “(Optional) preliminary steps for establishing VSAM RLS” on page 78
- “Steps for creating the VSAM object store and ICL data sets and indexes” on page 79
- “Steps for creating additional alternate indexes” on page 80
- “(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS” on page 81
- “(Optional) steps for adding VSAM buffer space” on page 82.

You need to perform the tasks this chapter if:

- You are configuring PKI Services for the first time or adding a new CA domain.
- You are migrating from a z/OS system running Version 1 Release 4 or lower that did not include support for additional alternate indexes. (You will need to execute the procedure in “Steps for creating additional alternate indexes” on page 80.)
- Your organization is using a sysplex for PKI Services daemons.
- You wish to tune VSAM performance.

The MVS programmer performs the following tasks:

- If configuring PKI Services for the first time, create the VSAM object store and ICL data sets and indexes.
- If using a sysplex, perform the preliminary steps for establishing VSAM record-level sharing (RLS).
- If using a sysplex and migrating from a z/OS system running Version 1 Release 4 or lower, enable existing PKI Services VSAM data sets for VSAM RLS.
- If migrating from a z/OS system running Version 1 Release 4 or lower, add additional VSAM alternate indexes.
- If desired, tune VSAM data set performance.

---

### Planning VSAM storage requirements

The MVS programmer uses the IKYCVSAM sample JCL to create two VSAM data sets (clusters):

- A data set for the request database (object store)
- A data set for the issued certificate list (ICL).

The MVS programmer also uses the same sample JCL to create five alternate index data sets (paths):

1. Transaction ID alternate index into the object store
2. Status alternate index into the object store
3. Requestor alternate index into the object store
4. Status alternate index into the ICL
5. Requestor alternate index into the ICL.

The IKYCVSAM sample JCL contains default values for the primary and secondary extent allocations for these data sets. The default allocation for base clusters is CYL(3,1). For alternate indexes, it is TRK(5,1). You need to update these values

## Creating VSAM data sets

based on your anticipated future needs. Use the following guidelines to update the space allocation parameters for the DEFINE CLUSTER and DEFINE ALTERNATEINDEX statements. (For more information about IDCAMS, see *z/OS DFSMS Access Method Services for Catalogs*.)

### Determining storage needs for ICL

The ICL maintains a permanent record for each certificate PKI Services issues. There is one ICL record for each issued certificate. Unless set up otherwise, the ICL will continuously grow over time as more certificates are issued. Assuming average-size certificates, one ICL record will occupy 1024 bytes of storage. With an allocation of CYL(3,1), the data set can have a maximum of 125 cylinders on a single 3390 volume for a total size of 105 MB. This would mean the data set should be able to hold at least 102500 certificates. If multiple volume support is used, you can double this amount.

If your anticipated needs differ greatly from the above value, you need to adjust the space allocation parameters CYL(3,1) on the DEFINE CLUSTER statement for the ICL. (This is the second DEFINE CLUSTER statement in IKYCVSAM. See “IKYCVSAM” on page 373 for a code sample of this file.) You may also want to proportionally adjust the space allocation parameters TRK(5,1) on the DEFINE ALTERNATEINDEX statements for the ICL. These are defined in the DEFALTDX job step. (Their names contain the icl qualifier.)

### Determining storage needs for the object store

The object store holds records to track active certificate requests. There is one object store record for each active certificate request and potentially another record to post the certificate to the LDAP directory. Object store records are not permanent. They are deleted when they are no longer needed. Unlike the ICL, the object store does not grow beyond a certain point, unless there is a sharp increase in certificate request activity. Assuming average-size certificate requests, one object store record and its companion posting record will occupy a total of 2560 bytes of storage. With a space allocation of CYL(3,1), the data set can have a maximum of 125 cylinders on a single 3390 volume for a total size of 105 MB. This would mean the data set should be able to hold at least 41000 concurrent certificate requests. If multiple volume support is used, you can double this amount.

If your anticipated needs different greatly from the preceding value, you need to adjust the space allocation parameters CYL(3,1) on the DEFINE CLUSTER statement for the object store. (This is the first DEFINE CLUSTER statement in IKYCVSAM. See “IKYCVSAM” on page 373 for a code sample of this file.) You may also want to proportionally adjust the space allocation parameters TRK(5,1) on the DEFINE ALTERNATEINDEX statements for the object store. These are defined in the DEFALTDX job step. (Their names contain the ost qualifier.)

---

## (Optional) preliminary steps for establishing VSAM RLS

Your team can configure PKI Services to take advantage of a Parallel Sysplex environment. This enables you to start multiple instances of the PKI Services daemon (one per image) that work in unison. The daemons are totally independent of each other, but they all act upon a single common data store containing the ICL and ObjectStore VSAM data sets.

If you want to run multiple instances of PKI Services in a Parallel Sysplex (one per image), you must first establish the data sharing environment suitable for VSAM record-level sharing (RLS).

**Before you begin:** The following steps assume that the coupling facility has already been set up. If this is not the case, for information on how to set up the coupling facility, see *z/OS MVS Programming: Sysplex Services Guide*. Also see “(Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS” on page 81 for additional information about setting up VSAM data sets to run PKI Services in a sysplex.

Perform the following steps to establish VSAM RLS. For specific information on how to perform these steps, see the chapter about administering VSAM record level sharing in *z/OS DFSMS Storage Administration Reference*.

1. Define and activate at least two sharing control data sets (SHCDS) and one spare SHCDS for recovery purposes.

2. Define CF lock structure to MVS.

3. Define CF lock structure in the SMS base configuration.

4. Define at least one storage class for VSAM record-level sharing (RLS).

You must record the name of this storage class for use in creating the VSAM data sets for PKI Services.

Table 24. VSAM RLS information you need to record

VSAM information you need to record	Your value
Name of storage class for VSAM RLS	

## Steps for creating the VSAM object store and ICL data sets and indexes

You need to perform this task only if you are configuring PKI Services for the first time.

PKI Services uses VSAM data sets to store requests in progress and issued certificates. You need to create these data sets manually.

**Before you begin:** If you also want run multiple instances of PKI Services in a Parallel Sysplex (one per image), you need to have performed the steps described in “(Optional) preliminary steps for establishing VSAM RLS” on page 78.

Perform the following steps to create the VSAM object store and ICL data sets and indexes (if you are configuring PKI Services for the first time):

1. Copy the sample JCL in SYS1.SAMPLIB(IKYCVSAM) to your JCL data set. (See “IKYCVSAM” on page 373 for a code sample of this file.)
2. Update your data set as directed in the instructions in the prolog of the sample JCL:
  - a. Change the JOB card.
  - b. Change the VOL statements.

## Creating VSAM data sets

- If you are running multiple instances of PKI Services in a Parallel Sysplex, replace the VOL statements with STORCLAS statements that specify the storage class recorded in Table 24 on page 79, for example:  
STORCLAS(VSAMRLS)
- If you are running without a Parallel Sysplex, replace the vvvvvv in the VOL statements with a VOL=SER suitable for your VSAM data sets.
- c. If you are running multiple instances of PKI Services in a Parallel Sysplex, remove the SPANNED and CISIZE statements in the file. These lines follow:  
SPANNED -  
CISZ(512) -
- d. You can optionally change the data set names but must remember to make equivalent changes in the pkiserv.conf file if you do so. (See Step 1b on page 60.)
- e. Update the primary and secondary extent allocations based on your anticipated future needs. (See “Planning VSAM storage requirements” on page 77 for guidelines on determining the space you will need.) These are the default allocations for each type of data set:  

<b>Base cluster</b>	CYL(3,1)
<b>Alternate indexes</b>	TRK(5,1)

**Guideline:** Do not change any numeric values, other than the primary and secondary space allocation values for the base cluster and alternate index data sets.

- 
3. Submit the job when your changes are complete.
- 

## Steps for creating additional alternate indexes

You need to perform this task only if you are migrating from a system running z/OS Version 1 Release 4 or lower that did not include support for additional alternate indexes.

PKI Services uses these VSAM alternate indexes to access the ObjectStore and ICL data sets. You must allocate these alternate index data sets manually.

**Before you begin:** If you also want run multiple instances of PKI Services in a Parallel Sysplex (one per image), you need to have performed the steps described in “(Optional) preliminary steps for establishing VSAM RLS” on page 78.

Perform the following steps to create additional alternate indexes for the VSAM object store and ICL data sets only if you are migrating from a system running z/OS Version 1 Release 4 or lower that did not include support for additional alternate indexes.

1. Copy the sample JCL in SYS1.SAMPLIB(IKYMVSAM) to your JCL data set. (See “IKYMVSAM” on page 377 for a code sample of this file.)
2. Update your data set as directed in the instructions in the prolog of the sample JCL:
  - a. Change the JOB card.
  - b. Change the VOL statements.

- If you are using VSAM record level sharing, remove the VOL statements.
  - If you are not using VSAM record level sharing, replace the vvvvvv in the VOL statements with a VOL=SER suitable for your VSAM data sets.
- c. You can optionally change the data set names but must remember to make equivalent changes in the pkiserv.conf file if you do so. (See Step 1b on page 60.)
  - d. Update the primary and secondary extent track allocations based on your anticipated future needs. (See “Planning VSAM storage requirements” on page 77 for guidelines on determining the space you will need.) These are the default allocations on the DEFINE ALTERNATEINDEX statements for alternate indexes is TRK(5,1).

**Guideline:** Do not change any numeric values, other than the primary and secondary space allocation values for the alternate index data sets.

- 
3. Submit the job when your changes are complete.
- 

### (Optional) steps for enabling existing PKI Services VSAM data sets for VSAM RLS

To run PKI Services in parallel, the UNIX programmer must specify SharedVSAM=T in the pkiserv.conf configuration file. (See the SharedVSAM row in Table 19 on page 51.) The MVS programmer enables the sysplex to access the VSAM data sets.

**Before you begin:** You need to have performed the steps described in “(Optional) preliminary steps for establishing VSAM RLS” on page 78.

Perform the following steps to enable your existing PKI Services data sets for VSAM record-level sharing (RLS).

1. Copy the sample reallocation JCL in SYS1.SAMPLIB(IKYRVSAM) to your JCL data set.

**Attention:** Do not run IKYCVSAM by mistake because this JCL will destroy your existing VSAM data sets.

- 
2. Update your data set, following the instructions in the prolog of the sample JCL:

- a. Change the JOB card.
- b. Change the STORCLAS statements.

**Note:** Before submitting this job, check your access control service (ACS) routines for naming convention definitions that will be used to create the VSAM RLS data sets.

- c. Rename the source data sets to the names of your existing ObjectStore and ICL data sets.
- d. Change the destination data set names.

**Note:** Remember to give the UNIX programmer the data set names so the UNIX programmer can make equivalent changes in the pkiserv.conf file. See “(Optional) Steps for updating the configuration file” on page 51.



## Creating VSAM data sets

- e. Update the primary and secondary extent allocations based on your anticipated future needs. (See “Planning VSAM storage requirements” on page 77 for guidelines on determining the space you will need.) These are the default allocations for each type of data set:

**Base cluster** CYL(3,1)

**Alternate indexes** TRK(5,1)

**Guideline:** Do not change any numeric values, other than the primary and secondary space allocation values for the base cluster and alternate index data sets.

3. Submit the job when your changes are complete.

---

## Tuning VSAM performance

Depending on your environment, your VSAM performance may be improved by providing buffer space for the VSAM data sets as part of the IKYSPROC (alias PKISERVD) started procedure.

### (Optional) steps for adding VSAM buffer space

Perform the following steps to add buffer space for the VSAM data sets as part of the IKYSPROC (alias PKISERVD) started procedure. When completed, you will need to stop and restart PKI Services before your changes will take effect.

1. Make a backup copy of SYS1.PROCLIB(PKISERVD). (See “PKISERVD sample procedure to start PKI Services daemon” on page 384 for a code sample of this file.)
2. Edit the JCL in SYS1.PROCLIB(PKISERVD) in the following ways.
  - a. Append the DD statements shown in the following example to the bottom of the PKISERVD procedure.
  - b. Change the names of the VSAM data sets shown in the example to the names that you used when you executed the IKYCVSAM sample job to allocate the data sets. (See “Steps for creating the VSAM object store and ICL data sets and indexes” on page 79.)

**Example:**

```
/* ObjectDSN data set
//OST DD DSN=PKISRVD.VSAM.OST,DISP=SHR,
// AMP=('BUFNI=6,BUFND=4')
/* ObjectTidDSN data
//TID DD DSN=PKISRVD.VSAM.OST.PATH,DISP=SHR,
// AMP=('BUFNI=6,BUFND=4')
/* ObjectStatusDSN data set
//OSTAT DD DSN=PKISRVD.VSAM.OST.STATUS,DISP=SHR,
// AMP=('BUFNI=1,BUFND=4')
/* ObjectRequestorDSN data set
//OREQ DD DSN=PKISRVD.VSAM.OST.REQUESTR,DISP=SHR,
// AMP=('BUFNI=1,BUFND=4')
/* ICLDSN data set
//ICL DD DSN=PKISRVD.VSAM.ICL,DISP=SHR,
// AMP=('BUFNI=6,BUFND=4')
/* ICLStatusDSN data set
//ISTAT DD DSN=PKISRVD.VSAM.ICL.STATUS,DISP=SHR,
```



```
// AMP=('BUFNI=1,BUFND=4')
/* ICLRequestorDSN data set
//IREQ DD DSN=PKISRVDS.VSAM.ICL.REQUESTR,DISP=SHR,
// AMP=('BUFNI=1,BUFND=4')
```

3. Edit the names in the new DD statements above to match the data sets in the **ObjectStore** section of the PKI Services configuration file (pkiserv.config).

4. Optionally, you may need to adjust the numeric values for BUFNI and BUFND.  
**Tip:** The STATUS and REQUESTR data sets are accessed sequentially only, while the others are accessed both sequentially and directly. Keep this in mind when adjusting the values. (For more information on VSAM buffer space, see *z/OS DFSMS Using Data Sets*.)

5. Edit the **ObjectStore** section of the PKI Services configuration file (pkiserv.config) to change the existing data set names to the following DD names:

Default value	Suggested value
ObjectDSN='pkisrvd.vsam.ost'	ObjectDSN=DD:OST
ObjectTidDSN='pkisrvd.vsam.ost.path'	ObjectTidDSN=DD:TID
ObjectStatusDSN='pkisrvd.vsam.ost.status'	ObjectStatusDSN=DD:OSTAT
ObjectRequestorDSN='pkisrvd.vsam.ost.requestr'	ObjectRequestorDSN=DD:OREQ
ICLDSN='pkisrvd.vsam.icl'	ICLDSN=DD:ICL
ICLStatusDSN='pkisrvd.vsam.icl.status'	ICLStatusDSN=DD:ISTAT
ICLRequestorDSN='pkisrvd.vsam.icl.requestr'	ICLRequestorDSN=DD:IREQ

6. Save your changes.
7. Stop and restart PKI Services.



---

## Chapter 10. Starting and stopping PKI Services

You start the PKI Services daemon or daemons the first time you are configuring PKI Services or if you are adding sysplex support to run multiple independent instances of PKI Services (one per image) on a sysplex. The MVS programmer performs these tasks.

---

### Steps for starting the PKI Services daemon

You need to start the PKI Services daemon if:

- You are configuring PKI Services for the first time.
- You want to enable Simple Certificate Enrollment Protocol (SCEP).
- You renewed your CA or RA certificate.
- You want to use Parallel Sysplex support and need to run another instance of the PKI Services on a different image in the sysplex.
- You stopped PKI Services and need to restart it.
- You created a new (additional) CA domain and want to start it.

#### Before you begin:

- Your z/OS HTTP Server should be SSL-enabled (see Chapter 7, “Updating z/OS HTTP Server configuration and starting the server,” on page 67) and the uncustomized PKISERV application ready for use.
- If you are starting PKI Services for the first time, you need to know the runtime directory, called *runtime-dir* in the command examples. The default is */etc/pkiserv/*. The MVS programmer was asked to record any changes to the default; see Table 3 on page 9.
- If you are starting PKI Services for a new CA domain, you need to know the job name that contains the instance of the daemon you need to start. Do not use these steps. Instead see “Adding a new CA domain” on page 161 for steps to add a new domain and start the new daemon.

Perform the following steps to start the PKI Services daemon and view your Web pages:

1. If you have not done so already, start the Web server and the LDAP server.
2. If you want to test the configuration to this point before customizing PKI Services (preferred), you need to temporarily prevent PKI Services from posting issued certificates to LDAP because posting to LDAP will not be successful. Have the UNIX programmer perform the following steps to prevent PKI Services from posting issued certificates to LDAP:
  - a. Edit the PKI Services configuration file (by default, this is: */etc/pkiserv/pkiserv.conf*).
  - b. Set NumServers=0 in the **LDAP** section of the file.
  - c. Exit to save your changes.

**Note:** After testing the configuration, you need to stop PKI Services and undo the change in this step (see Step 2 on page 87) and then restart PKI Services.

## Starting and stopping PKI Services

3. Start the PKI Services daemon from the MVS console by entering the following command:

```
S PKISERVD
```

**Notes:**

- a. If you are migrating from z/OS V1R3 and adding Parallel Sysplex support (you want to start multiple instances of the PKI Services daemon on different images in the sysplex), the first time you start the daemon, you must ensure that the system you start first is the one that has access to the old work files (by default, /var/pkiserv). (These are on the same system you were using for PKI Services before adding sysplex support).
- b. You must start the PKI Services daemon only from a started procedure. PKI Services rejects all other methods of starting the daemon (including INETD, /etc/rc, UNIX shell, or submitted JCL job).
- c. Depending on the amount of customization you did, there are various versions of the preceding command to start the PKI Services daemon. For example, if you changed the pkiserv.envars file (see Step 3 on page 47), you need to specify its new location as a parameter in the START command:

```
S PKISERVD,DIR='runtime-dir'
```

(Single quotation marks are required to maintain the character case of the values being assigned to the substitution parameters.)

The command in the following example specifies the runtime directory and the file name of the environment variables file:

**Example:**

```
S PKISERVD,DIR='/etc/pkiserv',FN='pkiserv.envars'
```

The default time zone is EST5EDT. If you need to change this, you can supply the new value as a parameter, as in the following examples:

**Examples:**

```
S PKISERVD,TZ=PST8PDT
```

```
S PKISERVD,JOBNAME=jobname,DIR='/etc/pkiserv',FN='pkiserv.envars',TZ=PST8PDT
```

4. Go to your Web pages by entering the following URL from your browser:

```
http://webserver-fully-qualified-domain-name/PKIServ/public-cgi/camain.rexx
```

The *webserver-fully-qualified-domain-name* is the common name (CN) portion of the Web server's distinguished name; see Table 11 on page 29.

You should be able to go through your Web pages to request, retrieve, and revoke a certificate of type "PKI browser certificate for authenticating to z/OS". Ensure you can do this before trying to customize the application.

5. If you elected to test the configuration, you need to stop PKI Services (see "Stopping the PKI Services daemon"), undo the change in Step 2 on page 85 (see Step 2 on page 87 for steps on undoing the change), and then restart PKI Services.

---

## Stopping the PKI Services daemon

Perform the following steps to stop the PKI Services daemon:

1. To stop the PKI Services daemon, enter one of the following two commands  
You can use either the following MODIFY (or **F**) console command:

```
F PKISERVD,STOP
```

or the STOP (or **P**) command:

```
P PKISERVD
```

---

2. If you changed the PKI Services configuration file (as previously suggested—see Step 2 on page 85), have the UNIX programmer undo that change now by performing the following steps:
  - a. Edit the PKI Services configuration file (by default `/etc/pkiserv/pkiserv.conf`).
  - b. Set `NumServers=n` in the **LDAP** section of the file, where *n* is the same number of LDAP servers indicated in Table 23 on page 72.
  - c. Exit to save your changes.

## Starting and stopping PKI Services

---

## Part 3. Customizing PKI Services

This part includes the following:

- Chapter 11, “Customizing the end-user Web application,” on page 91 provides an overview of the `pkiserv.tpl` file, which contains the certificate templates, and explains customizing the end-user Web pages.
- Chapter 12, “Customizing the administration Web pages,” on page 141 provides an overview of the CGI scripts and explains how to customize the administration Web pages.
- Chapter 13, “Advanced customization,” on page 145 explains:
  - “Scaling for high volume installations” on page 145
  - “Using certificate policies” on page 146
  - “Updating the signature algorithm” on page 149
  - “Customizing distribution point CRLs” on page 150
  - “Creating a distribution point ARL” on page 156
  - “Adding an application domain” on page 158
  - “Adding a new CA domain” on page 161
  - “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175
  - “Using the PKI exit” on page 180





---

## Chapter 11. Customizing the end-user Web application

For certificate processing to work, you need to customize the end-user Web pages at least to some degree. Before you begin to customize Web pages, you need to understand the `pkiserv.tmpl` certificate templates file. This file contains certificate templates, which define the fields that comprise a specific certificate request. This chapter describes the `pkiserv.tmpl` certificate templates file and explains how to use it to customize the end-user Web pages. This chapter also explains the relationship between CGIs and the certificate templates file. Finally, this chapter also discusses customizing e-mail notifications. (Sending e-mail notifications is an optional feature.)

---

### Contents of the `pkiserv.tmpl` certificates templates file

The `pkiserv.tmpl` certificate templates file contains certificate templates that define the fields that comprise a specific certificate request. The file contains a mixture of true HTML and HTML-like tags. The HTML can contain JavaScript™ for input field verification.

The main sections of the `pkiserv.tmpl` certificate templates file are listed in Table 25:

*Table 25. Structure and main divisions of the certificate template file (`pkiserv.tmpl`)*

---

A prolog section of comments explaining main sections, subsections, named fields, and substitution variables.

---

A DEBUG flag appears right after the prolog section. (You can change `DEBUG=0` to `DEBUG=1` to get CGI debugging information.)

---

#### **APPLICATION sections**

The APPLICATION sections contain subsections, which produce certain Web pages, such as the PKI Services home page shown in Figure 6 on page 199. For details, see “The APPLICATION sections” on page 99.

---

#### **TEMPLATE sections**

These are the certificate templates (models) that contain the HTML to produce certificate request forms. They also define the fields that are permissible in the certificate. For details, see “TEMPLATE sections” on page 103.

---

#### **INSERT sections**

These contain HTML for certain Web pages, such as the “Request submitted successfully” Web page, and certificate field dialogs, such as text entry boxes (the common name INSERT produces a text box where the user enters this information) and drop-downs. For details, see Figure 10 on page 207.

---

The `pkiserv.tmpl` file begins with a prolog. This is a section of comments that explains the main sections and subsections of the file. Any line with a `#` in column 1 is a comment.

Only the APPLICATION sections and TEMPLATE sections can contain subsections, but all three can contain named fields and substitution variables.

### What are substitution variables?

A substitution variable holds a value that HTML code can reference. At run time, the actual value replaces a substitution variable.

## Customizing the end-user Web pages

You use square brackets to delineate a substitution variable.

### Example:

[base64cert]

### Notes:

1. Substitution variables are case-sensitive.
2. Depending on the section where a substitution variable is present, it may not have a valid meaning. For example, the [base64cert] substitution variable is meaningless before the certificate is retrieved. Therefore, in this case, the value of [base64cert] would be the null string (an empty string).

Table 26 summarizes valid substitution variables:

Table 26. Substitution variables

Substitution variable	Description
altrawvalue	The concatenated value of the AltOther fields.
base64cert	The requested certificate, base64-encoded.
browsertype	<p>A special substitution variable to qualify named fields only. It enables the different browsers, Netscape and Internet Explorer, to perform browser-specific operations, such as generating a public and private key pair. To do this, Netscape uses a KEYGEN HTML tag while Internet Explorer uses ActiveX controls.</p> <p>For example, suppose you specify %%PublicKey[browsertype]%% in a TEMPLATE CONTENT section. If the user referencing this section uses the Netscape Navigator browser, then INSERT PublicKeyNS is included. If the user's browser is Microsoft Internet Explorer, INSERT PublicKeyIE is included.</p>
cadomain	The CA domain name used to examine the preregistration record in the Simple Certificate Enrollment Protocol (SCEP).
errorinfo	Information such as the return code and reason code related to a failing SAF call.
iecert	The requested certificate in a form that Microsoft Internet Explorer accepts.
optfield	A special substitution variable that should be placed in any certificate field name INSERT where the end user can supply the value. It makes the input field optional.
printablecert	This contains the certificate details so that the end user can confirm that the certificate is the correct one to renew or revoke. The displayed data is extracted from the ICL entry.
readonly	An attribute of the input field in the AltOther_OID INSERT. It is converted to a null string when the INSERT is used for input purposes, such as when requesting a certificate. It is substituted with the string readonly when the INSERT is used for output purposes, such as when displaying request or certificate information.
tplname	A certificate template name. This is primed from the HTML tag <SELECT NAME="Template"> in the <APPLICATION NAME=PKISERV> section. The end user selects it on the first Web page.
transactionid	A unique value returned from a certificate request.

## What are named fields?

Named fields insert common HTML code, such as a common input field or a page header or footer, in a Web page. (Each named field refers to a corresponding INSERT section.) A named field is delineated with %%.

### Examples:

```
%%Country%%
%%-pagefooter%%
```

**Note:** Named fields are case-sensitive.

A named field can include or not include a dash. A named field without a dash, such as %%Label%% may have a special meaning as a certificate field. Its special meaning depends on the section in which it appears. (See “Relationship between CGIs and the pkiserv.tmpl file” on page 123 for more information.)

A named field with a dash, such as %%-pagefooter%%, has no special meaning. PKISERV treats it simply as HTML code to insert. Any special meaning the named field might have, based on the section in which it is contained, is ignored. For example, in a TEMPLATE CONTENT section (see “TEMPLATE sections” on page 103) if you specify %%-pagefooter%%, -pagefooter is not considered a certificate field name. However, the INSERT section with the name -pagefooter is included in the HTML page displayed to the end user.

## INSERT sections

Although the INSERT sections are at the end of the pkiserv.tmpl certificate templates file, they are explained first because of their relationship to named fields. As previously indicated, any named field used in the pkiserv.tmpl file must be defined in a corresponding INSERT section.

Unlike the APPLICATION sections and TEMPLATE sections, INSERT sections can have no subsections. The following is the format of an INSERT section:

**<INSERT NAME=*insert-name*>...</INSERT>**

An INSERT contains HTML that either:

- defines a certificate field
- defines other common HTML that can be referenced in other sections.

The following example of an INSERT defines a certificate field.

### Example:

```
<INSERT NAME=Country>
<p> Country [optfield] <BR>
<INPUT NAME="Country" TYPE="text" SIZE=2 maxLength="2">
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidCountry(frm){
  if ("[optfield]" == "" && frm.Country.value == "") {
    alert("Enter required field."); frm.Country.focus();
    return false;
  }
  return true;
}
//-->
</SCRIPT>
</INSERT>
```

The next example defines other common HTML:

### Example:

```
<INSERT NAME=-pagefooter>  
<p>email: webmaster@your_company.com  
</INSERT>
```

To reference an INSERT, you use a named field of the form `%%insert-name%%`, for example `%%Country%%` or `%%-pagefooter%%`.

The `pkiserv.tmp1` certificate templates file contains INSERT sections of several main types:

- Sample INSERTs, which are includable code inserts. (This is common HTML for Web page content as listed in Table 27.)
- Certificate fields that are defined in INSERT sections. (See Table 28 on page 95.) These include:
  - X.509 fields (for example, `OrgUnit`)
  - Non-X.509 fields (for example `UserId`).

Table 27. Sample INSERTs

INSERT NAME	Contents
-AdditionalHeadIE	ActiveX controls to enable Internet Explorer to generate a key pair
-requestok	HTML for the Web page “Request submitted successfully” after a successful certificate request (for both original requests and renewals). (For a sample of this Web page, see Figure 10 on page 207.)
-requestbad	HTML for the Web page “Request was not successful”
-renewrevokeok	HTML for the Web page “Request submitted successfully” after a successful attempt to revoke a certificate. (See Figure 15 on page 212 for a sample of the Web page to renew or revoke a certificate.)
-renewrevokebad	HTML for the Web page “Request was not successful” after an unsuccessful attempt to renew or revoke a certificate. (See Figure 15 on page 212 for a sample of the Web page to renew or revoke a certificate.)
-preregok	HTML for the Web page “Preregistration successful” after a successful attempt to preregister a client for a certificate.
-return10cert	This returns a #10 certificate.
returnbrowsercertNS	This contains <code>[base64cert]</code> , which is the base64 substitution variable.
returnbrowsercertIE	This contains a script for producing a popup window installing your certificate (if you are using the Microsoft Internet Explorer browser). See Figure 12 on page 209 for a sample of this Web page.

### Named fields in INSERT sections

Most of the following fields are X.509 fields. Table 28 on page 95 summarizes the named fields in INSERT sections. (See **Restrictions** at the end of the table.)

Table 28. Named fields in INSERT sections

Field	Description
AltDomain	<p>The host name of the machine where a certificate will be installed. This is a text field of up to 100 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
AltEmail	<p>The user's e-mail address, including the @ character and any periods (.). This is a text field of up to 100 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
AltIPAddr	<p>The unique IP version 4 address that specifies the location of the server or device on the Internet, for example, 9.67.97.103. (PKI Services supports only IP version 4 addresses.) The IP address is in dotted decimal format and is a text field of up to 15 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
AltOther <sup>1</sup>	<p>A free form value for the other name of the subject's alternate name. Unlike the other INSERTs, you must customize it before you use it. The name of this INSERT consists of the string AltOther, concatenated with an underscore (_), then followed by the OID you specify it in the format of the following sample: AltOther_1_2_3_4_5. (See "Customizing the OtherName field" on page 137.)</p> <p>You may have more than one input field but the total length of these fields together with the length of the OID and the comma can not exceed 255 bytes. The resulting AltOther field is built by concatenating the dotted-decimal OID that matches the INSERT name, a comma, and the value of the input field. This is a text field of up to 255 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
AltURI	<p>A name or address referring to an Internet resource; a URL is one kind of uniform resource identifier. This is a text field of up to 100 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
ChallengePassPhrase <sup>1</sup>	<p>The passphrase the user entered when requesting a certificate. The user types the same passphrase, exactly as entered on the request form. This is a case-sensitive text field of up to 32 characters.</p>
ClientName <sup>1</sup>	<p>Name of the person or device being preregistered. This is a text field of up to 64 characters.</p> <p><b>Restriction:</b> The first 32 characters of the name must be unique, irrespective of case, for each preregistered user.</p>
CommonName	<p>For browser certificates, this is your name, such as John Smith. (You can use your first and last name, in that order.) For server certificates, this is name by which the server's administrator wants it to be known. For SSL servers, the SSL protocol requires the CommonName to be the fully qualified domain name of the server, for example, www.ibm.com. CommonName is a text field of up to 64 characters.</p> <p>Although CommonName is a constant, no value is assigned to it. This indicates that RACF must determine the value. The user authenticates by specifying a user ID and password. (If UserId is listed in the APPL section, this means the application provides the user ID and password.) Providing the user ID and password enables RACF to look up the CommonName value in the user's profile.</p> <p><b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>
Country	<p>The country where your organization is located. This is a 2-character text field.</p> <p><b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.</p>

## Customizing the end-user Web pages

Table 28. Named fields in INSERT sections (continued)

Field	Description												
Email <sup>1</sup>	This is a deprecated insert for the e-mail address for the distinguished name; use the Mail insert instead. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.												
EmailAddr <sup>1</sup>	The e-mail address for the distinguished name. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.												
ExtKeyUsage <sup>1</sup>	The intended purpose of the certificate. Possible values are: <table> <tr> <td><b>clientauth</b></td><td>Client side authentication</td></tr> <tr> <td><b>codesigning</b></td><td>Code signing</td></tr> <tr> <td><b>emailprotection</b></td><td>Email protection</td></tr> <tr> <td><b>ocspsigning</b></td><td>OCSP response signing</td></tr> <tr> <td><b>serverauth</b></td><td>Server side authentication</td></tr> <tr> <td><b>timestamping</b></td><td>Digital timestamping.</td></tr> </table>	<b>clientauth</b>	Client side authentication	<b>codesigning</b>	Code signing	<b>emailprotection</b>	Email protection	<b>ocspsigning</b>	OCSP response signing	<b>serverauth</b>	Server side authentication	<b>timestamping</b>	Digital timestamping.
<b>clientauth</b>	Client side authentication												
<b>codesigning</b>	Code signing												
<b>emailprotection</b>	Email protection												
<b>ocspsigning</b>	OCSP response signing												
<b>serverauth</b>	Server side authentication												
<b>timestamping</b>	Digital timestamping.												
HostIdMap <sup>1</sup>	This is the user ID for authorization purposes, in an e-mail type of format: <i>subject-id@host-name</i>  For example, this could be dsmith@ibm.com. This is a text field of up to 100 characters.  There are three ways to use %%HostIdMap%: <ul style="list-style-type: none"> <li>• If you place it in the CONTENT section, the end user can specify the value (or values since it may be repeated).</li> <li>• You can also place it in the APPL section that the application provides. If you do so, it should have the following form:  %%HostIdMap=@host-name%%  The host-name is the hardcoded system name for the current system. The application provides the user ID as the user entered it when prompted for user ID and password. Note that, for this to function properly, the z/OS HTTP Server protection scheme for the request must force a prompt for user ID and password. Thus, only one HostIdMap is provided using this method.</li> <li>• A third way to specify HostIdMap is to place %%HostIdMap%% in the ADMINAPPROVE section. This allows the administrator to fill in the value when approving the certificate request. See "Administering HostIdMappings extensions" on page 242 for more information.</li> </ul>												
KeyProt <sup>1</sup>	(This is for the Internet Explorer browser only.) This asks if the user wants to enable strong private key protection. The drop-down choices are Yes and No.												

Table 28. Named fields in INSERT sections (continued)

Field	Description		
KeyUsage	The intended purpose of the certificate. Each possible value is shown here with its intended purpose and possible PKIX bits:		
	<b>KeyUsage value</b>	<b>Intended purpose</b>	<b>PKIX bits</b>
	<b>certsign</b>	Certificate and CRL signing	KeyCertSign and cRLSign
	<b>crlsign</b>	CRL signing	cRLSign
	<b>dataencrypt, dataencipherment, or dataenciph</b>	Data encryption	dataEncipherment
	<b>digitalsig or digitalsignature</b>	Authentication	digitalSignature
	<b>docsign or nonrepudiation</b>	Document signing	nonRepudiation
	<b>handshake</b>	Protocol handshaking (for example, SSL)	digitalSignature and keyEncipherment
	<b>keyagree or keyagreement</b>	Key agreement	keyAgreement
	<b>keycertsign</b>	Certificate signing	keyCertSign
	<b>keyencrypt, keyencipherment, or keyenciph</b>	Key transport	keyEncipherment
Label <sup>2</sup>	The label assigned to the requested certificate. This is a text field of up to 32 characters.		
Locality	The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.		
Mail <sup>1</sup>	The e-mail address for the distinguished name. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.		
NotBefore	Number of days (0–30) before the certificate becomes valid.		
NotAfter	Number of days (365–720) that the certificate is current. This is a period of time from 365 days (1 year) to 720 days (2 years).		
NotifyEmail <sup>1</sup>	The e-mail address for notification purposes. This is a text field of up to 64 characters. <b>Note:</b> When a certificate is created and posted to LDAP, the NotifyEmail value, if specified, is posted as the MAIL attribute. If the MAIL attribute already exists in that directory entry, its value is replaced by the new value. If both NotifyEmail and Email appear on one request, they must have the same value.		
Org	Organization. The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.		
OrgUnit	The name of your division or department. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.		
OrgUnit2	The name of your division or department. (There can be more than one organizational unit field on a request form. For example, one could be for your department and another for your division.) This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.		



## Customizing the end-user Web pages

Table 28. Named fields in INSERT sections (continued)

Field	Description
PassPhrase <sup>1</sup>	The user decides this and enters and then reenters it when requesting a certificate (and must later supply this value when retrieving the certificate). This is a case-sensitive text field of up to 32 characters. There is no minimum number of characters, and the user can use any characters, but alphanumeric characters (A–Z, a–z, and 0–9) are suggested.
PostalCode <sup>1</sup>	The zip code or postal code. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
PublicKey	The base64-encoded #10 certificate request. (This is for server or device enrollment only.) You create a certificate request on behalf of another server (which could be a z/OS server or other type of server) or device for which you are requesting a certificate. You use software specific to that server to generate the #10 request before going to the PKI Services Web site. Save the request in a file. Then open the file in a text editor such as Windows Notepad and copy the and paste the contents into the text box on the enrollment form. A text area of 70 columns and 12 rows is allocated for this certificate request. Here is an example of the certificate request:  -----BEGIN NEW CERTIFICATE REQUEST----- MIIBiDCB8gIBADAZMRcwFQYDVQDEw5Kb2huIFB1YmxpYzCBnzANBgkqhkiG 9w0BAQEFAAOBjQAwgYkCgYEASt1cJHAGPqi60jAyL+xNbt8z5ngmvq02V003oYu /mEnQtRM96e+2jbmDCRo5tWVklG40Yf9ZVB5biURMJFLztfa4AVdEVtun8DH2pwc wiNIZZcC1Zym5adurUmyDk64PgiiIPMQS/t0ttG4c5U8uWSK0b1J4V4f7ps+t1aG t+cCAwEAAaAwMC4GCSqGSIb3DQEJDDjEhMB8wHQYDVr00BBYEFAlKTovBBvnFqDA0 1oIhtRinwRC9MA0GCSqGSIb3DQEBBQUAA4GBAIBCVpwYvppIX3HHmpKZPNY8Snsz AJrDsgAEH51W0IRGywhqKcLLxa9htoQai6cdc8RpFVtwk6UfdCOGxMn4aFb34Tk3 5WYdz0iHXg8MhHiB3EruwdWs+S7Fv3JhU3FLwU61FLfAjbVi+35iEWQymOR6mE5W CathprmGfKRsdE5E -----END NEW CERTIFICATE REQUEST-----
PublicKeyIE <sup>1</sup>	(This is for the Internet Explorer browser only.) This is the cryptographic service provider. The user selects a value from a drop-down list (Microsoft Base Cryptographic Provider or Microsoft Enhanced Cryptographic Provider).
PublicKeyNS <sup>1</sup>	(This is for the Netscape browser only.) This is the key size for your public/private key pair. The user selects a value from the drop-down list. Larger keys are more secure, but they also increase the time needed for connecting to a secure session.
Requestor <sup>1</sup>	The user's name, used for tracking the request. This can be in any format, for example, John Smith or John. J. Smith. (This can differ from the common name, especially if the request is for a server certificate.) The value is saved with the request and issued certificate, but it is not a field in the created certificate. The default value is taken from the leftmost RDN in the subject's distinguished name, truncated to 32 characters.
SerialNumber <sup>1</sup>	Serial number of the subject device. This is a text field of up to 64 characters.
SignWith	For PKI the component and for SAF the component and key-label used to sign this certificate, indicating the provider for certificate generation. This is a text field of up to 45 characters. It can be SAF or PKI Services, as shown in the following examples.  <b>Examples:</b>  "SAF:CERTAUTH/Local CA Cert" "PKI:"  For SAF, the label of the signing certificate must be included. The first example shows the SignWith field in a SAF template. It includes the signing certificate, a CERTAUTH certificate labeled 'Local CA Cert'.  For PKI, it is an error to include the signing certificate. The second example shows the SignWith field in a PKI template. Notice that this contains no signing certificate.



Table 28. Named fields in INSERT sections (continued)

Field	Description
StateProv	The state or province where your organization is located. Your registration policies determine whether you spell out the full name of the state or province or use an abbreviation. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Street <sup>1</sup>	The street address. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
Title	Job title. This is a text field of up to 64 characters. <b>Note:</b> The value is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.
TransactionId	PKISERV Web pages assign this after the user requests a certificate. When it is displayed, the user needs to record this number. This is a text field of up to 56 characters.
I UnstructAddr <sup>1</sup>	Unstructured address of the subject device. This is a text field of up to 64 characters.
I UnstructName <sup>1</sup>	Unstructured device name. This is a text field of up to 64 characters.
UserId	The owning SAF user ID. This is a text field of up to 8 characters.

**Restrictions:**

1. This field is applicable for only PKI certificates (certificates using the PKI: value in the SignWith field).
2. This field is applicable for only SAF certificates (certificates using the SAF: value in the SignWith field).

## The APPLICATION sections

The APPLICATION sections identify the application domains supported by PKI Services. The default certificate templates file (pkiserv.tmp1) ships with two applications sections, PKISERV (for PKI administrators) and CUSTOMERS (for general users).

The following is the format of the APPLICATION sections:

**<APPLICATION NAME=*appl-name*>...</APPLICATION>**

Each application section begins with an application name definition.

**Examples:**

```
<APPLICATION NAME="PKISERV">
```

This application contains support for all templates and functions.

```
<APPLICATION NAME="CUSTOMERS">
```

This application contains support for all templates and functions but does not include the button on the bottom of the PKI Services home page that directs users to the administration page.

Each APPLICATION section can contain the following subsections:

- CONTENT
- RECONTENT
- RESUCCESSCONTENT
- REFAILURECONTENT
- ADMINHEADER (only appears in the PKISERV application)
- ADMINFOOTER (only appears in the PKISERV application)
- ADMINSCOPE (only appears in the PKISERV application)

**<CONTENT>...</CONTENT>**

This subsection contains the HTML to display the PKI Services home Web page to the end user who is requesting and retrieving certificates. (See Figure 6 on page 199 for a sample Web page.) This subsection should contain one or more named fields (see “What are named fields?” on page 93) identifying certificate templates to use for requesting or managing certificates through this application. These template names should match the HTML selection value associated with them.

**<RECONTENT>...</RECONTENT>**

This subsection contains the HTML to display information about the certificate so that the end user can confirm that this is the correct certificate to renew or revoke. (See Figure 15 on page 212 for a sample Web page.) This subsection uses the substitution variable [printablecert], which contains the data extracted from the ICL entry. (See “What are substitution variables?” on page 91.)

**<RESUCCESSCONTENT>...</RESUCCESSCONTENT>**

This subsection contains the HTML to display a Web page to the end user when the revocation request is successful. Any named fields in this subsection are interpreted as HTML content inserts (for example, a page footer) that INSERT sections define. For PKISERV, the INSERT sections are included as part of the HTML for the Web page displayed to the end user.

**<REFAILURECONTENT>...</REFAILURECONTENT>**

This subsection contains the HTML to display a Web page to the end user when renewal or revocation request is unsuccessful. Any named fields in this subsection are interpreted as content inserts (for example, a page footer) that INSERT sections define. For PKISERV, the INSERT sections are included as part of the HTML for the Web page displayed to the end user.

**<ADMINHEADER>...</ADMINHEADER>**

This subsection contains the general installation-specific HTML content for the header of all administration Web pages. See “Steps for customizing the administration Web pages” on page 143 for more information.

**<ADMINFOOTER>...</ADMINFOOTER>**

This subsection contains the general installation-specific HTML content for the footer of all administration Web pages. See “Steps for customizing the administration Web pages” on page 143 for more information.

**<ADMINSCOPE>...</ADMINSCOPE>**

This optional subsection allows the administrator to choose a different CA domain. For more information, see “Adding a new CA domain” on page 161.

The following table summarizes the contents (Web pages) that the subsections of the APPLICATION sections generate.

Table 29. Subsections of the APPLICATION sections

Section or subsection	Contents
CONTENT	HTML for the “PKI Services Certificate Generation Application” Web page. (For a sample of this Web page, see Figure 6 on page 199.)

Table 29. Subsections of the APPLICATION sections (continued)

Section or subsection	Contents
RECONTENT	HTML for the Web page “Renew or revoke a browser certificate”. (For a sample of this Web page, see Figure 15 on page 212.)
RESUCCESSCONTENT	Contains only the named field <code>%%-renewrevokeok%%</code> (whose associated INSERT contains HTML for the Web page “Request submitted successfully”).
REFAILURECONTENT	Contains only the named field <code>%%-renewrevokebad%%</code> (whose associated INSERT contains HTML for the Web page “Request was not successful”).
ADMINSCOPE	This is for an administration page. It contains only the named field <code>%%SelectCADomain%%</code> to prompt the administrator to choose a domain. (For more information, see “Adding a new CA domain” on page 161.)
ADMINHEADER	This is for an administration page. (For more information, see “Customizing the administration Web pages” on page 143.)
ADMINFOOTER	This is for an administration page. (For more information, see “Customizing the administration Web pages” on page 143.)

## Templates that PKI Services provides

PKI Services provides the templates to request the following certificates:

- One-year SAF server certificate
- One-year SAF browser certificate
- One-year PKI SSL browser certificate (See Figure 8 on page 205 to see a sample of this Web page.)
- One-year PKI SSL S/MIME browser certificate
- Two-year PKI browser certificate for authenticating to z/OS
- Two-year PKI Authenticode—code signing server certificate
- Two-year PKI Windows logon certificate
- Five-year PKI SSL server certificate
- *n*-year PKI browser certificate for extensions demonstration
- Five-year SCEP certificate - Preregistration
- Five-year PKI IPSEC server (firewall) certificate
- Five-year PKI intermediate CA server certificate

The following table describes the certificate templates that PKI Services provides:

Table 30. Certificate templates PKI Services provides

Certificate template	Description
One-year SAF server certificate	This template allows end users to request a server certificate using native SAF certificate generation facilities (rather than PKI Services certificate generation facilities). The certificate is used for handshaking only (for example, SSL). This certificate is auto-approved.

## Customizing the end-user Web pages

Table 30. Certificate templates PKI Services provides (continued)

Certificate template	Description
One-year SAF browser certificate	This template allows end users to request a browser certificate. SAF certificate generation facilities (rather than PKI Services certificate generation facilities) create the certificate. The requestor must input a label (see Table 28 on page 95 for descriptions of fields) because the certificate is stored in a RACF database. This certificate is auto-approved.
One-year PKI SSL browser certificate	This template allows end users to request a browser certificate that PKI Services generates. The end user enters the common name. (See Table 28 on page 95 for descriptions of fields.) This template contains an ADMINAPPROVE section. Therefore, certificates requested using this template require administrator approval before being issued. The user ID and password are not required but the passphrase is required.
One-year PKI S/MIME browser certificate	This template allows end users to request a browser certificate that PKI Services generates. This is similar to the one-year PKI SSL browser certificate except the end user selects AltEmail.
Two-year PKI browser certificate for authenticating to z/OS	<p>This template allows end users to request a browser certificate that PKI Services generates. This is similar to the one-year PKI SSL browser certificate except this includes the %%HostIdMap%% INSERT and this certificate is auto-approved.</p> <p>%%HostIdMap%% is intended as a replacement for adding (and mapping) the certificate to a RACF user ID.</p> <p>This template specifies %%HostIdMap=@ host-name%% and %%UserId%% in the APPL section. This template does not require administrator approval but has protection through the user ID and password. (For more information about %%HostIdMap%%, see the HostIdMap field in Table 28 on page 95.)</p>
Two-year PKI Authenticode—code signing server certificate	This template allows end users to request a server certificate be used to sign software that will be downloaded across an untrusted medium. It also demonstrates how to define extensions for template specific certificate policies and third party provided OCSP.
Two-year PKI Windows logon certificate	<p>This template allows end users to request a certificate to use when logging in with a smart card to a Windows desktop as an Active Directory user. This template supports Internet Explorer based requests and supports the following cryptographic services providers (CSPs).</p> <ul style="list-style-type: none"> <li>• Datakey</li> <li>• Gemplus</li> <li>• Infineon SICRYPT</li> <li>• Schlumberger</li> </ul> <p>Support for additional CSPs can be added when you customize the template.</p>
Five-year PKI SSL server certificate	This template allows end users to request a server certificate that PKI Services generates. This is similar to the SAF server template except that this template contains an ADMINAPPROVE section. Therefore, certificates requested using this template require administrator approval before being issued. The user ID and password are not required but the passphrase is required.
Five-year PKI IPSEC server (firewall) certificate	This template allows end users to request a server certificate that PKI Services generates. This is similar to the five-year PKI SSL server certificate except that KeyUsage constants handshake and dataencrypt are hardcoded. Also, the end user selects AltEmail, AltIPAddr, AltURI, and AltDomain.

Table 30. Certificate templates PKI Services provides (continued)

Certificate template	Description
Five-year PKI intermediate CA server certificate	This template allows end users to request a server certificate that PKI Services generates. This is similar to the PKI SSL server template except that KeyUsage is hardcoded as certsign. Also, this certificate is auto-approved (because it runs under the user ID of the requestor, that is the person requesting this must be highly authorized). The user ID and password are required, and the units of work should run under the client's ID. In other words, the end user must be someone who can do this using RACDCERT alone, that is, must have CONTROL authority to IRR.DIGTCERT.GENCERT, and so forth. Given this requirement, the administrator need not approve this. The PassPhrase is required.
Five-year SCEP certificate—Preregistration	This template supports certificate preregistration for Simple Certificate Enrollment Protocol (SCEP) clients. The PassPhrase is required.
<i>n</i> -year PKI browser certificate for extensions demonstration	This template creates a browser certificate that has most of its information provided by the user rather than controlled by the administrator. The certificate contains all the supported extensions.

## TEMPLATE sections

TEMPLATE sections define the fields that comprise a specific certificate request. They define the certificate templates referenced in the APPLICATION section. The pkiserv.tmp1 certificate templates file contains eight TEMPLATE sections, for the eight certificates the preceding section describes.

Each template section begins with one or more template names.

**<TEMPLATE NAME=*tmpl-name*>...</TEMPLATE NAME>**

The pkiserv.tmp1 certificate templates file that ships with PKI Services includes lines like the following:

**Example:**

```
<TEMPLATE NAME=1-Year PKI SSL Browser Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=1YBSSL>
```

The true name of a certificate template is its actual complete name. This is the name in the first line, 1-Year PKI SSL Browser Certificate. However, you can refer to a single template by more than one name by using an alias. The template name in the second line, PKI Browser Certificate, is an alias. An alias simply differentiates browser from server certificates. Finally, renewing a certificate requires recalling the template name, so the template name must be stored with the certificate. The NICKNAME (or short name) serves this purpose.

**Notes:**

1. You can have more than one alias. (Use an additional <TEMPLATE NAME=*alias*> line for each one.)
2. The value of a NICKNAME is an 8-character string.
3. SAF certificate templates do not include nicknames.

The following table shows the true name, alias, and nickname for each certificate template:

## Customizing the end-user Web pages

Table 31. Names, aliases, and nicknames of certificate templates

True name	Alias	Nickname
1-Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1-Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
2-Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
2-Year PKI Authenticode—Code Signing Certificate	PKI Server Certificate	2YIACS
2-Year PKI Windows Logon Certificate	PKI Browser Certificate	2YBWL
5-Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5-Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5-Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
5-Year SCEP Certificate - Preregistration	—	5YSCEPP
<i>n</i> -Year PKI Certificate for Extensions Demonstration	PKI Browser Certificate	SAMPLB
1-Year SAF Browser Certificate	SAF Browser Certificate	—
1-Year SAF Server Certificate	SAF Server Certificate	—

TEMPLATE sections can have the following subsections:

- CONTENT
- APPL
- CONSTANT
- ADMINAPPROVE
- SUCCESSCONTENT
- FAILURECONTENT
- RETRIEVECONTENT
- RETURNCERT
- PREREGISTER

### <CONTENT>...</CONTENT>

This subsection contains the HTML to display a Web page to the end user requesting a certificate of a specific type. (See Figure 8 on page 205 for a sample Web page.) Field names on the certificate request (such as a text box where the user enters a value for Common Name) match the names of INSERT sections. The following examples show the INSERT sections corresponding to the field names `%%CommonName%%` and `%%Requestor (optional)%%`.

#### Examples:

```
<INSERT NAME=CommonName>
<p> Common Name [optfield]
<BR>
<INPUT NAME="CommonName" TYPE="text" SIZE=64 maxlength="64">
<INSERT>
```

```
<INSERT NAME=Requestor>
<p> Your name for tracking this request [optfield] <BR>
<INPUT NAME="Requestor" TYPE="text" SIZE=32 maxlength="32">
<INSERT>
```

Named fields in this subsection are optional if the named field contains more than one word within the %% delimiters (as in %%Requestor (optional)%%). The user need not supply a value for Requestor.

### <APPL>...</APPL>

This subsection identifies certificate fields for which the application itself should provide values. This subsection should contain only named fields, one per line. The only supported named fields allowed in this section are:

- UserId
- HostIdMap

#### Example:

```
<APPL>
%%UserId%%
%%HostIdMap=@www.ibm.com%%
</APPL>
```

### <CONSTANT>...</CONSTANT>

This subsection identifies certificate fields that have a constant (hardcoded) value for everyone. This subsection should contain only named fields, one per line. The syntax for specifying the values is %%field-name=field-value%%:

#### Example:

```
%%KeyUsage=handshake%%
```

In addition to the named fields listed in Table 28 on page 95, you may also include the following named fields in this subsection only.

#### Critical

Identifies a certificate extension that is to be marked critical in the issued certificates. This name-value pair may be repeated for each extension to be marked critical. Here is the list of acceptable values for **Critical**:

- BasicConstraints (ignored as this extension is always marked critical)
- KeyUsage (ignored as this extension is always marked critical)
- ExtKeyUsage
- SubjectAltName, AltEmail, AltIPAddr, AltDomain, AltURI
- HostIdMappings, HostIdMap
- CertificatePolicies, CertPolicies

#### Example:

```
%%Critical=ExtKeyUsage%%
```

#### Rules:

1. If you have specified configuration file setting PolicyRequired=T, then specifying %%Critical=CertPolicies%% will be ignored. The configuration file setting PolicyCritical will determine if the CertificatePolicies extension is marked critical. See “Using certificate policies” on page 146 for more information.



2. When ExtKeyUsage is extracted from an input PKCS #10 certificate request, the critical flag in the request is ignored. Therefore, setting `%%Critical=ExtKeyUsage%%` is the only way to get the ExtKeyUsage extension marked critical.

### CertPolicies

Identifies the certificate policies that are to be included in the issued certificates. The value is a vector of numbers each representing one of the PolicyName values specified in the **CertPolicy** section of the configuration file.

#### Example:

```
%%CertPolicies=3 6 10%%
```

**Rule:** If you have specified configuration file setting `PolicyRequired=T`, then specifying `%%CertPolicies=any-value%%` will be ignored. All issued certificates will have the same certificate policies as defined in the configuration file. See “Using certificate policies” on page 146 for more information.

### AuthInfoAcc

Indicates the information necessary for the AuthorityInfoAccess extension. The value specifies a two-part, comma-separated string identifying the access method (OCSP or IdentrusOCSP) and the access URL. The URL must be specified in HTTP-protocol format only. (LDAP protocol is not supported.) The name-value pair may be repeated for each value required in the extension.

#### Examples:

```
%%OCSP,URL=https://ocsp.vendor.com%%  
%%IdentrusOCSP,URI=https://identrus200.identrus.com%%  
%%OCSP,URI=http://www.mycompany.com/PKIServ/public-cgi/caocsp%%
```

### <ADMINAPPROVE>...</ADMINAPPROVE>

This optional subsection contains the named fields that the administrator can modify when approving certificate requests. (The named fields refer to INSERT sections.) When an end user requests a certificate, the certificate request may contain fields that the end user cannot see. When approving a request, the administrator can modify:

- Fields that are present and visible to the end user in the certificate request, for example Common Name
- Fields that are not visible to the end user but are hardcoded (in the CONSTANT subsection) in the template, for example Organizational unit
- Fields that are not visible to the end user and that the PKI Services administrator can add, for example, HostIdMappings extension or an empty Organizational Unit field (these are listed in the <ADMINAPPROVE> section, and either the end user did not fill them in or they are not present on the template request form).

The presence of this section (even if empty) indicates that an administrator must approve this request. The absence of this section indicates using auto-approval.



**Note:** In the `pkiserv.tmp1` certificate templates file, the only certificate templates that are auto-approved are the following:

- One-year SAF server certificate
- One-year SAF browser certificate
- Two-year PKI browser certificate for authenticating to z/OS
- Five-year PKI intermediate CA server certificate

You can put the following fields in the ADMINAPPROVE section:

- AltDomain
- AltEmail
- AltIPAddr
- AltOther\_OID
- AltURI
- CommonName
- Country
- EndDate
- ExtKeyUsage
- HostIdMap (can repeat)
- KeyUsage
- Locality
- Org
- OrgUnit (can repeat)
- StartDate
- StateProv
- Title

**Note:** The following fields are not modifiable and are ignored in the ADMINAPPROVE section:

- Label
- PublicKey
- Requestor
- SignWith
- UserId

(For information about fields, see Table 28 on page 95.)

**Example:**

```
<ADMINAPPROVE>
%%KeyUsage%%
%%CommonName%%
%%OrgUnit%%
%%Org%%
%%Country%%
%%HostIdMap%%
%%HostIdMap%%
%%HostIdMap%%
%%HostIdMap%%
</ADMINAPPROVE>
```

**<SUCCESSCONTENT>...</SUCCESSCONTENT>**

This subsection contains the HTML to display to the end user a Web page saying that the certificate request was submitted successfully. Any named fields in this subsection are interpreted as

content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML to display a Web page to the end user.

In all of the templates included with PKI Services, `<SUCCESSCONTENT>` contains only the named field `%%-requestok%%`. (See “What are named fields?” on page 93 for an explanation of named fields.) This contains HTML for the Web page “Request submitted successfully”. (For a sample of this Web page, see Figure 10 on page 207.)

### **`<FAILURECONTENT>...</FAILURECONTENT>`**

This subsection contains the HTML to display to the end user a Web page saying the certificate request was not submitted successfully. Any named fields in this subsection are interpreted as content inserts defined by INSERT sections. For PKISERV, the INSERT sections are included as part of the HTML to display a Web page to the end user.

In all of the templates included with PKI Services, `<SUCCESSCONTENT>` contains only the named field `%%-requestbad%%`. (See “What are named fields?” on page 93 for an explanation of named fields.) This contains HTML for the Web page that says, “Request was not successful”.

### **`<RETRIEVECONTENT>...</RETRIEVECONTENT>`**

This subsection contains the HTML to display to the end user a Web page to enable certificate retrieval. Any named fields in this subsection are interpreted as content inserts that the INSERT sections define. For PKISERV, the INSERT sections are included as part of the HTML presented to the end user.

For a sample of a Web page this section generates, see Figure 11 on page 208. You may want to look at this Web page while reading the following explanation:

In all of the templates included with PKI Services, `<RETRIEVECONTENT>` contains the following:

- The named field `%%-copyright%%`, which displays any copyright information. (See “What are named fields?” on page 93 for an explanation of named fields.)
- The title of the Web page (This appears in the banner of your browser. Figure 11 on page 208 does not include the banner header but shows only the frame containing the content and not the browser window displaying the content.)
- A JavaScript script for processing the fields the user enters the Web page.
- A heading that says “Retrieve Your (name of certificate)”. This uses the substitution variable `[tmplname]`. (See “What are substitution variables?” on page 91 for an explanation of substitution variables.)
- Text: a heading and paragraph about bookmarking this Web page.
- The named field `%%TransactionId%%` — A field where you enter your transaction ID if it is not already displayed.
- A field where you enter the passphrase you entered on the certificate request form.

**<RETURNCERT>...</RETURNCERT>**

This subsection contains the HTML to display to the end user a Web page upon successful certificate retrieval. For PKISERV, if the certificate being retrieved is a browser certificate, then this section must contain a single line containing a browser qualified INSERT name.

**Example:**

```
%%returnbrowsercert[browsertype]%%
```

Additionally, INSERTs for Netscape (returnbrowsercertNS) and Internet Explorer (returnbrowsercertIE) containing browser-specific HTML for returning certificates must be defined elsewhere in the pkiserv.tmp1 certificates template file. If the certificate being retrieved is a server certificate, this section should contain the HTML necessary to present the certificate to the user as text.

**<PREREGISTER>...</PREREGISTER>**

This optional subsection indicates the creation of a preregistration record and contains the Simple Certificate Enrollment Protocol (SCEP) rules for approval of a SCEP request. (For details, see “Overview of certificate request processing for preregistered SCEP clients” on page 176.)

**Example:**

```
<PREREGISTER>
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
</PREREGISTER>
```

**Summary of subsections contained in certificate templates**

The following table summarizes the subsections that are present in the various certificate templates in the pkiserv.tmp1 file (as it is shipped):

Table 32. Summary of subsections in certificate templates

Subsection (in TEMPLATE section)	1-year PKI SSL browser	1-year PKI SSL S/MIME browser	1-year SAF browser	1-year SAF server	2-year PKI browser for z/OS	2-year PKI Auth-enti-code signing server	2-year PKI Win-dows logon browser	5-year PKI SSL server	5-year PKI IPSEC server (fire-wall)	5-year PKI inter-med-iate CA server	5-year SCEP pre-reg-istra-tion	n-year PKI browser exten-sions demo
CONTENT	X	X	X	X	X	X	X	X	X	X	X	X
APPL			X	X	X					X		X
CONSTANT	X	X	X	X	X	X	X	X	X	X	X	X
ADMINAPPROVE	X	X				X	X	X	X			X
SUCCESSCONTENT	X	X	X	X	X	X	X	X	X	X	X	X
FAILURECONTENT	X	X	X	X	X	X	X	X	X	X	X	X
RETRIEVECONTENT	X	X	X	X	X	X	X	X	X	X		X
RETURNCERT	X	X	X	X	X	X	X	X	X	X		X
PREREGISTER											X	

## Summary of fields in certificate templates

The tables in this topic summarize the fields contained in each certificate template that PKI Services provides.

Fields in the PKI browser certificate templates	Table 33
Fields in the PKI server certificate templates	Table 34 on page 111
Fields in the SAF (browser and server) and SCEP certificate templates	Table 35 on page 113

The following tables identify each template field as one of the following:

- Required
- Optional
- Provided by the application
- Constant (supplied value is shown)
- Blank (field is not present in either the CONTENT or CONSTANT section)

Table 33. Summary of fields for PKI browser certificate templates

	One-year PKI SSL browser	One-year PKI S/MIME browser	Two-year PKI browser for z/OS	Two-year PKI Windows logon certificate	<i>n</i> -year PKI browser extensions demon- stration
Field name					
AltDomain					Optional
AltEmail		Required		Optional	
AltIPAddr					Optional
AltOther_ <i>OID</i>				Constant <sup>1</sup>	Optional
AltURI					Optional
AuthInfoAcc					Constant <sup>2</sup>
CertPolicies					Constant: 1
ClientName					
CommonName	Required		Constant <sup>3</sup>	Optional	
Country					Optional
Critical					
Email	Optional				Optional
EmailAddr					Optional
ExtKeyUsage	Constant: clientauth		Constant: clientauth	Constants: clientauth and mssmartlogon	Optional
HostIdMap <sup>4</sup>			Application provides		Optional
KeyUsage	Constant: handshake			Constant: digitalSig	Required
Label					Optional
Locality					Optional
Mail					Optional

Table 33. Summary of fields for PKI browser certificate templates (continued)

	One-year PKI SSL browser	One-year PKI S/MIME browser	Two-year PKI browser for z/OS	Two-year PKI Windows logon certificate	n-year PKI browser extensions demon- stration
Field name					
NotAfter	Constant: 365		Constant: 730		Optional
NotBefore	Constant: 0				Optional
NotifyEmail	Optional				
Org	Constant: The Firm				Optional
OrgUnit	Constant: Class 1 Internet Certificate CA				Required
OrgUnit2					Optional
PassPhrase	Required				
PostalCode					Optional
PublicKey	Browser provided				
Requestor	Optional				
SerialNumber					Optional
SignWith	Constant: PKI :				
StateProv					Optional
Street					Optional
Title					Optional
UnstructAddr					Optional
UnstructName					Optional
UserId			Application provides		
<b>Notes:</b>					
1. The constant value is _1_3_6_1_4_1_311_20_2_3.					
2. The constant value is OCSP,URL=https://IV.OCSP.BankXYZ.com.					
3. Although CommonName is a constant, no value is assigned to it. This indicates that RACF must determine the value. The user authenticates by specifying a user ID and password. (If UserId is listed in the APPL section, this means the application provides the user ID and password.) Providing the user ID and password enables RACF to look up the CommonName value in the user's profile.					
4. The HostIdMap value is formed by concatenating UserId with @host-name.					

Table 34. Summary of fields for PKI server certificate templates

Field name	Two-year PKI Authenticode code signing server	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI intermediate CA server
AltDomain		Optional		
AltEmail	Required	Optional		
AltIPAddr		Optional		
AltOther_ <i>OID</i>				
AltURI		Optional		
AuthInfoAcc	Constant <sup>1</sup>			

## Customizing the end-user Web pages

Table 34. Summary of fields for PKI server certificate templates (continued)

Field name	Two-year PKI Authenticode code signing server	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI intermediate CA server
CertPolicies	Constant: 1			
ClientName				
CommonName	Constant: My Company Code Signing Certificate	Optional		
Country		Optional		
Critical	Constant: ExtKeyUsage			
Email		Optional		
EmailAddr				
ExtKeyUsage	Constant: codesigning	Constant: serverauth		
HostIdMap <sup>2</sup>				
KeyUsage	Constants: digitalsig and docsign	Constant: handshake	Constants: handshake and dataencrypt	Constant: certsign
Label				
Locality		Optional		
Mail				
NotAfter	Constant: 730	Constant: 1825		
NotBefore	Constant: 0			
NotifyEmail	Required		Optional	
Org	Constant: The Firm	Optional		
OrgUnit	Optional			
OrgUnit2		Optional		
PassPhrase	Required			
PostalCode		Optional		
PublicKey	Required			
Requestor	Optional			
SerialNumber				
SignWith	Constant: PKI :			
StateProv		Optional		
Street		Optional		
Title				
UnstructAddr				
UnstructName				
UserId				Application provides

Table 34. Summary of fields for PKI server certificate templates (continued)

Field name	Two-year PKI Authenticode code signing server	Five-year PKI SSL server	Five-year PKI IPSEC server (firewall)	Five-year PKI intermediate CA server
<b>Notes:</b> <ol style="list-style-type: none"> <li>The constant value is OCSP,URL=https://ocsp.vendor.com.</li> <li>The HostIdMap value is formed by concatenating UserId with @host-name.</li> </ol>				

Table 35. Summary of fields for SAF and SCEP certificate templates

Field name	One-year SAF server	One-year SAF browser	Five-year SCEP preregistration
AltDomain	Optional		
AltEmail	Optional		
AltIPAddr	Optional		
AltOther_OID			
AltURI	Optional		Optional
AuthInfoAcc			
CertPolicies			
ClientName			Required
CommonName	Optional	Constant <sup>1</sup>	Optional
Country	Required	Constant: US	Optional
Critical			
Email			Optional
EmailAddr			Optional
ExtKeyUsage			Optional
HostIdMap <sup>2</sup>			Optional
KeyUsage			Optional
Label	Required		Optional
Locality	Optional		Optional
Mail			Optional
NotAfter	Constant: 365		Constant: 1825
NotBefore	Constant: 0		
NotifyEmail			Optional
Org	Required	Constant: The Firm	Optional
OrgUnit	Required	Constants: OrgUnit=SAF template certificate and OrgUnit=Nuts and Bolts Division	Optional
OrgUnit2	Optional		Optional
PassPhrase			Required
PostalCode			Optional
PublicKey	Required <sup>3</sup>	Browser provided <sup>4</sup>	Optional

Table 35. Summary of fields for SAF and SCEP certificate templates (continued)

Field name	One-year SAF server	One-year SAF browser	Five-year SCEP preregistration
Requestor			Optional
SerialNumber			Optional
SignWith	Constant: SAF:CERAUTH/taca		Constant: PKI:
StateProv	Optional		Optional
Street			Optional
Title			Optional
UnstructAddr			Optional
UnstructName			Optional
UserId	Application provides		
<b>Notes:</b>			
1. Although CommonName is a constant, no value is assigned to it. This indicates that RACF must determine the value. The user authenticates by specifying a user ID and password. (If UserId is listed in the APPL section, this means the application provides the user ID and password.) Providing the user ID and password enables RACF to look up the CommonName value in the user's profile.			
2. The HostIdMap value is formed by concatenating UserId with @host-name.			
3. The PublicKey is the PKCS #10 request.			
4. The PublicKey field is coded with the <i>browsertype</i> substitution variable.			

## Examining the pkiserv.tmpl file

This topic contains excerpts from the following sections of the pkiserv.tmpl file. Each excerpt contains numbered pointers that describe the important tags in each section.

- “Examining the APPLICATION section”
- “Examining the TEMPLATE section” on page 117
- “Examining the INSERT section” on page 120

The pkiserv.tmpl file begins with a prolog section of comments explaining main sections, subsections, named fields, and substitution variables. The prolog section is followed by a DEBUG tag that you can change from the default (DEBUG=0) to DEBUG=1 to get CGI debugging information.

## Examining the APPLICATION section

The APPLICATION section of the pkiserv.tmpl file contains two sample applications named PKISERV and CUSTOMERS.

- “Examining the PKISERV application”
- “Examining the CUSTOMERS application” on page 116

### Examining the PKISERV application

The following example is an excerpt of the PKISERV application in the APPLICATION section of the pkiserv.tmpl file. (The vertical ellipses indicate omitted sections.)

```
# =====
#
# Application - PKISERV
```



```

#
# The installation should customize the CONTENT, ADMINHEADER
# ADMINFOOTER, and ADMINSCOPE subsections as appropriate
#
# =====
#
<APPLICATION NAME=PKISERV> 1
<CONTENT> 2
<HTML><HEAD>
<TITLE>PKI Administrators Start Page</TITLE>
%%-copyright%%
</HEAD>
<BODY>
<<H1>PKI Administrators Start Page</H1>
<p>
<A HREF="/PKIServ/cacerts/cacert.der"> 3
Install the CA certificate to enable SSL sessions for PKI Services </A>
<H2>Choose one of the following:</H1>
<p>
<h3>Manage existing certificates and certificate requests</h3>
# The following action will force userid/pw authentication for
# administrators
<FORM name=admform METHOD=GET ACTION="/PKIServ/ssl-cgi/auth/admain.rexx"> 4
# The following action will force client certificate authentication for
# administrators
#<FORM name=admform METHOD=GET
# ACTION="/PKIServ/clientauth-cgi/auth/admain.rexx">
<p>
<INPUT TYPE="submit" VALUE="Administration Page">
</FORM>
# Multiple CA mode - replicate and modify the following H3 and FORM
# section for each CA domain.
<h3>Go to the Customers' home page </h3>
<FORM name=admform METHOD=GET ACTION="/Customers/public-cgi/camain.rexx"> 5
<p>
<INPUT TYPE="submit" VALUE="Customers' Home Page">
</FORM>
<p> %%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<ADMINHEADER> 6
<HTML><HEAD>
<TITLE>Web Based Certificate Generation Administration</TITLE>
%%-copyright%%
</HEAD>
<BODY>
</ADMINHEADER>
<ADMINFOOTER>
<p> %%-pagefooter%% 7
</BODY>
</HTML>
</ADMINFOOTER>
<ADMINSCOPE> 8
# Uncomment the following line to enable multiple CA domains
# %%SelectCADomain%%
</ADMINSCOPE>
</APPLICATION>

```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the PKISERV application.

1. This is the beginning of the APPLICATION section. The name of the application is PKISERV.
2. This is the beginning of the CONTENT subsection. The CONTENT subsection contains HTML to display the Web page where the administrator begins. The

- TITLE indicates the main heading of that Web page, “PKI Administrators Start Page.” (See Figure 18 on page 218 for a sample of that Web page.)
3. The HREF tag is the link to install the CA certificate in the browser.
  4. The ACTION tag indicates where to go when the user clicks the **Administration Page** button. (See Figure 21 on page 222 for a sample of that Web page.)
  5. The ACTION tag indicates where to go when the user clicks the **Customers' Home Page** button. (See Figure 6 on page 199 for a sample of that Web page.)
  6. The ADMINHEADER subsection references the %%-copyright%% named field, which is defined in the INSERT section. This should contain the copyright statement for your company.
  7. The ADMINFOOTER subsection references the %%-pagefooter%% named field, which is defined in the INSERT section. This named field should specify the e-mail address of your PKI Services administrator.
  8. The ADMINSCOPE subsection references the %%SelectCADomain%% named field, which is defined in the INSERT section. When you have multiple CA domains, you can use this variable to allow PKI administrators to select a CA domain on the administrator's home page. (See “Adding a new CA domain” on page 161 for details about implementing multiple CA domains.)

### Examining the CUSTOMERS application

The following example is an excerpt of the CUSTOMERS application in the APPLICATION section of the pkiserv.tmp1 file. (The vertical ellipses indicate omitted sections.)

```
# =====
#
# Application - CUSTOMERS
#
# The installation should customize the CONTENT subsection as appropriate.
# =====
#
<APPLICATION NAME=CUSTOMERS> 1
<CONTENT> 2
<HTML><HEAD>
<TITLE> Customers Certificate Generation Application </TITLE>
%%-copyright%%
</HEAD>
<BODY>
<H1>PKI Certificate Generation Application</H1>
<p>
<A HREF="/PKIServ/cacerts/cacert.der"> 3
Install the CA certificate to enable SSL sessions for PKI Services </A>
<H2>Choose one of the following:</H2>
<ul>
<li><h3>Request a new certificate using a model</h3>
<FORM name=mainform METHOD=GET ACTION="/[application]/ssl-cgi/catmpl.rexx"> 4
<p> Select the certificate template to use as a model
<SELECT NAME="Template"> 5
  %%1-Year PKI SSL Browser Certificate%%
    <OPTION>1-Year PKI SSL Browser Certificate
  %%1-Year PKI S/MIME Browser Certificate%%
    <OPTION>1-Year PKI S/MIME Browser Certificate
  %%2-Year PKI Windows Logon Certificate%%
    <OPTION>2-Year PKI Windows Logon Certificate
  :
</HTML>
</CONTENT>
<RECONTENT> 6
<HTML><HEAD>
<TITLE> Customers Renew or Revoke a Browser Certificate </TITLE>
```

```

%%-copyright%%
</HEAD>
<BODY>
<H1>Renew or Revoke a Browser Certificate</H1>
:
</BODY>
</HTML>
</RECONTENT>
<RESUCCESSCONTENT> 7
  %%-renewrevokeok%%
</RESUCCESSCONTENT>
<REFAILURECONTENT> 8
  %%-renewrevokebad%%
</REFAILURECONTENT>
</APPLICATION>

```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the CUSTOMERS application.

1. This is the beginning of the APPLICATION section. The name of the application is CUSTOMERS.
2. This is the beginning of the CONTENT subsection. The CONTENT subsection contains HTML to display the Web page where the end user requests or retrieves a certificate. The <H1> indicates the main heading of that Web page, “PKI Certificate Generation Application.” (See Figure 6 on page 199 for a sample of that Web page.)
3. The HREF tag is the link to install the CA certificate in the browser.
4. The ACTION tag indicates where to go when the user clicks the **Request certificate** button.
5. The SELECT tag produces a drop-down that lists the certificate templates the user can request. (The named fields, which are bracketed with %% symbols, are the names of the certificate templates.)
6. The RECONTENT section contains the HTML to display the Web page where the end user renews or revokes a certificate. The main heading on this Web page is “Renew or Revoke a Browser Certificate”. (See Figure 15 on page 212 for a sample of that Web page.)
7. The RESUCCESSCONTENT subsection references the %%-renewrevokeok%% named field, which is defined in the INSERT section. This contains HTML for the Web page displayed when the user’s attempt to revoke a certificate is successful. The main heading on this Web page is “Request submitted successfully”. (See Figure 10 on page 207 for a sample of that Web page.)
8. The REFAILURECONTENT subsection references the %%-renewrevokebad%% named field, which is defined in the INSERT section. This contains HTML for the Web page displayed when the user’s attempt to renew or revoke a certificate fails. The main heading on this Web page is “Request was not successful”.

## Examining the TEMPLATE section

The TEMPLATE section follows the APPLICATION section and contains several sample templates. The following example is an excerpt from the TEMPLATE section of the pkiserv.tpl file. (The vertical ellipses indicate omitted sections.)

```

# =====
#
# Template Name - 2-Year PKI Browser Certificate For Authenticating
#               to z/OS 1
#
# Function - Creates a 2-Year certificate good for authenticating to z/OS.
#
# User input fields:

```

## Customizing the end-user Web pages

```
# Requestor - optional
# PassPhrase - required
# PublicKey - required (Provided by the browser itself)
# NotifyEmail - optional
:
:
# =====
#
<TEMPLATE NAME=2-Year PKI Browser Certificate For Authenticating To z/OS> 2
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=2YBZOS>
<CONTENT> 3
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE> 4
%%-copyright%% 5
%%-AdditionalHead[browsertype]%%
</HEAD>

<BODY>
<H1>2-Year PKI Browser Certificate For Authenticating To z/OS</H1> 6
<p>
<H2>Choose one of the following:</H2>
:
:
#<FORM NAME="CertReq" METHOD=POST ACTION= 7
#      "[application]/ssl-cgi-bin/careq.rexx" onSubmit=
      "return ValidateEntry(this)">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s) 8
<SCRIPT LANGUAGE="JavaScript"> 9
<!--
:
:
//-->
</SCRIPT>
  %%Requestor (optional)%%
  %%NotifyEmail (optional)%%
  %%PassPhrase%%
  %%PublicKey2[browsertype]%%

<p>
<INPUT TYPE="Submit" VALUE="Submit certificate request">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="[application]/ssl-cgi-bin/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</ul>
<p>%%-pagefooter%% 10
</BODY>
</HTML>
</CONTENT>
<APPL> 11
  %%UserId%%
  %%HostIdMap=@host-name%%
</APPL>
<CONSTANT> 12
  %%CommonName=%%
  %%OrgUnit=Class 1 Internet Certificate CA%%
  %%Org=The Firm%%
  %%KeyUsage=handshake%%
  %%ExtKeyUsage=clientauth%%
  %%NotBefore=0%%
  %%NotAfter=730%%
  %%SignWith=PKI:%%
</CONSTANT>
```

```
<SUCCESSCONTENT> 13
%%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT> 14
%%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT> 15
<HTML><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
:
:
//-->
</SCRIPT>
</HEAD>

<BODY>
<H1> Retrieve Your [tplname]</H1> 16
<H3>Please bookmark this page</h3>
:
:
#<FORM NAME=retrieveform METHOD=POST ACTION= 17
# "[application]/ssl-cgi-bin/cagetcert.rexx" onSubmit=
:
:
</FORM>
:
:
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT> 18
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>
```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the `TEMPLATE` section.

1. The template begins with a block comment identifying the template and explaining its use and fields.
2. There are three names for each certificate (except for SAF templates, which do not include nicknames). The first TEMPLATE NAME line defines the true (actual, complete) name of the certificate. The next TEMPLATE NAME line defines an alias. (This simply differentiates browser from server certificates.) The NICKNAME defines an 8-character string.
3. The CONTENT subsection contains the HTML to display a Web page to the end user requesting this type of certificate. (The CGI script `catmpl.rexx` displays this content.)
4. The title contains the heading that appears at the very top of the browser when the Web page is displayed.
5. The `%%-copyright%%` named field displays the copyright statement.
6. The heading is the main heading on the Web page for requesting the selected certificate.
7. The ACTION tag indicates that the CGI script that gets control when the user clicks the **Submit certificate request** button is `careq.rexx`.
8. Fields for which the user can supply input include `%%Requestor%%`, `%%PassPhrase%%`, `%%NotifyEmail%%`, and `%%PublicKey2%%`. (These fields are named fields that are defined in the INSERT section, which is shown later.) All fields not marked optional are required. `%%PublicKey2%%` contains the

## Customizing the end-user Web pages

- substitution variable, [browsertype]. This is replaced at run time with IE or NS, depending on the browser the user has. This is necessary because the browsers behave differently for key generation and certificates.
9. This JavaScript script provides the underlying logic for the text entry that the user must perform.
  10. The %%-pagefooter%% named field is defined in the INSERT section (shown later). This contains the e-mail address of the PKI Services administrator.
  11. The APPL subsection indicates the fields that careq.rexx itself provides, in this case, %%UserId%% and %%HostIdMap%%. (These are set from the z/OS HTTP Server environment variable REMOTE\_USER.)
  12. The CONSTANT subsection has hardcoded values to use, for example (for the non-SAF certificates), the signing certificate is PKI:.
  13. The SUCCESSCONTENT subsection contains the HTML to display upon successfully requesting the certificate. It includes the %%-requestok%% named field. (This is defined in the INSERT section, shown later. See list item 1 on page 122.)
  14. The FAILURECONTENT subsection contains the HTML to display when the certificate request is unsuccessful. This subsection contains the %%-requestbad%% named field. (This named field is defined in the INSERT section, shown later.)
  15. The -requestok INSERT (mentioned in list item 13) includes an ACTION that calls caretrieve.rexx, which displays the HTML in the RETRIEVECONTENT subsection. The first time the Web page is displayed, it includes the transaction ID associated with the certificate request. If the user leaves the Web page and then returns, the transaction ID field must be filled in. Entering the transaction ID and clicking the **Continue** button calls cagetcert.rexx.
  16. The main heading on the Web page is "Retrieve Your (Name of Certificate)".
  17. The ACTION is to call cagetcert.rexx as list item 15 indicates.
  18. The RETURNCERT subsection contains the %%return10cert%% named field, which is defined in an INSERT. (See list item 4 on page 122.)

## Examining the INSERT section

The final section of the pkiserv.tmp1 file contains several sample INSERTS. The following example is an excerpt from the INSERT section of the pkiserv.tmp1 file. (The vertical ellipses indicate omitted sections.)

```
# =====
#
# Sample INSERTS
#
# =====
#
<INSERT NAME=-AdditionalHeadIE>
<OBJECT
  classid="clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
  CODEBASE="xenroll.cab#Version=5,131,3659,0"
  id="certmgr"
>
</OBJECT>
</INSERT>

<INSERT NAME=-requestok> 1
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Success</TITLE>
</HEAD>
<BODY>
<H1> Request submitted Successfully</H1>
```

```

[errorinfo]
<p> Here's your transaction ID. You will need it to retrieve your
certificate. Press 'Continue' to retrieve the certificate.
<p> <TABLE BORDER><TR><TD>[transactionid]</TD></TR></TABLE>
<FORM METHOD=GET ACTION="/[application]/ssl-cgi/caretrieve.rexx"> 2
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT NAME="TransactionId" TYPE="hidden" VALUE="[transactionid]">
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>

<INSERT NAME=-requestbad> 3
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
:
<INSERT NAME=-return10cert> 4
<HTML><HEAD>
<TITLE> Web Based SAF Certificate Generation Application Pg 4</TITLE>
</HEAD>
<BODY>
<H1> Here's Your Certificate. Cut and Paste it to a File</H1>
<TABLE BORDER><TR><TD>
<PRE>
[base64cert] 5
</PRE>
</TD></TR></TABLE>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
:
</BODY>
</HTML>
</INSERT>
#
# =====
#
# X.509 fields (INSERTs) valid for certificate requests
#
# =====
#
:
<INSERT NAME=PublicKeyIE> 6
<SCRIPT LANGUAGE="VBScript">
<!--
:
// -->
</SCRIPT>

# =====

```

## Customizing the end-user Web pages

```
:
:
<INSERT NAME=PassPhrase> 7
<p> Pass phrase for securing this request. You will need to supply
this value when retrieving your certificate [optfield] <BR>
<INPUT NAME="PassPhrase" TYPE="password" SIZE=32 maxlength="32"> <BR>
<p> Reenter your pass phrase to confirm <BR>
<INPUT NAME="ConfirmPassPhrase" TYPE="password" SIZE=32
maxlength="32">
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidPassPhrase(frm){
  if ("[optfield]" == "" && frm.PassPhrase.value == "") {
    alert("Enter required field."); frm.PassPhrase.focus();
    return false;
  }
  if ("[optfield]" == "" && frm.ConfirmPassPhrase.value == "") {
    alert("Reenter the pass phrase."); frm.ConfirmPassPhrase.focus();
    return false;
  }
  if (frm.PassPhrase.value != frm.ConfirmPassPhrase.value) {
    alert("Passwords don't match. Reenter."); frm.PassPhrase.focus();
    return false;
  }
  return true;
}
//-->
</SCRIPT>
</INSERT>
:
<INSERT NAME=-pagefooter>
<p>email: webmaster@your_company.com
</INSERT>
```

The numbers in the following list refer to the highlighted tags in the preceding excerpt of the INSERT section.

1. The -requestok INSERT has the logic to generate the certificate. If the certificate is successfully generated, a Web page (whose main heading is "Request submitted successfully") is displayed. This Web page includes the transaction ID.
2. The -requestok INSERT includes an ACTION that calls caretrieve.rexx, which allows the user to retrieve the certificate.
3. Alternately, if the request is not successful, the -requestbad INSERT gains control.
4. (The caretrieve.rexx CGI displays the RETRIEVECONTENT subsection (see list item 15 on page 120) HTML, which displays a Web page that prompts the user for the transaction ID associated with the certificate request. The user enters the transaction ID (and any password) and clicks the **Continue** button, which calls cagetcert.rexx.) The cagetcert.rexx CGI calls R\_PKIServ for EXPORT of the certificate. If the export is successful, cagetcert.rexx displays the HTML under the RETURNCERT subsection. (See list item 18 on page 120.)
5. The base64-encoded certificate is displayed on the Web page by using the [base64cert] substitution variable.
6. This is a browser-qualified PublicKey INSERT for Internet Explorer.
7. Additional INSERTs are certificate field name INSERTs. These describe the fields using the HTML dialogs that are displayed on the Web pages if the user is allowed to input these fields. For example, PassPhrase is a text field with a maximum length of 32 characters. The two-year PKI browser certificate for authenticating to z/OS allows the user to fill in this field. (%%PassPhrase%% is listed in the input fields; see list item 8 on page 119.)



## Relationship between CGIs and the pkiserv.tpl file

CGIs for the end-user Web pages are execs that gain control when the end user clicks an action button—for example, the **Request certificate** button on the PKI Services home page. The CGIs read the `pkiserv.tpl` file to determine the action to perform. They resolve substitution variables in the `pkiserv.tpl` file.

The following are the CGIs for the end-user Web pages (including their directories):

- `/usr/lpp/pkiserv/PKIServ/public-cgi/camain.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/catmpl.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/careq.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/caretrieve.rexx`
- `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx`
- `/usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/cadisplay.rexx`
- `/usr/lpp/pkiserv/PKIServ/clientauth-cgi-bin/camodify.rexx`

The following table summarizes the actions the CGIs perform:

Table 36. CGI actions for end-user Web pages

CGI exec	Action	Sample Web page
camain.rexx	<ul style="list-style-type: none"> <li>When user clicks the <b>Request certificate</b> button, this calls <code>catmpl.rexx</code>, passing it a parameter identifying the selected template.</li> <li>The user can click the <b>Pick up certificate</b> button to go directly to <code>caretrieve.rexx</code> (if the certificate is already requested).</li> <li>The user can click the <b>Renew or revoke certificate</b> button to go to <code>cadisplay.rexx</code>.</li> <li>An administrator can click the <b>Go to administration page</b> button to go to <code>admmain.rexx</code>. (See Table 40 on page 141 for more information about <code>admmain.rexx</code>.)</li> </ul>	See Figure 6 on page 199.
catmpl.rexx	<ul style="list-style-type: none"> <li>Displays Web page coded in the HTML under the CONTENT subsection (of a TEMPLATE section).</li> <li>When the user clicks the <b>Submit certificate request</b> button, this passes template and field name parameters to <code>careq.rexx</code>.</li> <li>When the user clicks the <b>Retrieve your certificate</b> button, this passes control to <code>caretrieve.rexx</code>.</li> </ul>	See Figure 8 on page 205.
careq.rexx	<ul style="list-style-type: none"> <li>Processes field names under the APPL subsection (of a TEMPLATE section).               <p><b>Note:</b> Depending on the template, this can be:</p> <ul style="list-style-type: none"> <li>– <code>UserId</code> only</li> <li>– <code>UserId</code> and <code>HostIdMap</code>.</li> </ul> </li> <li>Processes hardcoded field names under the CONSTANT subsection (of a TEMPLATE section).</li> <li>Depending on the results, displays Web page coded in the HTML under the SUCCESSCONTENT or FAILURECONTENT subsection (of a TEMPLATE section):               <ul style="list-style-type: none"> <li>– The SUCCESSCONTENT subsection includes a <b>Continue</b> button the user can click to continue to <code>caretrieve.rexx</code>.</li> </ul> </li> </ul>	See Figure 10 on page 207.

## Customizing the end-user Web pages

Table 36. CGI actions for end-user Web pages (continued)

CGI exec	Action	Sample Web page
caretrieve.rexx	<ul style="list-style-type: none"><li>Displays Web page coded in the HTML under the RETRIEVECONTENT subsection (of a TEMPLATE section). This HTML prompts the user to enter the transaction ID and a password if the user entered one when requesting the certificate.</li><li>When the user clicks the <b>Retrieve and install certificate</b> button, this passes the transaction ID parameter to cagetcert.rexx.</li></ul>	See Figure 11 on page 208.
cagetcert.rexx	<ul style="list-style-type: none"><li>Displays Web page coded in the HTML under RETURNCERT subsection (of a TEMPLATE section). This HTML determines which of the following forms to use when returning the certificate:<ul style="list-style-type: none"><li>as a base64-encoded certificate (for server certificates)</li><li>as an ActiveX object (for Microsoft Internet Explorer browser certificates)</li><li>as an application/x-x509-user-certificate MIME type (for Netscape browser certificates).</li></ul></li></ul>	See Figure 12 on page 209.
cadisplay.rexx	<ul style="list-style-type: none"><li>Displays Web page coded in the HTML under the RECONTENT subsection (of the APPLICATION section).</li><li>For renewing a certificate, the user fills in the passphrase and clicks the <b>Renew</b> button. For revoking a certificate, the user clicks the <b>Revoke</b> button. Both actions call camodify.rexx.</li></ul>	See Figure 15 on page 212.
camodify.rexx	<ul style="list-style-type: none"><li>Displays Web page coded in the HTML under the SUCCESSCONTENT subsection (of a TEMPLATE section) for a successful renewal. The SUCCESSCONTENT subsection includes a <b>Continue</b> button the user can click to call caretrieve.rexx.</li><li>Displays the Web page coded in HTML under the RESUCCESSCONTENT subsection (of the APPLICATION section) for a successful revocation.</li></ul>	See Figure 10 on page 207.

## Steps for performing minimal customization

You need to perform these steps only if you are customizing certificate templates for the first time. If your company used an earlier release of PKI Services, you do *not* need to do so again.

**Before you begin:** Review the certificate templates and decide if there are any that you want to remove from the pkiserv.tmp1 certificates template file. If so, do this first. (To remove a certificate template, you can simply remove its name from the appropriate APPLICATION sections.)

### Notes:

- Fields such as %%0rg%%, %%Country%%, and so forth are used to form the subject's distinguished name. Therefore, make sure that the name formed has a suffix that matches a suffix that the LDAP directory supports (that is, that it matches one of the suffix values in the LDAP server configuration file).
- The default name of the LDAP server configuration file is slapd.conf for the z/OS Integrated Security Services LDAP server.

Perform the following steps to do the minimal updates on the remaining certificate templates:

1. For the SAF templates, update the following fields as needed:

- a. If present, replace the OrgUnit values in the following lines with values more appropriate to your organization:

```
%%OrgUnit=Nuts and Bolts Division%%
%%OrgUnit=SAF template certificate%%
```

- b. Replace `taca` in the following line with the correct label of the CERTAUTH signing certificate:

```
%%SignWith=SAF:CERTAUTH/taca%%
```

---

2. For the PKI templates, replace the OrgUnit value in the following line with a value more appropriate for your organization:

```
%%OrgUnit=Class 1 Internet Certificate CA%%
```

---

3. If present, replace The Firm with the name of your company in the following %%Org line:

```
%%Org=The Firm%%
```

---

4. If your company location is not the United States, update the following line by specifying the correct two-letter country abbreviation:

```
%%Country=US%%
```

---

5. If present, replace `host-name` with the domain name of this system in the following %%HostIdMap line:

```
%%HostIdMap=@host-name%%
```

You also need to follow the instructions in “Administering HostIdMappings extensions” on page 242.

---

6. For non-SAF certificates, you can notify users when certificate requests are rejected or when certificates are ready for retrieval or are expiring.

- a. If you do not want to have NotifyEmail appear as an input field for any non-SAF certificates, delete the NotifyEmail lines in the following locations in the TEMPLATE section for this certificate:

- In the header:

```
# NotifyEmail - optional
```

- In the list of fields:

```
%%NotifyEmail (optional)%%
```

- b. If you do not want to have NotifyEmail appear as an input field for renewal of any non-SAF certificates, delete the following NotifyEmail line in the APPLICATION section and in the list of fields:

```
%%NotifyEmail (optional)%%
```

---

7. Insert the copyright statement for your company in the `-copyright` named field in the INSERT section.
- 

8. Insert the e-mail address of your company's PKI Services administrator in the `-pagefooter` named field in the INSERT section.

### Steps for additional first-time customization

---

You need to perform these steps only if you are customizing certificate templates for the first time. If your company used an earlier release of PKI Services, you do *not* need to perform these steps.

Perform the following steps if you want to perform additional customization of the end-user Web pages:

1. Review the templates and decide which one(s) you need to update.
2. If necessary, change the true name, alias, or nickname, as in the following lines.

```
<TEMPLATE NAME=true_name>  
<TEMPLATE NAME=alias>  
<NICKNAME=nickname>
```

#### *true\_name*

Is the whole and complete name of the certificate template.

#### *alias*

Differentiates browser from server certificates. An alias is not required. You can have more than one alias.

#### *nickname*

Is an 8-character name. SAF certificates do not have nicknames. If a nickname is not present, the certificate is not renewable.

#### **Example:**

```
<TEMPLATE NAME=1-Year PKI SSL Browser Certificate>  
<TEMPLATE NAME=PKI Browser Certificate>  
<NICKNAME=1YBSSL>
```

3. If necessary, in the CONTENT subsection, change the certificate fields listed. The following example is from the one-year PKI SSL browser certificate template.

#### **Example:**

```
<p> Enter values for the following field(s)  
%%CommonName%%  
%%Requestor (optional)%%  
%%PassPhrase%%  
%%PublicKey2[browsertype]%%
```

4. If you add required fields in the preceding step, update the JavaScript code that is part of the embedded HTML to check for required fields that are missing.

#### **Example:**

```
ValidCommonName(frm) &&  
ValidPassPhrase(frm) &&  
ValidPublicKey2(frm) &&
```

---

5. If necessary, in the APPL subsection, change the list of certificate fields that the application provides. (Currently, the only supported fields are UserId and HostIdMap.) The following example is from the two-year PKI browser certificate for authenticating to z/OS:

**Example:**

```
<APPL>
  %%UserId%%
  %%HostIdMap=@host-name%%
</APPL>
```

6. If necessary, in the CONSTANT subsection, update the list of certificate fields whose values are hardcoded. The following example is from the one-year PKI SSL browser certificate template:

**Example:**

```
<CONSTANT>
  %%NotBefore=0%%
  %%NotAfter=365%%
  %%KeyUsage=handshake%%
  %%OrgUnit=Class 1 Internet Certificate CA%%
  %%Org=The Firm%%
  %%SignWith=PKI:%%
</CONSTANT>
```

**Note:** If you update the CONSTANT subsection to create subject distinguished names, make sure that the names match the LDAP suffix defined for your LDAP server. Otherwise the certificates are not posted to LDAP. PKI Services constructs the subject distinguished name from the fields specified in the following order:

- CommonName
- Title
- OrgUnit (if repeating, in the order that they appear in the template file)
- Org
- Locality
- StateProv
- Country

7. If necessary, edit the ADMINAPPROVE subsection. (Certificates requiring an administrator's approval have an ADMINAPPROVE subsection. The absence of the ADMINAPPROVE subsection indicates auto-approval for requests.) Make sure the ADMINAPPROVE subsection, if present, correctly lists the minimum set of certificate fields that the administrator can change.

**Notes:**

- a. There may be more fields in the ADMINAPPROVE subsection than fields that the user can complete in the certificate request (because the users do not necessarily see all fields).
- b. Do not include the Requestor, Label, UserId, PublicKey, or SignWith fields in the ADMINAPPROVE subsection; these fields cannot be changed and are ignored if present. (See page 107 for a list of fields that can be in the ADMINAPPROVE subsection.)

The following example of the ADMINAPPROVE subsection is from the one-year PKI SSL browser certificate template:

### Example:

```
<ADMINAPPROVE>
%%CommonName (Optional)%%
%%OrgUnit (Optional)%%
%%OrgUnit (Optional)%%
%%Org (Optional)%%
%%NotBefore (optional)%%
%%NotAfter (Optional)%%
%%KeyUsage (Optional)%%
%%HostIdMap (Optional)%%
%%HostIdMap (Optional)%%
%%HostIdMap (Optional)%%
%%HostIdMap (Optional)%%
</ADMINAPPROVE>
```

**Note:** The four %%HostIdMap%% lines in the example indicate that the approver can provide up to four HostIdMap entries.

---

### 8. If necessary, update the following:

- The SUCCESSCONTENT subsection contains only the %%-requestok%% named field, which contains the HTML for the Web page whose main heading is “Request submitted successfully”. To make changes to this Web page, update the -requestok INSERT (in the INSERT section of pkiserv.tpl):

```
<INSERT NAME=-requestok>
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Success</TITLE>
</HEAD>
<BODY>
<H1> Request submitted Successfully</H1>
[errorinfo]
<p> Here's your transaction ID. You will need it to retrieve your
certificate. Press 'Continue' to retrieve the certificate.
<p> <TABLE BORDER><TR><TD>[transactionid]</TD></TR></TABLE>
<FORM METHOD=GET ACTION="/PKIServ/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT NAME="TransactionId" TYPE="hidden" VALUE="[transactionid]">
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
```

- The FAILURECONTENT subsection contains only the %%-requestbad%% named field, which contains the HTML for the Web page whose main heading is “Request was not successful”. To make changes to this Web page, update the requestbad INSERT:

```
<INSERT NAME=-requestbad>
<HTML><HEAD>
<TITLE> Web Based Certificate Generation Failure</TITLE>
</HEAD>
<BODY>
<H1> Request was not successful</H1>
<p> Please correct the problem or report the error to your Web admin
person<br>
<PRE>
[errorinfo]
</PRE>
```

```
<p>%%-pagefooter%%
</BODY>
</HTML>
</INSERT>
```

---

9. If necessary, update the RETRIEVECONTENT subsection.

**Note:** See “Steps for changing the runtime user ID for retrieving certificates” on page 134 for directions for changing the runtime user ID for retrieving a certificate.

- a. The RETRIEVECONTENT subsection includes the %%-copyright%% named field. If you want to make any changes in the copyright statement, update the copyright INSERT. (The following is the copyright INSERT as it is originally provided in the pkiserv.tmp1 file. You should have previously updated this INSERT by providing information tailored to your company, as described in “Steps for performing minimal customization” on page 124.)

```
<INSERT NAME=-copyright>
<!--
/*****
/*
/* LICENSED MATERIALS - PROPERTY OF IBM
/* THIS SCRIPT IS "RESTRICTED MATERIALS OF IBM"
/* 5647-A01 (C) COPYRIGHT IBM CORP. 2000,2001
/*
/*
*****/
-->
</INSERT>
```

- b. If necessary, update any desired Web page content (such as headers, footers, titles, background colors, frames, links, and so on) for the Web page whose main heading is “Retrieve Your (certificate template name)”.

---

10. If you are updating the template for a server certificate, you can update the HTML in the RETURNCERT subsection to customize the returned Web page. (For a browser template, you cannot change the RETURNCERT subsection. It must contain the %%returnbrowsercert%% named field, which contains the [ browsertype ] substitution variable. The INSERT section contains browser-specific returnbrowsercert INSERTs.)

---

## Steps for retrofitting release changes into the PKI Services certificate templates

If you used an earlier release of PKI Services, you may need to retrofit changes in the pkiserv.tmp1 certificate templates file. (You would not want to replace the file if you customized it in the previous release.)

You can use a file comparison tool to compare the new PKI Services certificates template file (/usr/lpp/pkiserv/samples/pkiserv.tmp1) and your existing PKI Services certificates template file (/etc/pkiserv/pkiserv.tmp1).

Perform the following steps to retrofit changes into the pkiserv.tmp1 certificate templates file so you do not lose any customizations you made in a previous release.

## Customizing the end-user Web pages

1. Make a backup copy of your current certificate templates file. For example, enter from the UNIX command line:  

```
cp /etc/pkiserv/pkiserv.tpl /etc/pkiserv/pkiserv.backup
```
2. Copy the new sample templates file to the runtime location. (This is the copy you will edit.)  

```
cp /usr/lpp/pkiserv/samples/pkiserv.tpl /etc/pkiserv/pkiserv.tpl
```
3. Using a compare program of your choice, compare the two template files:  

```
/etc/pkiserv/pkiserv.tpl  
/etc/pkiserv/pkiserv.backup
```
4. Edit the runtime copy of the templates file (/etc/pkiserv/pkiserv.tpl). Using the compare output generated in Step 3, merge the changes you made to the original template file into the runtime copy of the templates file.
5. Exit the file to save your changes.

## Locating code for customizing end-user Web pages

For ongoing customization of end-user Web pages, you must know the code locations for those Web pages. The following table summarizes this information:

Table 37. Location of code for various Web pages

Main header (and sample Web page if any)	Location of code in pkiserv.tpl certificate templates file
"1-Year PKI S/MIME Browser Certificate"	TEMPLATE section, CONTENT subsection
"1-Year PKI SSL Browser Certificate" (See Figure 8 on page 205.)	TEMPLATE section, CONTENT subsection
"2-Year PKI Browser Certificate For Authenticating To z/OS"	TEMPLATE section, CONTENT subsection
"2-Year PKI Authenticode - Code Signing Certificate"	TEMPLATE section, CONTENT subsection
"5-Year PKI Intermediate CA Certificate"	TEMPLATE section, CONTENT subsection
"5-Year PKI IPSEC Server (Firewall) Certificate"	TEMPLATE section, CONTENT subsection
"5-Year PKI SSL Server Certificate"	TEMPLATE section, CONTENT subsection
"5-Year SCEP certificate - Preregistration"	TEMPLATE section, CONTENT subsection
"n-Year PKI Certificate for Extensions Demonstration"	TEMPLATE section, CONTENT subsection
"Here's Your Certificate. Cut and Paste it to a File"	INSERT section, -return10cert INSERT. (This is referenced in the RETURNCERT subsection of the TEMPLATE section of each certificate template.)
"Internet Explorer Certificate Install" (See Figure 12 on page 209.)	INSERT section, returnbrowsercertIE INSERT



Table 37. Location of code for various Web pages (continued)

Main header (and sample Web page if any)	Location of code in pkiserv.tmpl certificate templates file
"Preregistration successful"	INSERT section, -preregok INSERT. (This is referenced in the SUCCESSCONTENT subsection of the TEMPLATE section of each certificate template.)
"PKI Services Certificate Generation Application" (See Figure 6 on page 199.)	APPLICATION section, CONTENT subsection
"Renew or Revoke a Browser Certificate" (See Figure 15 on page 212.)	APPLICATION section, RECONTENT subsection
"Request submitted successfully" (For submitting a successful certificate request or renewal, see Figure 10 on page 207.)	<ul style="list-style-type: none"> <li>For a successful certificate request or renewal: INSERT section, -requestok INSERT. (This is referenced in the SUCCESSCONTENT subsection of the TEMPLATE section of the appropriate certificate template.)</li> <li>For a successful certificate revocation: INSERT section, -renewrevokeok INSERT. (This is referenced in the RESUCCESSCONTENT subsection of the APPLICATION section.)</li> </ul>
"Request was not successful"	<ul style="list-style-type: none"> <li>For an unsuccessful certificate request: INSERT section, -requestbad INSERT. (This is referenced in the FAILURECONTENT subsection of the TEMPLATE section of each certificate template.)</li> <li>For an unsuccessful certificate revocation request: INSERT section, -renewrevokebad INSERT. (This is referenced in the REFAILURECONTENT subsection of the APPLICATION section.)</li> </ul>
"Retrieve Your 1-Year PKI S/MIME Browser Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 1-Year PKI SSL Browser Certificate" (See Figure 11 on page 208.)	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 2-Year PKI Browser Certificate For Authenticating To z/OS"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 2-Year PKI Authenticode - Code Signing Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 5-Year PKI Intermediate CA Certificate"	TEMPLATE section RETRIEVECONTENT subsection
"Retrieve Your 5-Year PKI IPSEC Server (Firewall) Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your 5-Year PKI SSL Server Certificate"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your [tmplname]"	TEMPLATE section, RETRIEVECONTENT subsection
"Retrieve Your SAF Browser Certificate 1-Year"	TEMPLATE section, RETRIEVECONTENT subsection

Table 37. Location of code for various Web pages (continued)

Main header (and sample Web page if any)	Location of code in pkiserv.tmpl certificate templates file
"Retrieve Your SAF Server Certificate 1-Year"	TEMPLATE section, RETRIEVECONTENT subsection
"SAF Browser Certificate 1-Year (Auto Approved)"	TEMPLATE section, CONTENT subsection
"SAF Server Certificate 1-Year (Auto Approved)"	TEMPLATE section, CONTENT subsection

**Note:** Fields (such as the Key Usage (KeyUsage) drop down or the Organizational Unit (OrgUnit) text field) are defined in the pkiserv.tmpl certificate templates file, in the INSERT section. (See Table 28 on page 95 for descriptions of the fields.)

---

## Steps for adding a new certificate template

Perform the following steps to add a new certificate template:

1. Review the contents of the eight certificate templates provided with PKI Services to determine the one that most closely approximates the certificate template you want to add.
2. After you have determined the certificate template to use as a model, copy this section in the certificate templates file.
3. Provide a new name, alias, and, if present, nickname for the certificate template.
4. Follow the remaining steps, starting at Step 3 on page 126 in the preceding section.

---

## Changing the runtime user ID

When the PKI Services CGIs are called, they are assigned a runtime user ID. This is the identity that is associated with the unit of work (task). This identity must be authorized to call the function being requested. (See Chapter 16, "RACF administration for PKI Services," on page 241 for more information.) Most of the templates run under the surrogate user ID (PKISERV) for requesting a certificate and for subsequently retrieving it.

There are two exceptions:

- The two SAF templates run under PKISERV for requesting a certificate but run under the client's user ID for certificate retrieval.
- The five-year PKI intermediate CA template runs under the client's user ID for requesting a certificate and for certificate retrieval.

The advantage of having PKISERV as the runtime user ID is that this is the only user ID that needs to be authorized for requesting certificates. The advantage of using the client's user ID is that you have greater control over who can request and

retrieve certificates. For example, you can require the user to authenticate by entering user ID and password before requesting or retrieving a certificate.

You can control the user ID under which a certificate request or retrieval runs by selectively commenting and uncommenting FORM statements in the `pkiserv.tpl` file. (For requesting a certificate, the FORM statements are in the appropriate TEMPLATE section, in the CONTENT subsection. For retrieving a certificate, the FORM statements are in the appropriate TEMPLATE section, in the RETRIEVECONTENT subsection.)

There are three levels of access control for requesting and retrieving certificates:

- Under the client's ID with user ID and password authentication
- Under the surrogate user ID with user ID and password authentication
- Under the surrogate user ID without user ID and password authentication.

Protection directives in the z/OS HTTP Server configuration file (which defaults to `/etc/httpd.conf`) enforce these three levels of access control. The default configuration for PKI Services maps the three levels of access control to the following CGI directories respectively:

- `/PKIServ/ssl-cgi-bin/auth`
- `/PKIServ/ssl-cgi-bin/surrogateauth`
- `/PKIServ/ssl-cgi-bin`

Each of the request and retrieve CGIs reside in all three directories. Thus, when you run a CGI you get the protection established for the directory from which it is called.

Each certificate template contains several FORM statements (two commented out and one uncommented, which is active) that determines which of these applies. You can change the access control by uncommenting one of the FORM statements that is commented out and commenting out the one that is active.

## Steps for changing the runtime user ID for requesting certificates

Perform the following steps to change the runtime user ID for requesting a certificate.

1. In the `pkiserv.tpl` file, find the CONTENT subsection of the TEMPLATE section for the template whose user ID you want to change. Locate the lines containing the FORM statements, such as those in the following example:

### Example:

```
<h3><li>Request a New Certificate
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "/PKIServ/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "/PKIServ/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
<FORM NAME="CertReq" METHOD=POST ACTION=
           "/PKIServ/ssl-cgi-bin/careq.rexx" onSubmit=
```

Notice that the preceding lines contain three FORM statements. The first two FORM statements are commented out, so they are not active. They are for:

- Requesting the certificate under the client's ID and using user ID and password authentication
- Requesting the certificate under the surrogate ID and using user ID and password authentication

The third FORM statement is for requesting the certificate under the surrogate user ID without user ID and password authentication. This is active (it is not commented out).

- 
2. To change the runtime user ID, remove the comment delimiter (#) from in front of the lines for the commented-out FORM statement you want to use and insert the comment delimiter in front of the lines for the bottom FORM statement.
- 

## Steps for changing the runtime user ID for retrieving certificates

Perform the following steps to change the runtime user ID for retrieving a certificate.

1. In the pkiserv.tmpl file, find the RETRIEVECONTENT subsection of the TEMPLATE section for the template whose user ID you want to change. Locate the lines containing the FORM statements, such as those in the following example:

### Example:

```
<H1> Retrieve Your [tmplname]
<H3>Please bookmark this page
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.
```

```
# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#     "/PKIServ/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=
    "/PKIServ/ssl-cgi-bin/cagetcert.rexx" onSubmit=
```

Notice that the preceding lines contain three FORM statements. The first two FORM statements are commented out (they are not active). These are for:

- Retrieving the certificate under the client's ID
- Retrieving it under the surrogate ID, but requiring user ID and password authentication.

The third FORM statement is for retrieving the certificate under the surrogate user ID without user ID and password authentication. This is active (it is not commented out).

- 
2. To change the runtime user ID, remove the comment delimiter (#) from in front of the lines for the commented-out FORM statement you want to use and insert the comment delimiter in front of the lines for the bottom FORM statement.
-

## Customizing e-mail notifications sent to users

You can optionally notify a user by sending an e-mail message when:

- A certificate request is rejected
- A certificate is ready for retrieval
- A certificate is about to expire (unless it has already been renewed or revoked).

Once a day, PKI Services checks the issued certificate list (ICL) for expiring certificates. (The `ExpireWarningTime` parameter (see the **CertPolicy** section in Table 19 on page 51) sets the time interval of how long before the certificate expires that the message is sent.) When PKI Services finds an expiring certificate, it sends an expiration warning message to the client (unless the certificate has already been revoked). Regardless of whether sending the expiration warning message is successful, PKI Services makes only one attempt to send a notification message. If the e-mail address is incorrect or the user renews the certificate and retrieves it before the expiration message is sent, no expiration messages is sent.

If you are not sending e-mail notifications, see Step 6b on page 125 for directions.

If you are sending e-mail notifications, you need to:

- Have copies of the forms in the runtime directory. (For information about copying the message forms to the runtime directory, see Step 2 on page 47.
- Customize the forms. (For details, see “Steps for customizing e-mail notification forms” on page 137.)
- Include the `NotifyEmail` field on certificate requests. This field is already included in the `pkiserv.tmpl` certificate template file. If you are *not* sending e-mail notifications, you need to delete the `NotifyEmail` lines in the `pkiserv.tmpl` file; for details, see Step 6b on page 125.)

For more information about the `NotifyEmail` field, see Table 28 on page 95. For information about fields on request forms, see Table 51 on page 201.

The following examples (of notices you can send to users) are in the sample directory:

```
From:dime-o-cert PKI
Subject:Certificate Ready For Pick Up
```

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%,

Thank you for choosing dime-o-cert PKI. The certificate you requested for subject %dn% is now ready for pickup. Please visit <http://www.dimeocert.com/PKIServ/public-cgi/camain.rexx> to retrieve your certificate. You will need the transaction ID listed below and your passphrase that you entered when you submitted the request.

%%transactionid%

## Customizing the end-user Web pages

From:dime-o-cert PKI  
Subject:Certificate Request Rejected

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. We are sorry to inform you that your certificate request for subject %%dn%% has been rejected. Please contact the PKI Services administrator at 1-800-xxx-xxxx. You will need the transaction ID listed below.

%%transactionid%%

From:dime-o-cert PKI  
Subject:Certificate Expiration

Attention - Please do not reply to this message as it was automatically sent by a service machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. The certificate you requested for subject %%dn%% expires at %%notafter%% local time. If you wish to renew your certificate, please visit <http://www.dimeocert.com/PKIServ/public-cgi/camain.rexx>. If this is a browser certificate, you must use the same workstation and browser that you used when you requested the original certificate. If this is a server certificate, you will have to submit a #10 certificate request.

### Notes:

1. PKI Services automatically provides the To: value in the forms. You can include From: or Subject: or both at the top of the file.
2. You must have a blank line between the Subject and the body of the form.

The following table summarizes the variables you can use in the forms when you customize them. At runtime, PKI Services replaces these with their actual values.

*Table 38. Descriptions of variables for forms*

Variable	Description
%%dn%%	The subject's distinguished name. (This is valid in all the forms.)
%%notafter%%	The certificate expiration date and time in local time in the format YYYY/MM/DD HH:MM:SS. (This is valid only in the expiring.form. It is ignored in the other forms.)
%%requestor%%	The requestor of the certificate. PKI Services obtains this information from the Requestor field the user submits on the original certificate request. (This field is valid in all the forms.)
%%transactionid%%	The transaction ID (CertId) returned. (This is valid for ready and reject forms only. It is ignored in the expiring form.)

The following table summarizes the substitution variables contained in the ready, rejected, and expiring examples:

Table 39. Summary of substitution variables in forms

Referenced substitution variables	readymsg.form	rejectmsg.form	expiring.form
%%dn%%	X	X	X
%%notafter%%	(ignored)	(ignored)	X
%%requestor%%	X	X	X
%%transactionid%%	X	X	(ignored)

## Steps for customizing e-mail notification forms

Perform the following steps to customize the ready, rejected, and expiring forms:

1. Make sure the forms you want to use (readymsg.form, rejectmsg.form, and expiringmsg.form) are present in the runtime directory. (By default, the runtime directory is /etc/pkiserv/. For information about copying files, see Step 2 on page 47.)

---

2. Update the form. At minimum:
  - a. Delete the first four (comment) lines (as shown in the following), so that the first two lines in your file are the From: and Subject: lines:
 

```
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. ...
# Status = HKY...
```
  - b. Specify your company (instead of dime-o-cert) in the From: line and in the first line of the main paragraph
  - c. If appropriate, update the subject.

**Note:** There must be a blank line between the subject and the body of the note.

- d. If you are updating a ready or expiring form, change the URL in the main paragraph to customize it for your company.
- e. If you are updating a reject form, change the telephone number in the main paragraph to customize it for your company.

Make any other needed changes. (You can use variables in the body of the form, but you cannot include %%transactionid%% in the expiring form or %%notafter%% in the ready or reject form.)

3. Save the file.

## Customizing the OtherName field

When you use the OtherName field, you are able to bind additional identities or owner information to the subject of the certificate using the subject alternate name extension. These identities may take different forms, such as employee numbers, customer account numbers, and other identities that you choose to use.



## Customizing the end-user Web pages

The OtherName value is a concatenated string that consists of one or more pairs of OIDs and their associated values. The string is saved in the subject alternate name extension in the certificate.

PKI Services implements the OtherName field as a customizable INSERT called AltOther\_<OID>. The following certificate template in pkiserv.tmp1 is supplied to illustrate the use of the INSERT fields.

Template Name - n-Year PKI Certificate for Extensions Demonstration

The *n*-year PKI certificate template builds a certificate using information provided primarily by users, rather than information that you control. For demonstration purposes, the template builds a certificate that contains all extensions supported by PKI Services. The template contains two sample OtherName fields:

AltOther\_1\_2\_3\_4\_5 Builds one input field.

AltOther\_1\_2\_3\_4\_6 Builds two input fields.

The AltOther\_1\_2\_3\_4\_5 string represents an OtherName field with OID 1.2.3.4.5, an 11-character string that stores a customer account number. The AltOther\_1\_2\_3\_4\_6 string represents an OtherName field with OID 1.2.3.4.6, a 17-character string that stores a 9-digit license number and an expiration date in the *yyyymmdd* format.

When you choose to use the OtherName field to build the subject alternate name extension, you may also want to customize the end-user Web pages to allow end-users to enter the required information using customized input screens that will be easier for them to use. For example, rather than asking a user to enter a string like the one shown below, you can prompt the user to enter a 9-digit license number and its expiration date.

### Example of an OtherName field value:

1.2.3.4.6,12345678920050215

## Steps for customizing the sample AltOther\_<OID>INSERTs

### Before you begin:

- Decide what identifiers you wish to add to the Subject Name Alternate extension.
- Select the registered OID value to use to represent your data string. Check the appropriate standards organization (ISO or ITU). If not already registered, register your own OID.
- Select which certificate templates you will update to add the Subject Name Alternate extension.
- Decide whether to use a sample INSERT for your AltOther\_<OID> INSERT or create your own INSERT. The sample INSERT called AltOther\_1\_2\_3\_4\_5 demonstrates using one input field. The sample insert called AltOther\_1\_2\_3\_4\_6 demonstrates using two input fields.
- Determine the following values you will use to customize your INSERT.
  - The OID value for your OtherName field
  - The name and length for each input field.
- Review Figure 3 on page 140. It contains a listing of the sample INSERT called AltOther\_1\_2\_3\_4\_6 which demonstrates using two input fields. The lines you are most likely to customize are marked in Figure 3 on page 140. The following steps refer to the marked lines.

Perform the following steps to customize the AltOther\_<OID> INSERT using Figure 3 on page 140 as a reference.



1. Change the OID value `_1_2_3_4_6` to the OID value you need in the line marked **1** and in all other lines in the sample INSERT. For example, if you chose OID `2.16.76.1.3.1` for your OtherName field, change all occurrences of `AltOther_1_2_3_4_6` to `AltOther_2_16_76_1_3_1`.

---
2. Customize the first input field description in the line marked **2** to prompt users of your Web page. For example, change Customer's driver license number (9 digits) to Enter your member card number.

---
3. Customize the first INPUT field name "Other2a" to your value in the line marked **3** and in all other lines in the sample INSERT. For example, change all occurrences of "Other2a" to "MemNum". Also, customize SIZE and maxlength as needed.

---
4. Customize the next input field description in the line marked **4** to prompt users of your Web page. For example, change Customer's driver license expiration date (yyyymmdd) to Enter your date of birth (yyyymmdd).

---
5. Customize the next INPUT field name "Other2b" to your value in the line marked **5** and in all other lines in the sample INSERT. For example, change all occurrences of "Other2b" to "Birthdate". Also, customize SIZE and maxlength as needed.

---
6. Customize the starting positions and lengths for each input field value in the lines marked **6** and **7**. For example, if the member card number is an 11-digit number, change  
`form.Other2a.value=form.altrawstring_1_2_3_4_6.value.substr(0,9)` to  
`form.MemNum.value=form.altrawstring_2_16_76_1_3_1.value.substr(0,11)`.

---
7. Customize the validation script that begins with the line marked **8**.

---
8. Change the OID value `1.2.3.4.6` to the OID value you need in the line marked **9**. For example, if you chose OID `2.16.76.1.3.1` for your OtherName field, change `1.2.3.4.6` to `2.16.76.1.3.1`.

---
9. Repeat steps 2 through 8 for each additional input field you need.

---

```

=====
# Sample AltOther INSERT with two input fields
# =====
<INSERT NAME=AltOther_1_2_3_4_6> 1
<INPUT NAME="AltOther_1_2_3_4_6" TYPE="hidden" maxLength="255">

<p> Other Name for alternate name: <BR>
<p> Customer's driver license number (9 digits) [optfield] <BR> 2
<INPUT NAME="Other2a" TYPE="text" SIZE=9 maxLength="9" [readonly]> 3
<p> Customer's driver license expiration date (yyyymmdd) [optfield] <BR> 4
<INPUT NAME="Other2b" TYPE="text" SIZE=8 maxLength="8" [readonly]> 5

<INPUT NAME="altrawstring_1_2_3_4_6" TYPE="hidden" VALUE="[altrawvalue]">

<SCRIPT LANGUAGE="JavaScript">
<!--
//This is the script that will be called at load time.
var form=document.forms[0]
if (form.altrawstring_1_2_3_4_6.value.length > 0) {
    //The name 'Otherx' needs to match with the above INPUT NAME.
    //Substr(start position, length)
    form.Other2a.value=form.altrawstring_1_2_3_4_6.value.substr(0,9) 6
    form.Other2b.value=form.altrawstring_1_2_3_4_6.value.substr(9,8) 7
}
//-->
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript"> 8
<!--
//This is the validation script
function ValidAltOther_1_2_3_4_6(frm){
    if (("["optfield]" == "" && frm.Other2a.value.length != 9) ||
        ("["optfield]" != "" && frm.Other2a.value != "" && frm.Other2a.value.length != 9)) {
        alert("Enter 9 digit license number.");
        frm.Other2a.focus();
        return false;
    }
    :
    //Build the entire AltOther field.
    if (frm.Other2a.value != "" && frm.Other2b.value != "")
        frm.AltOther_1_2_3_4_6.value = "1.2.3.4.6," + frm.Other2a.value + 9
        frm.Other2b.value;
    else
        frm.AltOther_1_2_3_4_6.value = "";
    return true;
}
//-->
</SCRIPT>
</INSERT>

```

Figure 3. Partial listing of the AltOther\_1\_2\_3\_4\_6 sample INSERT showing the lines you are most likely to customize

---

## Chapter 12. Customizing the administration Web pages

---

### CGIs for administration Web pages

CGIs for administration Web pages are execs that gain control when the user clicks an action button and render the Web pages dynamically. All of the administration CGIs are contained in the `/usr/lpp/pkiserv/PKIServ/ssl-cgi-bin/auth` directory.

The following table (which lists the CGI execs in logical order) summarizes the actions they perform:

*Table 40. CGI actions for administrative Web pages*

CGI exec	Action	Sample Web page
<code>admain.rexx</code>	This displays the administration home page. The main heading is “PKI Services Administration”. This Web page lets the administrator work with a single certificate request or certificate or search for certificate requests or certificates.	See Figure 21 on page 222.
<code>admpend.rexx</code>	On the administration home page, the administrator can search for certificate requests. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none"><li>• “Certificate Requests” Web page—This lists certificate requests matching the criteria and allows the administrator to process the certificate request(s).</li><li>• “Processing was not successful” Web page</li></ul>	For an example of the “Certificate Requests” Web page, see Figure 26 on page 228.
<code>admpendtid.rexx</code>	On the administration home page, the administrator can enter a transaction ID to work with a single certificate request. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none"><li>• “Single Request”—This lists the certificate request that matches the transaction ID and allows the administrator to process that certificate request.</li><li>• “Processing was not successful”</li></ul>	For an example of the “Single Request” Web page, see Figure 22 on page 223.
<code>admodtid.rexx</code>	This displays the “Modify and Approve Request” Web page that appears when the administrator decides to modify a request before approving it (on the “Single Request” Web page).	See Figure 25 on page 225.
<code>admicl.rexx</code>	On the administration home page, the administrator can search for certificates. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none"><li>• “Issued Certificates”—This lists the certificate(s) that match the search criteria and allows the administrator to revoke or delete selected certificate(s).</li><li>• “Processing was not successful”</li></ul>	For a sample of the “Issued Certificates” Web page, see Figure 26 on page 228.
<code>admiclcert.rexx</code>	On the administration home page, the administrator can enter a serial number to work with a single certificate. This displays a Web page whose main heading is one of the following: <ul style="list-style-type: none"><li>• “Single Issued Certificate”—This lists the certificate that matches the serial number ID and allows the administrator to revoke or delete that certificate.</li><li>• “Processing was not successful”</li></ul>	For a sample of the “Single Issued Certificate” Web page, see Figure 30 on page 233.

## Customizing the administration Web pages

Table 40. CGI actions for administrative Web pages (continued)

CGI exec	Action	Sample Web page
admacttid.rexx	Displays a Web page after the administrator processes a single certificate request (approving it with or without modifications, rejecting, or deleting it). This Web page has one of the following as its main heading: <ul style="list-style-type: none"> <li>• “Processing successful”</li> <li>• “Processing was not successful”</li> </ul>	For a sample of the Web page whose main heading is “Processing successful”, see Figure 23 on page 224.
admacttid2.rexx	This displays a Web page after the administrator approves a certificate request with modifications. The Web page has one of the following main headings: <ul style="list-style-type: none"> <li>• “Processing successful”</li> <li>• “Processing was not successful”</li> </ul>	For a sample of the Web page whose main heading is “Processing successful”, see Figure 23 on page 224.
admpendall.rexx	After the administrator searches for certificate requests and admpend.rexx displays the results, the administrator clicks a button to approve, reject, or delete selected certificate requests. This calls admpendall.rexx, whose main heading is one of the following: <ul style="list-style-type: none"> <li>• “Processing successful” if the action was successful</li> <li>• “Processing was not successful” if the action failed (for example, if the administrator tried to delete certificate requests that were already deleted)</li> <li>• “Processing partially successful” if not all of the selected requests are processed successfully</li> </ul>	<ul style="list-style-type: none"> <li>• For an example of the “Processing successful” Web page, see Figure 27 on page 230.</li> <li>• For an example of the “Processing was not successful” Web page, see Figure 28 on page 230.</li> <li>• For an example of the “Processing partially successful” Web page, see Figure 29 on page 231.</li> </ul>
admactcert.rexx	Displays a Web page after the administrator tries to revoke or delete one or more selected certificates. The Web page has one of the following main headings: <ul style="list-style-type: none"> <li>• “Processing successful”</li> <li>• “Processing was not successful”</li> </ul>	—
admiclall.rexx	After the administrator searches for certificates and admicl.rexx displays the results, the administrator clicks a button to revoke or delete selected certificates. This calls admiclall.rexx, which displays a Web page whose main heading is one of the following: <ul style="list-style-type: none"> <li>• “Processing successful” if the action was successful</li> <li>• “Processing was not successful” if the action failed</li> <li>• “Processing partially successful” if not all of the selected certificates are processed successfully</li> </ul>	—

---

## Customizing the administration Web pages

The administration Web pages are not as customizable as the end-user Web pages. You can customize page headers, footers, frames, links, colors, and so forth, but you cannot change internal Web page content. Except for identifying the fields that an administrator can change when approving certificate requests, the administration Web page logic is fixed.

However, you can make changes in the following subsections in the PKISERV APPLICATION section of the `pkiserv.tmp1` certificate template file. (These subsections appear in the application section of PKISERV only.)

<b>ADMINHEADER</b>	Contains the general installation-specific HTML content for the header of all the administration Web pages.
<b>ADMINFOOTER</b>	Contains the general installation-specific HTML content for the footer of all the administration pages.
<b>ADMINSCOPE</b>	This optional subsection allows the administrator to choose a different CA domain. For more information, see “Adding a new CA domain” on page 161.

---

## Steps for customizing the administration Web pages

Perform the following steps to customize the administration Web pages:

1. Add any desired Web page header for the administration pages to the ADMINHEADER subsection of the PKISERV APPLICATION section. (The ADMINHEADER subsection is near the end of the APPLICATION section.)

**Example:**

```
<ADMINHEADER>
<HTML><HEAD>
<TITLE>Web-Based Certificate Generation Administration</TITLE></HEAD>
<BODY>
</ADMINHEADER>
```

2. Add any desired Web page footer for the administration pages to the ADMINFOOTER subsection of the APPLICATION section. (The ADMINFOOTER subsection is near the end of the APPLICATION section.)

**Example:**

```
<ADMINFOOTER>
<p> email: webmaster@company.com
</BODY>
</HTML>
</ADMINFOOTER>
```

---

## Changing the runtime behavior for accessing administration pages

When the administrator tries to access the administration pages (by clicking the **Go to administration page** button on the PKI Services home page), access to the administration pages is controlled in one of the following ways:

## Customizing the administration Web pages

- A popup window appears, requiring the administrator to enter a user name and password. (See Figure 20 on page 219 for a sample of the authentication popup window.)
- Alternately, the administrator may have to authenticate by using a previously issued browser certificate. In other words, the administrator would need to have a certificate before visiting the administration Web pages.

By default, the first method is used. However, you can change the runtime behavior so that the second method is used instead. If you decide to use the second method, anyone intending to become a PKI Services administrator needs to request and retrieve a one-year PKI browser certificate for authenticating to z/OS before trying to access the administration pages.

**Note:** The one-year PKI browser certificate for authenticating to z/OS contains a HostIdMappings extension. (For more information, see Chapter 16, “RACF administration for PKI Services,” on page 241.)

## Steps for changing control of access to administration pages

Perform the following steps to change the access control of the administration pages to require authenticating by using a certificate:

1. Edit the `pkiserv.tmpl` certificate templates file and find the following lines in the PKISERV APPLICATION section:

```
# The following action will force userid/pw authentication for administrators
<FORM name=admform METHOD=GET ACTION="/PKIServ/ssl-cgi/auth/admain.rexx">
# The following action will force client certificate authentication
# for administrators
#<FORM name=admform METHOD=GET
# ACTION="/PKIServ/clientauth-cgi/auth/admain.rexx">
<p>
<INPUT TYPE="submit" VALUE="Go to Admin Pages">
</FORM>
```

The first FORM statement in these lines is active. (It is not commented out with # characters in front of the lines.) This requires authentication by entering the user name and password in a popup window. The second FORM statement is commented out (using # characters). This requires authentication by using a previously issued browser certificate.

2. Comment out the first FORM statement (add # characters in front of the FORM and ACTION lines) and uncomment the second FORM statement (removing the # characters in front of the FORM and ACTION lines).

---

## Chapter 13. Advanced customization

This chapter describes the advanced customization procedures available for PKI Services. All are optional.

- “Scaling for high volume installations”
- “Using certificate policies” on page 146
- “Updating the signature algorithm” on page 149
- “Customizing distribution point CRLs” on page 150
- “Creating a distribution point ARL” on page 156
- “Adding an application domain” on page 158
- “Adding a new CA domain” on page 161
- “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175
- “Using the PKI exit” on page 180

---

### Scaling for high volume installations

Some PKI Services installations manage a large number of certificates and certificates requests. The following guidelines can help you scale your system to maintain high performance in a high volume environment.

#### Guidelines:

1. Use distribution point CRLs if you will average more than 500 revoked non-expired certificates at any given time. For more information, see “Customizing distribution point CRLs” on page 150.
2. If you anticipate having a large number of certificate requests pending approval at any given time, implement a PKI exit to automate the approval process. (For more information, see “Using the PKI exit” on page 180.) This need arises from the human limitation rather than a technical one because it becomes nearly impossible to manually approve the requests when the volume grows too high.
3. To prevent name collisions in the LDAP directory, ensure that the subject distinguished names are unique. This can either be done by implementing a PKI exit to supply a unique name, or by enforcing the use of the MAIL= distinguished name attribute where you require the e-mail address to be unique.
4. Queries against the request or ICL database may time out if the database contains a large number of records. The performance of the query can be vastly improved by supplying the requestor’s name as additional search criteria if the saved requestor data is meaningful to your organization and it is recallable. In this case, a PKI exit can be used to supply a meaningful value, such as a Lotus® Notes short name or customer account number.
5. Keep the size of the request and ICL databases small by quickly removing records that are no longer needed. This can be done by setting low values for the following fields in the **ObjectStore** section of the PKI Services configuration file (pkiserv.conf):
  - RemoveCompletedReqs
  - RemoveInactiveReqs
  - RemoveExpiredCerts

## Using certificate policies

Certificates can contain a CertificatePolicies extension. This extension contains policy information, such as the way in which your CA operates and the intended purpose of the issued certificates. (For more information about this extension, see RFC 2459: *Certificate and CRL Profile*.)

The CertificatePolicies extension contains one or more PolicyInformation sequences. (Typical usage has just one of these.) The PolicyInformation sequence has the following format:

- Your Policy OID as registered with the appropriate standards organization (ISO or ITU)
- Zero or more PolicyQualifiers sequences, each having the following information:
  - Either a Certificate Practices Statement (CPS) URI
  - Or a UserNotice sequence, which consists of one or both of the following:
    - A notice (text string) intended to be viewed by customers using the certificate such as copyright or other legal information
    - Your organization's legal name (text string) with one or more notice numbers defined elsewhere, perhaps in your CPS.

By default, PKI Services does not include this extension in the certificates it creates. However, you can define your own CertificatePolicies extension by modifying fields in the **CertPolicy** section of the pkiserv.conf configuration file. You can also specify the PolicyRequired value to indicate whether CertificatePolicies extensions should be created for all certificate templates on a global basis or whether it will be individually created based on the specifications of each certificate template.

### **PolicyRequired=T**

Indicates that the CertificatePolicies extension will be created on a global basis with the same values for all certificate templates based on the specifications in the **CertPolicy** section of the PKI Services configuration file. Any policies specified within the template will be ignored.

See “Steps for creating the CertificatePolicies extension on a global basis” on page 147.

### **PolicyRequired=F (default)**

Indicates that the CertificatePolicies extension will be optionally created based on the specifications in the CONSTANT section for each individual certificate template. Policies specified within the template will be used.

See “Steps for creating the CertificatePolicies extension on a template basis” on page 148.

**Note:** PolicyCritical is ignored unless PolicyRequired=T. When PolicyRequired=F, setting %%Critical=CertPolicies%% in the CONSTANT section of the template will mark the extension critical.

**Restriction:** When policies are specified within an individual template, the policy data is saved with the request at the time the request is submitted or modified. Therefore, if PKI Services is stopped and restarted to make changes in the policy data before the certificate is issued, the changes will not be reflected in the issued certificate. However, the PolicyRequired=F setting is checked at the time the



certificate is issued. Therefore, if PKI Services is stopped and restarted to make changes to the `PolicyRequired` setting before the certificate is issued, the new setting will be used to determine which policy information is used (the global policy data or the data saved with the request.)

## Steps for creating the CertificatePolicies extension on a global basis

Perform the following steps to create your own CertificatePolicies extension on a global basis:

1. Edit the `pkiserv.conf` configuration file and find the **CertPolicy** section.

---
2. Change the value of `PolicyRequired` to T (True) as in the following line:  
`PolicyRequired=T`

---
3. If you want to have the extension marked critical (this is not suggested), set the `PolicyCritical` equal to T (True) as in the following line:  
`PolicyCritical=T`

---

4. Go to the **OIDs** section of the `pkiserv.conf` configuration file. By default (as shown in the following example), the name is `MyPolicy=1.2.3.4` and value is 1.2.3.4. The value of `MyPolicy` should be an installation-specific (registered) Object ID identifying your organization's certificate. Replace the value of `MyPolicy` in the following line with your Object ID.

**Example:**

```
[OIDs]
MyPolicy=1.2.3.4
```

Optionally, change the parameter name `MyPolicy` to your own installation-specific name. If you change the parameter name in this step, make a note of it. You need it for the next step. You can repeat the `MyPolicy` parameter using unique names and values if you need to define multiple policies.

**Example:**

```
MyPolicy=1.2.3.4
MyOtherPolicy=2.3.4.5
```

5. If you changed the parameter name `MyPolicy` in the previous step, go back to the **CertPolicy** section and update the `PolicyName1` line to change the `MyPolicy` parameter to the policy name you specified in the **OIDs** section:  
`[CertPolicy]`  
`PolicyName1=MyPolicy`

---

6. If you want to add qualifiers, perform the following steps:
  - a. Update the `Policy1Org` and `Policy1Notice1` fields in the following example:  

```
Policy1Org=My Company, Inc
Policy1Notice1=1
```

**Policy1Org**      Your organization's name, for example, International Business Machines, Inc.

### Policy1Notice1 through Policy1Noticen

Your notice numbers. (You may need more than one Policy1Noticen line, depending on how many notice numbers you have. Repeat the line as needed, by incrementing the suffix number on the keyword, for example Policy1Notice1, Policy1Notice2, and so forth.)

- b. Change the value of the UserNoticeText1 line shown in the following. The *statement* should be your notice text string, for example, Certificate for IBM internal use only.

```
UserNoticeText1=statement
```

- c. Change the value of the CPS1 line shown in the following. The value should be your CPS URI, for example, <http://www.ibm.com/cps.html>.

```
CPS1=http://www.mycompany.com/cps.html
```

If you do not want to add qualifiers, delete or comment out (by inserting a # character at the start of the line) the preceding lines.

- 
7. If you need multiple qualifiers, repeat the following fields as needed, incrementing the suffix numbers, for example:

```
PolicyName2=MyOtherPolicy
Policy2Org=International Business Machines, Inc.
Policy2Notice1=5
Policy2Notice2=9
UserNoticeText2=Certificate is intended for testing only
CPS2=http://www.ibm.com/cps2.html
```

- 
8. If you made any changes to the PKI Services configuration, stop and restart PKI Services to activate the changes.
- 

## Steps for creating the CertificatePolicies extension on a template basis

Perform the following steps to create your own CertificatePolicies extension on an individual template basis:

1. Edit the `pkiserv.conf` configuration file and find the **CertPolicy** section.

---

2. Change the value of `PolicyRequired` to F (False) as in the following line:  

```
PolicyRequired=F
```

---

3. Follow steps 4 through 7 in “Steps for creating the CertificatePolicies extension on a global basis” on page 147 to create the individual policies you need.

---

4. Edit `pkiserv.tpl` and customize the **CONSTANT** subsection under the certificate template for which you need CertificatePolicies extensions.  
For example, if you have specified values for `PolicyName1`, `PolicyName3`, and `Place-name` in `pkiserv.conf`, then you may specify the certificate policies in `pkiserv.tpl` in the following ways:

```
%%CertPolicies=3%%
or
%%CertPolicies=3 6%%
or
%%CertPolicies=1 3 6%%
```

**Rule:** The policy numbers in the `pkiserv.tmp1` file must exist in the `pkiserv.conf` file. For each template, you can choose a different subset of these numbers.

If you want to make the `CertPolicies` extension critical, specify the following in the **CONSTANT** section:

```
%%Critical=CertPolicies%%
```

- 
5. If you made any changes to the PKI Services configuration, stop and restart PKI Services to activate the changes.
- 

## Updating the signature algorithm

By default, PKI Services uses the SHA1 with RSA encryption signature algorithm for signing certificates. If you need to use DSA or one of the older RSA algorithms, you can change the `SigAlg1` value in the **CertPolicy** section of the `pkiserv.conf` configuration file.

The maximum key length for the CA signing private key depends on the algorithm you use.

Key type	Maximum key length
RSA key	2048 bits
DSA key	1024 bits

If the certificate key type is RSA, specify the `SigAlg1` algorithm value as one of the following:

```
sha-1WithRSAEncryption (OID value 1.2.840.113549.1.1.5)
md-5WithRSAEncryption (OID value 1.2.840.113549.1.1.4)
md-2WithRSAEncryption (OID value 1.2.840.113549.1.1.2)
```

If the certificate key type is DSA, specify the `SigAlg1` algorithm value as follows:

```
id-dsa-with-sha1 (OID value 1.2.840.10040.4.3)
```

## Steps for changing the signature algorithm

**Before you begin:** Change the signing algorithm *before* you create any certificate requests. If changing the signing algorithm *after* some certificates requests have been created, you must wait until all requests are approved and the certificates created, or else you must add a `SigAlg2=old-signing-algorithm` line to the **CertPolicy** section. If you take this second option, `SigAlg1` becomes the signature algorithm for new requests.

Perform the following steps to change the signature algorithm:

1. Edit the `pkiserv.conf` configuration file and find the **OIDs** section.
- 
2. If you want to change from SHA1 encryption to MD5, add the following line:

```
md-5WithRSAEncryption=1.2.840.113549.1.1.4
```

Otherwise, to change to MD2, add the following line:

```
md-2WithRSAEncryption=1.2.840.113549.1.1.2
```

---

3. Find the **CertPolicy** section.
- 

4. If you want to change from SHA1 encryption to MD5, change sha-1WithRSAEncryption in the following line to md-5WithRSAEncryption. If you want to change to MD2, change sha-1WithRSAEncryption to md-2WithRSAEncryption.

```
SigAlg1=sha-1WithRSAEncryption
```

---

---

## Customizing distribution point CRLs

If your PKI Services installation is very active, many certificates can be in the revoked state at any one time. Therefore, the certificate revocation list (CRL) can become quite large, causing considerable network traffic and overhead to an application wishing to process it. Publishing partial CRLs to multiple distribution point (DP) CRLs is a way of keeping your CRLs small.

**Guideline:** Consider using distribution point CRLs if you anticipate averaging more than 500 revoked non-expired certificates at any given time.

You begin using distribution point CRLs when you accept the defaults settings contained in PKI Services configuration file (`pkiserv.conf`). You can customize those settings by specifying the number of certificates per DP CRL and by specifying the name of the DP CRL using the following two parameters in the **CertPolicy** section of the `pkiserv.conf`:

<code>CRLDistSize</code>	Specifies the maximum number of certificates to be managed by a single DP. This represents the number of entries in each DP CRL if all active certificates are revoked at once.
<code>CRLDistName</code>	Specifies the file name, or the constant portion of the leaf-node RDN, for the CRL distribution point.

You can choose to further customize your DP CRL processing to build the URI format name for the distribution point in the `CRLDistributionPoints` extension of each certificate. This allows your certificate validation programs to dynamically retrieve a CRL without being preconfigured with LDAP bind information. However, because bind credentials cannot be added to DP CRLs with URI format names, anonymous access is used to retrieve the CRL.

The URI format name is built in *addition* to the LDAP distinguished name of the DP CRL that is always added when `CRLDistSize` is greater than zero. You can add the URI format name by customizing the following two parameters in the **CertPolicy** section of the `pkiserv.conf`:

<code>CRLDistURIn</code>	Specifies the name for the DP CRL in the form of a URI that adds the protocol type and the server domain name.
<code>CRLDistDirPath</code>	Specifies the full path for the file system directory where PKI Services will save each DP CRL.

You can also choose to have PKI Services create a `CRLDistributionPoints` extension for each CA certificate in addition to non-CA certificates. You choose this by customizing the `ARLDist` parameter in the **CertPolicy** section of the `pkiserv.conf`. This creates a distribution-point authority-revocation list (DP ARL) for your CA certificates. See “Creating a distribution point ARL” on page 156 for details.

## Specifying the URI format

When you choose to use distribution points for CRL and ARL processing, PKI Services updates the `CRLDistributionPoints` extension with the distinguished name for the LDAP entry where the distribution point is posted. You can choose to add another name to the extension in the URI format which contains the protocol type and the server domain name in addition to the distinguished name. With the URI format, the location of the distribution point is self-contained in the `CRLDistributionPoints` extension.

The URI format contains the following information:

- the protocol type (LDAP or HTTP)
- the server domain name
- if the protocol is LDAP:
  - the distinguished name of the distribution point
  - for non-CA certificates, the attribute string `?certificateRevocationList`
  - for CA certificates, the attribute string `?authorityRevocationList`
- if the protocol is HTTP, the virtual or real pathname, ending with the file name—formed from the common name portion of the distinguished name of the distribution point with the `.crl` extension—where the distribution point CRL is stored.

### Examples:

```
ldap://ldap.bankxyz.com:389/CN=CRLlist1,OU=Bank XYZ
      Authority,O=Bank XYZ,C=US?certificateRevocationList
```

```
http://www.bankxyz.com/PKIServ/cacerts/CRLlist1.crl
```

**Restriction:** Special characters, such as spaces, quotation marks, and square brackets are not considered *safe* to use in URLs and should be encoded using the appropriate *escape* sequence. For details, see RFC 1738: *Uniform Resource Locators (URL)*.

## Determining CRLDistURI*n*

If you are using DP CRLs (you specified a `CRLDistSize` value greater than 1 in the **CertPolicy** section of `pkiserv.conf`), you can choose to further customize your DP CRL processing to build the URI format name for the DP CRL in the `CRLDistributionPoints` extension of each certificate. The URI format name is built in *addition* to the LDAP distinguished name of the DP CRL, as described in “Specifying the URI format.”

This is an optional parameter. If you do not specify a `CRLDistURIn` value, the URI format name will not be created. You can specify multiple entries for the `CRLDistURIn` parameter, using the parameters `CRLDistURI1`, `CRLDistURI2`, and so forth. This value is ignored if you did not specify `CRLDistSize` with a value greater than zero. The URI format will not be created if you specify `CRLDistURIn` with an *n* value of 0.

There are different ways to specify the value of `CRLDistURIn` for different protocols. **Valid values** include *one* of the following:

- A string that begins with the characters `http://` or `ldap://`
- A string that consists of `LdapServer $n$` , where  $n$  is greater than zero.

**Restriction:** PKI Services provides syntax checking based only on valid values for the `CRLDistURI $n$`  value. You must ensure that the URIs you choose can be accessed.

### Specifying an HTTP URI

For HTTP, specify the complete URL but do not specify the name of a file where the CRL DP is stored. The value for `CRLDistURI $n$`  may be specified with or without a trailing slash.

#### Example:

```
CRLDistURI1=http://www.bankxyz.com/PKIServ/cacerts/
```

### Specifying an LDAP URI

For LDAP, there are two ways to indicate the `CRLDistURI $n$`  value. Choose either of the following two methods:

- Specify the protocol and the domain name (and the port, if needed). The value for `CRLDistURI $n$`  may be specified with or without a trailing slash.

#### Example:

```
CRLDistURI1=ldap://ldap.bankxyz.com:389/
```

- Specify the keyword `LdapServer $n$`  to have PKI Services build the `CRLDistURI $n$`  value for you based on a server identified by the `Server $n$`  or `BindProfile $n$`  directives in the **LDAP** section of `pkiserv.conf`.

#### Example:

```
CRLDistURI3=LdapServer1
```

This example assumes that the first server specified in the **LDAP** section was similarly defined as one of the following examples:

#### Examples:

```
Server1=ldap.bankxyz.com:389
```

*or*

```
BindProfile1=LOCALPKI.BINDINFO.LDAP1
```

#### Rules for using the `LdapServer $n$` keyword:

1. You must have specified a value greater than zero for `NumServers` in the **LDAP** section of `pkiserv.conf`.
2. Each server represented by the  $n$  value in the `LdapServer $n$`  keyword must be identified in *one* of the following ways:
  - The corresponding LDAP server must be identified by a `Server $n$`  or `BindProfile $n$`  value in the **LDAP** section of `pkiserv.conf`, *or*
  - The corresponding LDAP server must be identified in the default FACILITY class profile `IRR.PROXY.DEFAULTS` and must follow the same identification requirements for PKI Services LDAP processing. See “Using encrypted passwords for LDAP servers” on page 260.

## Determining CRLDistDirPath

If the protocol for the URI you specified with `CRLDistURI $n$`  is HTTP protocol, you need to also determine your value for the `CRLDistDirPath` parameter. The `CRLDistDirPath` parameter specifies the full path of the `var` directory where PKI Services will save each DP CRL. The value can be specified with or without the

trailing slash. The default value is `/var/pkiserv/`. See “Specifying the URI format” on page 151. This value is ignored if you do not create a `CRLDistributionPoints` extension or if the URI protocol is LDAP.

## Steps for customizing distribution point CRLs

**Before you begin:** Before executing this procedure, be aware of the following restrictions:

- If running PKI Services in a sysplex, all instances of PKI Services must specify the same values for each of the above parameters.
- Once a value for `CRLDistName` has been set, it must not be changed or removed from the configuration file.
- Once a non-zero value has been set for `CRLDistSize`, it must not be changed back to zero or removed from the configuration file. Adjusting the value is acceptable.

Perform the following steps to customize distribution point CRLs:

1. Determine your value for the `CRLDistSize` parameter based on the following algorithm. The default value specified in `pkiserv.conf` is 500. Your value should be based on your desired average number of CRL entries per distribution point and your estimated revoked-certificate percentage as expressed by the following formula:

$$\text{CRLDistSize} = E \div P$$

where:

**E** is the desired average number of CRL entries per distribution point.

**P** is the estimated revoked-certificate percentage.

**Example:** If you estimate that 10% of the non-expired certificates will be in the revoked state at any given time and you wish the CRLs to average around 100 entries each, then:

$$\text{CRLDistSize} = 100 \div 0.10 = 1000$$

The `CRLDistSize` in bytes can be roughly estimated to be  $500 + (25 \times \text{number of CRL entries})$ . Using the example above, the average CRL size in bytes would be  $500 + (25 \times 100) = 3000$  bytes.

**Restriction:** A single CRL can not exceed 32K bytes in length. Therefore, you must limit its length. In addition, the longer the CRL, the longer it will take to process it.

### Rules:

- a. The value of `CRLDistSize` is a numeric value from 0–2147483647.
- b. A non-zero value indicates that distribution point (DP) CRLs will be created.
- c. A value of zero (the default) indicates that DP CRL processing will not be performed.

**Guideline:** If you anticipate a low revocation rate for active certificates, use a value of **0**. Your installation may not need to use distribution point CRLs and the global CRL may be sufficient.

2. If necessary, update the value of `CRLDistSize` in the **CertPolicy** section of `pkiserv.conf` to the customized value you determined in Step 1.  
If you selected the **0** value for `CRLDistSize`, complete this step and then continue with Step 12.



3. Determine your value for the `CRLDistName` parameter. The default value is `CRL`. The common name portion of the distinguished name of each DP CRL is formed by appending the DP number to this value. The CA's name is also appended. (See "How DP CRLs are published" on page 155.)

**Example:**

`CN=CRL3,OU=My Company Certificate Authority,O=My Company,C=US`

**Restrictions:**

- a. The value of `CRLDistName` must contain only alphanumeric characters.
- b. The length of the entire DP distinguished name should not exceed 255 bytes. (DP distinguished names that are longer will appear truncated in the PKIDPUBR audit record.)

- 
4. If necessary, update the value of `CRLDistName` in the **CertPolicy** section of `pkiserv.conf` to your customized value.

- 
5. Optionally, determine your value for the `CRLDistURI` parameter. Specifying this value will allow PKI Services to build a URI-formatted name for the DP CRL in each `CRLDistributionPoints` extension, if you also specified a `CRLDistSize` value greater than 1 in Step 2 on page 153. The URI format name is built in *addition* to the LDAP distinguished name of the DP CRL in each `CRLDistributionPoints` extension. If you do not specify a `CRLDistURI` value, the URI format name will not be created. See "Specifying the URI format" on page 151.

You can specify multiple entries for the `CRLDistURI` parameter, using the parameters `CRLDistURI1`, `CRLDistURI2`, and so forth.

- 
6. If necessary, update the value of `CRLDistURI` in the **CertPolicy** section of `pkiserv.conf` to your customized value or values.

- 
7. If the protocol for the URI you specified with `CRLDistURI` in Step 5 is LDAP protocol, skip to Step 12.

- 
8. If the protocol for the URI you specified with `CRLDistURI` in Step 5 is HTTP protocol, determine your value for the `CRLDistDirPath` parameter.

The `CRLDistDirPath` parameter specifies the full path of the var directory where PKI Services will save each DP CRL. The default value is `/var/pkiserv/`. The value can be specified with or without the trailing slash. See "Specifying the URI format" on page 151.

- 
9. If necessary, update the value of `CRLDistDirPath` in the **CertPolicy** section of `pkiserv.conf` to your customized value.

- 
10. Optionally, determine your value for the `ARLDist` parameter. Specifying this parameter creates a distribution point ARL so you can check revocation status for CA certificates without accessing the global ARL. See "Creating a distribution point ARL" on page 156.
-



11. If necessary, update the value of `ARLDist` in the **CertPolicy** section of `pkiserv.conf` to your customized value.
- 
12. If you made any updates to `pkiserv.conf`, stop and restart PKI Services to make your changes effective.
- 

**When you have finished:** If you selected a `CRLDistSize` value greater than zero, you have set up distribution point CRLs. Now, created certificates will contain the `CRLDistributionPoints` extension indicating the location of the DP CRL that will be checked for revocation information. If you specified a URI-formatted name with `CRLDistURI`, now your `CRLDistributionPoints` extensions will also contain a URI name for each DP CRL, containing the protocol type and server domain name. If you enabled the `ARLDist` option, you have set up a distribution point ARL for CA certificates.

## How distribution point CRLs work

PKI Services always creates a *global* CRL regardless of whether or not you choose to use DP CRLs. The global CRL contains revocation information for certificates that have no `CRLDistributionPoints` extension (in other words, certificates defined with `CRLDistSIZE=0`). When a certificate contains a `CRLDistributionPoints` extension, PKI Services publishes its revocation status to the appropriate DP CRL, not in the global CRL.

The following topics will help you understand more about how DP CRLs work. This information will be useful if you write applications that process CRLs.

### How DP CRLs are published

DP CRLs are published to LDAP at leaf nodes directly below the CA's entry. For example, if the CA's name is:

```
OU=My Company Certificate Authority,O=My Company,C=US
```

Then, the DP CRLs would be published to:

```
CN=DP-name,OU=My Company Certificate Authority,O=My Company,C=US
```

### How DP CRLs are partitioned

The partitioning of the overall CRL into partial CRLs is based on certificate serial number and the value of `CRLDistSize` in `pkiserv.conf`. For example, if `CRLDistSize` is 100 and `CRLDistName` is ABC, then certificates with serial numbers 1–100 will appear on DP ABC1; 101–200 on DP ABC2, and so on. PKI Services dynamically creates DP CRLs as needed as a part of certificate issuance. Existing DP CRLs are refreshed along with the global CRL during CRL interval processing.

As certificates expire, they are no longer eligible for revocation and will not appear on any CRL. Therefore, over time, each distribution point will become inactive. PKI Services automatically retires DP CRLs that become inactive by no longer publishing their CRLs. However, retired DP CRLs previously published to LDAP remain in LDAP. PKI Services makes no attempt to delete these.

Even when using distribution point CRLs, the single non-DP CRL (global CRL) is still created. Revoked certificates containing the `CRLDistributionPoints` extension will appear only on the appropriate DP CRL, not the global CRL.

### What about CA certificates?

PKI Services can be used to create other subordinate certificate-authority certificates. Since revocation activity against these CA certificates is normally low, PKI Services by default does not partition authority revocation lists (ARLs). You can choose to create a distribution point ARL in a single partition for the purpose of checking the revocation status of CA certificates. (See “Creating a distribution point ARL.”) When you choose to create a DP ARL, your CA certificates will contain a CRLDistributionPoints extension.

When you do not choose to create a DP ARL (ARLDist=F), applications wishing to check the revocation status of a CA certificate must check the global ARL. In addition, when ARLDist=F, CA certificates do not contain a CRLDistributionPoints extension, although they are treated as if they had the extension when determining the partitioning of the global CRL. For instance, with ARLDist=F and CRLDistSize=10, if you issue 10 CA certificates plus one non-CA certificate, the non-CA certificate information would be published to the second distribution point CRL. (The first DP CRL would remain empty.)

---

## Using the OCSP responder

As an alternative, or in addition to publishing revocation information with CRLs, you can choose to enable an Online Certificate Status Protocol (OCSP) responder. An OCSP responder is enabled when OCSPType is set to basic in the **CertPolicy** section of the PKI Services configuration file as shown in Table 19 on page 51, and when the certificate contains the necessary OCSP responder information in the AuthInfoAccess extension. (See “TEMPLATE sections” on page 103.)

---

## Creating a distribution point ARL

You can choose to create a distribution point (DP) authority revocation list (ARL) to support revocation status checking for certificate authority (CA) certificates. You choose DP ARL processing by customizing the ARLDist parameter in the **CertPolicy** section of the pkiserv.conf. If you do not customize this parameter, PKI Services does not partition the ARL and, therefore, applications must check the global ARL to check the revocation status of a CA certificate.

### ARLDist=F (default)

No distribution point ARL will be created.

### ARLDist=T

When distribution point CRLs are also enabled (when CRLDistSize is greater than zero), you can specify **T (True)** to create a distribution point ARL.

When DP ARL processing is enabled, PKI Services provides the following support:

- Create a single distribution point (DP) for all CA certificates
- Build a CRLDistributionPoints extension containing both the distinguished name and the URI format for the DP.

DP ARL processing for CA certificates is similar to the DP CRL processing for non-CA certificates with the following differences:

- There is only one DP ARL. Its name is formed by the value specified in the CRLDistName parameter in the **CertPolicy** section of the pkiserv.conf, appended with **0** (zero). By appending a zero, the name of the DP ARL never conflicts with the name of a DP CRL. For example, if CRLDistName=CRL, then the DP ARL is named CRL0, and the DP CRLs are named CRL1, CRL2, and so forth.

- The DP ARL is a mirror copy of the global ARL. In other words, each revoked CA certificate will appear in both the DP ARL and the global ARL. By contrast, a revoked non-CA certificate is listed in the DP CRL but not in the global CRL when DP CRL processing is enabled.
- The attribute string appended to the URI format for the LDAP protocol is `?authorityRevocationList`. Otherwise, the `CRLDistributionPoints` extension of a CA certificate appears quite similar to that of a non-CA certificate. See Figure 4 on page 158 for a sample `CRLDistributionPoints` extension for a CA certificate. This sample contains several different name formats. Notice the URI format located at the end of the sample.

## Advanced customization

```
SEQUENCE {
  OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
  OCTET STRING, encapsulates {
    SEQUENCE {
      SEQUENCE {
        [0] {
          [0] {
            [4] {
              SEQUENCE {
                SET {
                  SEQUENCE {
                    OBJECT IDENTIFIER
                      countryName (2 5 4 6)
                    UTF8String (1997) 'US'
                  }
                }
              SET {
                SEQUENCE {
                  OBJECT IDENTIFIER
                    organizationName (2 5 4 10)
                  UTF8String (1997) 'Mycompany'
                }
              }
              SET {
                SEQUENCE {
                  OBJECT IDENTIFIER
                    organizationalUnitName (2 5 4 11)
                  UTF8String (1997) 'Retail'
                }
              }
              SET {
                SEQUENCE {
                  OBJECT IDENTIFIER
                    commonName (2 5 4 3)
                  UTF8String (1997) 'CRL0'
                }
              }
            }
          }
        }
      }
    }
  }
  SEQUENCE {
    [0] {
      [0] {
        [6]
        'http://cr1.MyCompany.de/CRL0.cr1'
      }
    }
  }
  SEQUENCE {
    [0] {
      [0] {
        [6]
      }
    }
  }
  'ldap://ldap.MyCompany.de/CN=CRL0,OU=Retail,O=Mycompany,C=US?authorityRevocationList'
}
```

Figure 4. A sample *CRLDistributionPoints* extension for a certificate authority (CA) certificate

---

## Adding an application domain

This topic describes adding a new application domain. If you want to add a new certificate authority (CA) domain, see “Adding a new CA domain” on page 161.

By default, all CGIs reside under a common URL: `http(s)://<webserver-domain-name>/PKIServ`. Based on this, all users, including PKI administrators, have the same PKI Services home page, Web content and supported certificate templates. In other words, by default, there is a single application domain: PKISERV.

The sample PKI Services template file (`pkiserv.tpl`) contains two application sections: PKISERV and CUSTOMERS. The PKISERV application includes all templates and functions. The CUSTOMERS application contains all templates and functions but it does not contain the button at the bottom of the home page to link to the Administration home page. Therefore, using these two application sections, your users can be easily divided between two subsets—customers and administrators.

You will probably want to separate your administration users and your end users. You may also need to further subset your end user population by adding different application domains for different groups of end users. Both of these objectives can be accomplished by using multiple applications.

The PKI administrators can be directed to use the existing application domain: `http(s)://<webserver-domain-name>/PKIServ`, while each subset of end users can be given a new, unique application domain: `http(s)://<webserver-domain-name>/<appl-domain-name>`.

To create multiple application domains, execute the subtasks in Table 41. Both are tasks for the Web server programmer.

Table 41. Task roadmap for creating multiple application domains

Subtask	Associated procedures (See ...)
Update the PKI Services template file	"Steps for creating multiple application sections in the PKI Services template file"
Update the Web server configuration files	"Steps for adding application domains to the Web server configuration files" on page 160

## Steps for creating multiple application sections in the PKI Services template file

Perform the following steps to create multiple application sections to the PKI Services template file:

1. Edit the `pkiserv.tpl` file and find the CUSTOMERS application section.

2. Replicate the CUSTOMERS section and specify a unique name for the new application section.

`<APPLICATION NAME=appl-section-name>`

**Rule:** The application section name must be one word, all uppercase characters.

3. Determine which certificate types are required by this user subset. Based on these requirements, select the certificate templates that belong in the new application set by adding or removing template names from this new section as needed.

4. Customize the content of the Web pages for this application by modifying the `<CONTENT>...</CONTENT>` subsection. (See “TEMPLATE sections” on page 103 for a description of each subsection.)

---

5. Similarly, customize the original CUSTOMERS application by re-executing Steps 3 and 4, this time editing the content of the CUSTOMERS Web pages.

---

6. Repeat Steps 1–4 for each application section you need to add.

---

7. Optionally, rename the original CUSTOMERS application to a new section name, if desired.  
`<APPLICATION NAME=section-name>`

**Rule:** The application section name must be one word, all uppercase characters.

## Steps for adding application domains to the Web server configuration files

### Before you begin:

- This procedure requires Web server programming skills and requires editing both the `httpd.conf` file and the `httpd1443.conf` file.
- The home page URL for the new or renamed domains would be as follows:  
`http://<webserver-fully-qualified-domain-name>/<new-appl-domain-name>/public-cgi/camain.rexx`  
  
where *new-appl-domain-name* corresponds to the new section name added in the template file in “Steps for creating multiple application sections in the PKI Services template file” on page 159. However, in the Web server files, the new name is case-sensitive but need not be in upper case only.
- Make note of the case you select for each character of the new *new-appl-domain-name* name. This case-sensitive value will become part of the URL for your home page and you must use it consistently in each set of Protect, Redirect, and Exec statements you add to your Web server configuration file.
- The administration home page URL does not change. (There is one common administration application that handles all application domains.)
- If your PKI installation has changed the name of the Customers domain, you must change all occurrences of Customers to its new value in both files. (The new value is not case-sensitive.)
- If your installation has added a new application domain, use the following procedure.

Perform the following steps to add domains or rename the Customers domain in Web server configuration files—`httpd.conf` and `httpd1443.conf`—for each new application section added to `pkiserv.tmpl`:

1. Replicate each of the following lines in both the `httpd.conf` and `httpd1443.conf` files:  
    `Protect /Customers/...`  
    `Redirect /Customers/...`  
    `Exec /Customers/...`

2. Change each occurrence of the name Customers in the replicated lines in both files to the new application domain name.

**Example:**

```
Protect /Employees/...
Redirect /Employees/...
Exec /Employees/...
```

In contrast to the application section name, the domain name value is case-sensitive and it need not be uppercase. However, you must use it consistently in each set of Protect, Redirect, and Exec statements. This value will become part of the URL for your home page.

**When you are done:** You have defined a new PKI Services application domain at:

**Example:**

```
http://<webserver-fully-qualified-domain-name>/employees/public-cgi/camain.rexx
```

---

## Adding a new CA domain

When you want to operate more than one certificate authority (CA) on a single z/OS image, you must create a separate CA domain for each CA. Each CA domain uses its own daemon and operates as its own instance of PKI Services. This topic describes how to add a new CA domain. (If you want to add a new *application* domain, see “Adding an application domain” on page 158.)

When you add CA domains, you can create a PKI infrastructure that contains subsets of end user populations (application domains), each supported by its own unique PKI Services application (PKI Services daemon and URL) and optionally by its own dedicated set of PKI administrators. If you already use multiple application domains, the key advantage of adding multiple CA domains is that you can build a certificate hierarchy of CAs and optionally provide certificate services to multiple organizations.

By adding a new CA domain, your users still have a unique URL and set of certificate templates to choose from, but they also have the services of their own CA including the CA's certificate, signing key, VSAM data sets, and LDAP repository. Enabling multiple CAs is a natural extension for multiple application domains. Each CA domain can represent one instance of a CA, backed by a unique instance of the PKI Services daemon (and all its associated componentry), yet requiring no more than a single HTTP Server and a single set of CGIs.

Figure 5 on page 162 contains an illustration showing two CA domains—one for employees and one for customers. In the illustration, a single shared administrator supports both CA domains. (You can decide to share a common administrator across multiple CA domains or have separate administrators who are each dedicated to only one CA domain.)

## Advanced customization

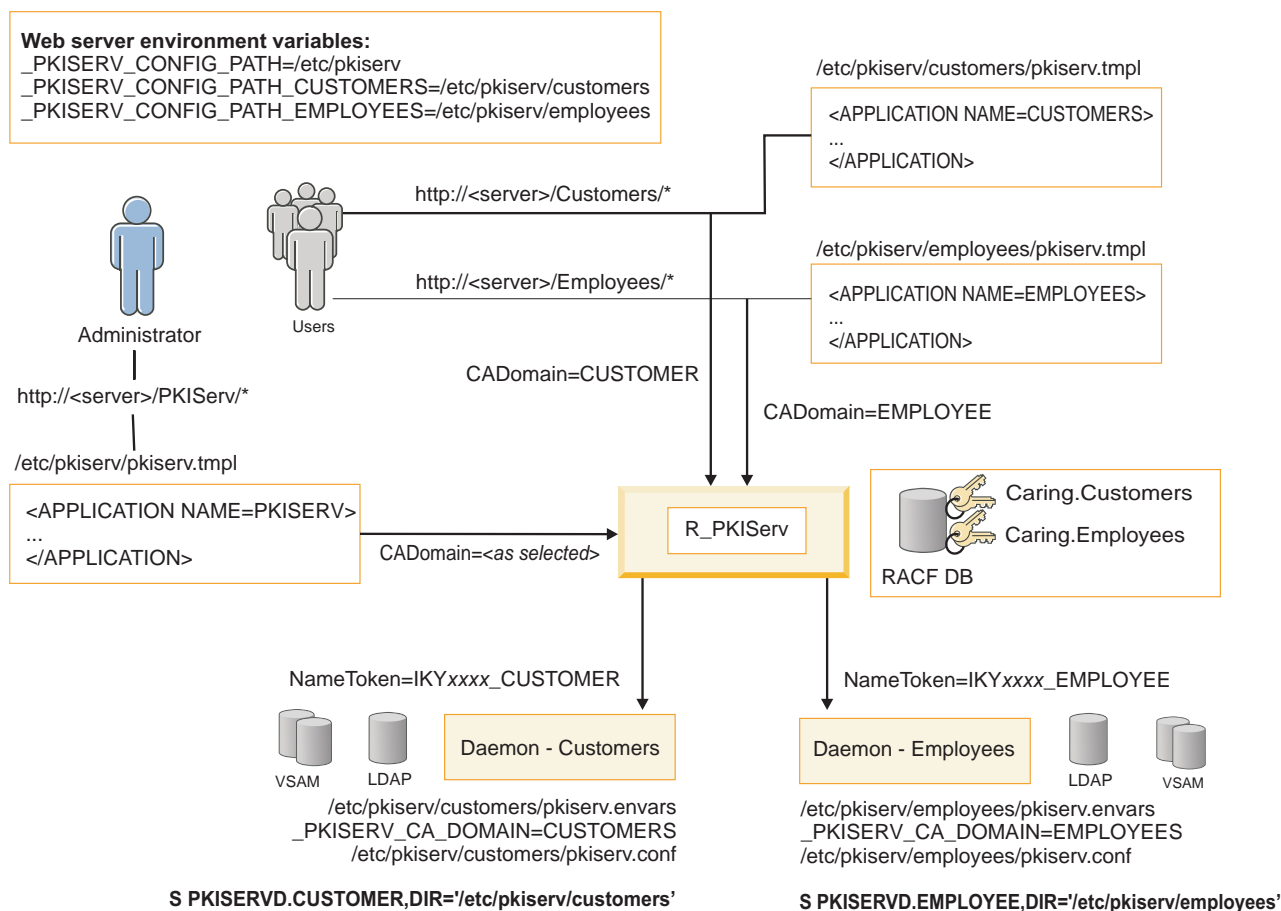


Figure 5. Illustration of two CA domains—one for employees and one for customers—administered by a single shared administrator who administers both domains

## Task overview

This topic includes a task roadmap that you can use to add a new CA domain. The roadmap includes several subtasks that are listed in as listed in Table 42 on page 164. It is intended to direct you to add a new CA domain after you have completed all required tasks in Part 2, “Configuring your system for PKI Services,” on page 25. Before you begin this task, you have already implemented and tested the default setup for PKI Services and ensured that it operates properly as a single CA domain.

For each CA you add, you create a dedicated copy of the VSAM files, CA certificate, key ring, and LDAP namespace. You also create a dedicated copy of the PKI Services configuration file (pkiserv.conf), templates file (pkiserv.tpl), and environment variables file (pkiserv.envars), each in its own directory. You update the following CA-specific information in these files:

- pkiserv.conf—contains the CA-specific key ring name, VSAM data set names, and optionally CRLDistDirPath.
- pkiserv.envars—contains a variable \_PKISERV\_CA\_DOMAIN to specify CA domain and the variable \_PKISERV\_CONFIG\_PATH sets the directory for each CA domain.
- pkiserv.tpl—contains the name of the end-user application section (default is CUSTOMER) that you rename to a CA-specific name, such as <APPLICATION



NAME=EMPLOYEE>. It also contains the name of the administrative application section (default is PKISERV) that you can rename to a CA-specific name, such as <APPLICATION NAME=ADMEMPLOYEES>.

For the Web server, you customize the HTTP configuration files and the HTTP environment variables file to add additional application domains. For example, based on Figure 5 on page 162, you would add the following URL (to the HTTP configuration files) and the following configuration path (to the HTTP environment variables file).

New application domain (URL)	New configuration path
http://<server-name>/Employees/*	_PKISERV_CONFIG_PATH_EMPLOYEES=/etc/pkiserv/employees

## Task roadmap for adding CA domains

### Before you begin:

- Complete all required tasks in Part 2, “Configuring your system for PKI Services,” on page 25. This task roadmap is intended to direct you to add a new CA domain *after* you have already implemented and tested the default setup for PKI Services and ensured that it operates properly as a single CA domain.
- Review Table 42 on page 164 to see the subtasks involved and the skills required for each subtask. (The team members listed are based on role definitions established in “Identifying skill requirements” on page 11.)

To create a new CA domain, complete the subtasks in Table 42. Subtask 1 guides you through planning. Subtask 2 is a one-time setup you do when you add your first additional CA domain. Subtasks 3–8 are each done once for every CA domain you add. (See Part 2, “Configuring your system for PKI Services,” on page 25 for additional details about these subtasks.)

**Guideline:** After you complete Subtasks 1 and 2 (planning and reconfiguring), perform Subtasks 3–8 for your *first* new CA domain and ensure that it operates properly before adding your second CA domain.

## Advanced customization

Table 42. Task roadmap for adding a new CA domain

Subtask	Team member	Associated procedure (See...)
1. Plan additional CA domains.	UNIX programmer	"Subtask 1: Steps for planning additional CA domains."
2. Reconfigure your initial CA domain to allow it to coexist with other CA domains. (This is a one-time setup.)	UNIX programmer	"Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains" on page 166.
3. Run IKYSETUP.	MVS programmer	"Subtask 3: Steps for running the IKYSETUP exec" on page 168.
4. Configure the UNIX environment.	UNIX programmer	"Subtask 4: Steps for configuring the UNIX environment" on page 170.
5. Update the PKI Services template file.	Web server programmer	"Subtask 5: Steps for updating the PKI Services template file" on page 171.
6. Update the Web server configuration.	Web server programmer	"Subtask 6: Steps for updating the Web server configuration" on page 173.
7. Allocate VSAM data sets.	MVS programmer	"Subtask 7: Steps for allocating VSAM data sets" on page 173.
8. Start PKI Services.	MVS programmer	"Subtask 8: Steps for starting PKI Services" on page 174.

## Recording your progress adding CA domains

As you complete each subtasks for adding a new CA domain, use Table 43 mark your progress. The tasks correspond to the subtasks listed in the roadmap in Table 42.

Table 43. Multiple CA domains: Worksheet #1 for recording progress adding new CA domains

Subtask	First CA domain	Second CA domain	Third CA domain
1. Plan additional CA domains.	<input type="checkbox"/>		
2. Reconfigure initial domain. (once only)	<input type="checkbox"/>		
3. Run IKYSETUP.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Configure the UNIX environment.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Update the PKI Services template file.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Update the Web server configuration.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Allocate VSAM data sets.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Start PKI Services.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Guideline:</b> Perform Subtasks 3–8 for your <i>first</i> new CA domain and ensure that it operates properly before adding your second CA domain.			

## Subtask 1: Steps for planning additional CA domains

Perform the following steps to plan additional CA domains.

1. Determine how many instances of PKI Services (CA domains) you will operate in addition to the initial domain you configured when you originally customized PKI Services.

For each CA domain, you need to pick a nickname to use as the CA domain name. The CA domain name is used to qualify the resources used by that CA domain. For example, the CA domain named Employees uses the following resources:

**Examples:**

- Web page URLs (in mixed case)—  
`http://webserver-domain-name/Employees/public-cgi/camain.rexx`
- Data set qualifiers (in upper case)—VSAM ICL data set  
`PKISRV.D.EMPLOYEE.VSAM.ICL`
- Pathnames (in lower case)—  
`/etc/pkiserv/employees/pkiserv.conf`

- 
2. Decide how you will administer multiple CA domains. Will you share a *common* set of administrators across all your CA domains or will you have a *dedicated* set of administrators for each CA domain?

If you use a *dedicated* set for each CA domain, you need to pick a second nickname for each CA domain—for its administrative domain.

- 
3. Determine your CA domain names. Unless you renamed the default domain names when you originally customized PKI Services, the initial name for the application domain is Customers and its administrative domain name is PKIServ. Your new CA domain names (nicknames) must differ from these values.

**Rules for domain names:**

- Domain names are 1–8 characters but may exceed 8 characters if the first 8 characters are unique from your other domain names.
- The characters in the domain name are limited to the following character set: alphanumeric characters (a–z, A–Z, 0–9) and the hyphen (-).
- The first character must not be a number or hyphen.

- 
4. Record information about your CA domains in Table 44 on page 166 and Table 45 on page 166.

Row **1** in each table is already filled in with the defaults for an initial CA domain (Customers). Row **2** in each table is an example of a new CA domain managed by the same (shared) group of administrators. Row **3** in each table is an example of the same CA domain from Row **2** managed by a *dedicated* group of administrators.

The rows in each table that are already filled in use the default values for the following variables when PKI Services was installed. (Your MVS programmer might have chosen different directories.)

Installation variable	Default directory name
<i>install-dir</i>	/usr/lpp/pkiserv
<i>runtime-dir</i>	/etc/pkiserv

- a. Fill in the values for new CA domains, administrative domains, and directories in Table 44 on page 166. You can add your information in the blank lines below or you can modify or cross out the sample rows.

## Advanced customization

Table 44. Multiple CA domains: Worksheet #2 for planning your domain names

CA domain name (runtime directory)	Truncated CA domain name	Administrative domain name (runtime directory)
1. Customers (/etc/pkiserv)	CUSTOMER	PKIServ (/etc/pkiserv)
2. Employees (/etc/pkiserv/employees)	EMPLOYEE	PKIServ (/etc/pkiserv)
3. Employees (/etc/pkiserv/employees)	EMPLOYEE	AdmEmployees (/etc/pkiserv/employees)
4.		
5.		

- b. Fill in your RACF user IDs, groups, and VSAM data set qualifiers in Table 45. You can add your information in the blank lines below or you can modify or cross out the sample rows.

Table 45. Multiple CA domains: Worksheet #3 for planning your RACF identifiers, z/OS UNIX identifiers, and VSAM data set names

Daemon user ID (UID)	Surrogate user ID (UID)	PKI administration group name (GID)	VSAM data set qualifiers
1. PKISRVD (554)	PKISERV (555)	PKIGRP (655)	PKISRVD.VSAM
2. PKISRVD (556)	PKISERV (557)	PKIGRP (657)	PKISRVD.EMPLOYEE.VSAM
3. PKIDEMP (556)	PKISEMP (557)	PKIGEMP (657)	PKISRVD.EMPLOYEE.VSAM
4.			
5.			

### Subtask 2: Steps for reconfiguring your initial CA domain to allow it to coexist with other CA domains

Perform the following steps to reconfigure your initial CA domain to allow it to coexist with other CA domains. (This is a one-time setup that will suffice no matter how many CA domains you add.)

1. If PKI Services is running, stop it by issuing the following MVS console command:  

```
P PKISERV
```
2. Update the PKI Services environment variables in the `pkiserv.envs` file as follows.
  - a. (Optional) If your initial CA domain does not use its own `pkiserv.envs` file, copy the default `pkiserv.envs` file from the PKI Services install directory by issuing the following command from the UNIX command line:  

```
cp -p /usr/lpp/pkiserv/samples/pkiserv.envs /etc/pkiserv
```
  - b. Edit the new copy of `pkiserv.envs` file by entering the following command:  

```
oedit /etc/pkiserv/pkiserv.envs
```
  - c. Add a PKI Services environment variable identifying your initial CA domain name (see Table 44) in uppercase characters.  
**Example:**  

```
_PKISERV_CA_DOMAIN=CUSTOMERS
```

### 3. Update the HTTP server's environment variables and configuration directives as follows.

#### a. Update the HTTP server's environment variables.

- 1) Edit the httpd.envvars file by entering the following command:

```
oedit /etc/httpd.envvars
```

- 2) Add an environment variable identifying the runtime directory of your initial CA domain. (Check Table 44 on page 166.)

##### Example:

```
_PKISERV_CONFIG_PATH_CUSTOMERS=/etc/pkiserv
```

- 3) (Optional) If you intend to have a dedicated set of administrators for each CA domain, add an environment variable that specifies the runtime directory for the administrative domain. (Check Table 44 on page 166.)

##### Example:

```
_PKISERV_CONFIG_PATH_PKISERV=/etc/pkiserv
```

#### b. Update the HTTP configuration directives.

- 1) Edit the httpd.conf file by entering the following command:

```
oedit /etc/httpd.conf
```

- 2) (Optional) If your HTTP configuration file includes an InheritEnv directive, add the following directive for each new CA domain you add. Replace the **CUSTOMERS** value with the `ca_domain` value you specified in Table 17 on page 36.

This directive specifies that you want your CGI programs to inherit the PATH environment variable so that the PKI Services Web pages of each CA domain can retrieve any certificate through the Web page of any CA domain.

##### Example:

```
InheritEnv _PKISERV_CONFIG_PATH_CUSTOMERS=/etc/pkiserv
```

### 4. Update the RACF access controls for the R\_PKIServ SAF callable service as follows. (Any change to environment variables in Step 3 requires a corresponding change to RACF access control.)

- a. Determine the PKI Services surrogate user ID (default is PKISERV) and the PKI Services administrators group (default is PKIGRP). To do this, refer to the log file created when the IKYSETUP REXX exec was originally run for your initial CA domain.

- b. Execute the following RACF commands from the TSO command line. Replace the highlighted values with your own, if different:

##### Examples:

```
RDELETE FACILITY IRR.RPKISERV.**
RDEFINE FACILITY IRR.RPKISERV.*.CUSTOMER
PERMIT IRR.RPKISERV.*.CUSTOMER CLASS(FACILITY) ID(PKISERV) ACCESS(CONTROL)
RDELETE FACILITY IRR.RPKISERV.PKIADMIN
RDEFINE FACILITY IRR.RPKISERV.PKIADMIN.CUSTOMER
PERMIT IRR.RPKISERV.PKIADMIN.CUSTOMER CLASS(FACILITY) ID(PKIGRP)
ACCESS(CONTROL)
PERMIT IRR.RPKISERV.PKIADMIN.CUSTOMER CLASS(FACILITY) ID(PKISERV)
ACCESS(NONE)
SETOPTS RACLIST(FACILITY) REFRESH
```

**Restriction:** If the name of your initial CA domain is longer than 8 characters, you must truncate it to exactly 8 characters when you define the resource name in the FACILITY class profiles. (In this example, the name CUSTOMERS was truncated to CUSTOMER in the second RDEFINE FACILITY command.)

5. (Optional) You have reconfigured your initial CA domain to allow it to coexist with other CA domains. If you want, you can test the reconfiguration now. To test it, do the following:
  - a. Restart PKI Services using the following MVS console command. Replace the highlighted values with your own, if different.

**Guideline:** To simplify your environment, start this instance of PKI Services using a JOBNAME that matches the truncated name of the CA domain. (See your truncated value in Table 44 on page 166.) If you use the truncated values as job names, it will be easier to distinguish multiple jobs that run PKI Services after you add other CA domains.

**Example:**

```
S PKISERVD,JOBNAME=CUSTOMER,DIR='/etc/pkiserv/'
```
  - b. Restart the HTTP servers to enable your environment variable changes.

```
F IMWEBSRV,APPL=-restart
```
  - c. Test that your PKI Services application is functioning properly. Go to your Web pages by entering the following URL from your browser:

```
http://webserver-fully-qualified-domain-name/PKIServ/public-cgi/camain.rexx
```

The *webserver-fully-qualified-domain-name* is the common name (CN) portion of the Web server's distinguished name; see Table 11 on page 29. You should be able to go through your Web pages to request, retrieve, and revoke a certificate of type "PKI browser certificate for authenticating to z/OS". Ensure you can do this before adding an additional CA domain.

**When you are done:** You have successfully reconfigured your initial CA domain to allow it to coexist with other CA domains. You can now perform each of the remaining subtasks once for each new CA domain.

Continue to the next subtask. **Guideline:** Complete Subtasks 3–8 for your *first* new CA domain and ensure that it operates properly before adding another CA domain.

### Subtask 3: Steps for running the IKYSETUP exec

**Before you begin:** This procedure requires you to be familiar with the information in Chapter 4, "Running IKYSETUP to perform RACF administration," on page 27. You will find additional details about the following steps there.

Perform the following steps to customize a unique execution of the IKYSETUP REXX exec for this new CA domain.

1. Locate the IKYSETUP exec you originally customized for your initial CA domain and copy it to a data set member that you can edit.
2. Edit the new copy of IKYSETUP and set the `ca_domain` variable to the name of this new CA domain. Type the domain name preserving the case of each character as you wish it to appear in Web page URLs.

3. If you intend to have a dedicated set of administrators for each CA domain, customize the following variables with your values for this CA domain.

Variable name	Use your value from ...
daemon_uid	Table 11 on page 29
pki_gid	
pkigroup_mem	
surrog_uid	
daemon	Table 17 on page 36
surrog	
pkigroup	Table 44 on page 166 (Use the truncated name of the administrative domain.)

4. (Optional) If you are creating multiple CAs as part of a certificate hierarchy where a previous CA domain is to be superior (as issuer or signer) of this CA domain, set `signing_ca_label` to match the label of the certificate in RACF that will issue the certificate for this CA domain.  
Otherwise, skip to Step 5 and leave `signing_ca_label=""` (the default).

5. Update any other values, such as `ca_dn` and `ra_dn`, that you choose to differ from your initial settings or the defaults.  
You need *not* change any values in this step unless you choose to set these values to something particular for your installation. (This is because when you specify a `ca_domain` value, the IKYSETUP exec automatically qualifies any value that PKI Services requires to be unique by adding the CA domain name.)

6. Execute IKYSETUP by entering the following TSO command:

```
EX 'data-set-name(new-member-name)' 'RUN(NO)'
```

7. Review the log data set to ensure that the commands created by IKYSETUP match your expectations. (For more information about these commands, see “Actions IKYSETUP performs by issuing RACF commands” on page 351.) Edit again as needed and rerun.

8. When you are satisfied with the commands and information in the log data set, rerun the IKYSETUP exec by entering the following TSO command:

```
EX 'data-set-name(new-member-name)' 'RUN(YES)'
```

9. Check your IKYSETUP log and record the name of the SAF key ring (your `ca_ring` value from Table 17 on page 36).

*Table 46. IKYSETUP log information you need to record*

Name of the SAF key ring:	



**When you are done:** You have customized and run the IKYSETUP exec for this CA domain. Record your progress in Table 43 on page 164.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

### Subtask 4: Steps for configuring the UNIX environment

**Before you begin:** This procedure requires you to be familiar with the information in Chapter 5, “Configuring the UNIX runtime environment,” on page 45. You will find additional details about the following steps there.

Perform the following steps to configure the UNIX environment for this new CA domain.

1. Set up a var directory for this CA domain. Perform the steps in “Steps for setting up the var directory” on page 63
2. Locate the pkiserv.conf, pkiserv.envars, and pkiserv.tmp1 files you originally used to create your initial CA domain. Copy them into the appropriate runtime directory for your new CA domain. (Check Table 44 on page 166.) For a new CA domain called Employees, run the following commands from the UNIX command line. (You may have to make the directory first.)

**Examples:**

```
mkdir /etc/pkiserv/employees
chown pkisrvc /etc/pkiserv/employees
cp -p /etc/pkiserv/* /etc/pkiserv/employees
```

3. Edit the new pkiserv.conf file by entering the following command:

**Example:**

```
oedit /etc/pkiserv/employees/pkiserv.conf
```

4. Change the following sections of pkiserv.conf as described for this CA domain. (Find detailed information for each variable in Table 19 on page 51.)

<b>ObjectStore</b>	Qualify each VSAM data set name with the CA domain name. <b>Example:</b> ObjectDSN='pkisrvc.employee.vsam.ost' (See “Subtask 7: Steps for allocating VSAM data sets” on page 173.)
<b>CertPolicy</b>	If CRLDistDirPath is not null, modify it to reference the correct subdirectory. (You may have to create this directory.) <b>Example:</b> CRLDistDirPath=/var/pkiserv/employees.
<b>General</b>	Update each pathname to the correct subdirectory. <b>Example:</b> ReadyMessageForm=/etc/pkiserv/employees/readymsg.form
<b>SAF</b>	Update the key ring name to match the ca_ring value you recorded in Table 46 on page 169. <b>Example:</b> PKISRVC/Caring.Employees
<b>LDAP</b>	Do not update the <b>LDAP</b> section unless you need to change the LDAP directory. If you need to change it, see “Steps for tailoring the LDAP section of the configuration file” on page 72.



Make sure the LDAP directory is configured with a suffix for this CA domain. (See the explanation for the `Suffix` variable in Table 20 on page 65.)

5. (Optional) Change other values in any section of `pkiserv.conf` as desired for this CA domain.

6. Edit the new `pkiserv.envars` file by entering the following command:

**Example:**

```
oedit /etc/pkiserv/employees/pkiserv.envars
```

7. Define the `_PKISERV_CA_DOMAIN` environment variable for this CA domain name. (For details, see “The `pkiserv.envars` environment variables file” on page 349.)

**Example:**

```
_PKISERV_CA_DOMAIN=EMPLOYEE
```

**When you are done:** You have updated the `pkiserv.conf` and `pkiserv.envars` files for this CA domain. Record your progress in Table 43 on page 164.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

## Subtask 5: Steps for updating the PKI Services template file

**Before you begin:** This procedure requires you to be familiar with the information in Chapter 11, “Customizing the end-user Web application,” on page 91. You will find additional details about the following steps there.

Perform the following steps to customize the PKI Services template file (`pkiserv.tpl`) for this new CA domain.

1. Edit the new `pkiserv.tpl` file (you copied it in Step 2 of “Subtask 4: Steps for configuring the UNIX environment” on page 170) by entering the follow command from the UNIX command line:

**Example:**

```
oedit /etc/pkiserv/employees/pkiserv.tpl
```

2. Locate all occurrences of the CA domain named `Customers` (in mixed case, upper case, or lower case) and change them to the name of this new CA domain, being careful to preserve the case.

**Example:** If the new CA domain is `Employees` (in mixed case), change the default values to the name of your new CA domain.

Default values for the Customers CA domain	New values for the Employees CA domain
<code>ACTION="/Customers/..."</code> (in mixed case)	<code>ACTION="/Employees/..."</code> (also in mixed case)
<code>&lt;APPLICATION NAME=CUSTOMERS&gt;</code> (in upper case)	<code>&lt;APPLICATION NAME=EMPLOYEES&gt;</code> (also in upper case)

3. If you intend to have the *same* set of administrators for all your CA domains, *skip* this step and proceed to Step 4.

If you intend to have a *dedicated* set of administrators for each CA domain, change the name of the PKISERV application section to the corresponding name of the administrative domain in Table 44 on page 166. This value must be specified in uppercase characters only.

**Example:** If the new administrative domain is named AdmEmployees, change the default value (PKISERV) to the name of your new administrative CA domain.

Default value for the PKISERV administrative CA domain	New value for the new AdmEmployees administrative CA domain
<APPLICATION NAME=PKISERV> (in upper case)	<APPLICATION NAME=ADMEMPLOYEES> (also in upper case)

4. If you intend to have the same set of administrators for all your CA domains, edit the main templates file (/etc/pkiserv/pkiserv.tpl) as follows:

Do not update the PKISERV application section in this domain-specific pkiserv.tpl file; it will *not* be used. The PKISERV application section in the templates file for your initial CA domain will be used instead.

- a. Replicate the following lines in the APPLICATION section of the PKISERV application:

```
<h3>Go the Customers' home page </h3>
<FORM name=admform METHOD=GET
  ACTION="/Customers/public-cgi/camain.rexx">
<p>
<INPUT TYPE="submit" VALUE="Customers' Home Page">
</FORM>
```

- b. Change all occurrences of the string Customers in the replicated lines to the name of this CA domain, being careful to preserve case.

**Example:** Change ACTION="/Customers/public-cgi/camain.rexx"> to ACTION="/Employees/public-cgi/camain.rexx">.

- c. Uncomment the %%SelectCADomain%% directive in the ADMINSCOPE subsection of the APPLICATION section for the PKISERV application by removing the leading # character. (The %%SelectCADomain%% directive enables multiple CA administration.)

- d. Update the SelectCaDomain insert to include an OPTION entry for this CA domain. If this CA domain will be used more often than any other, mark the entry SELECTED and removed SELECTED from any other entry.

**Example:**

```
<INSERT NAME=SelectCaDomain>
<p> Select the CA domain to work with
<SELECT NAME="domain">
# rename and replicate the following line for every CA domain and
# determine which one should be SELECTED by default, if any
<OPTION VALUE="Employees" SELECTED>Employees
<OPTION VALUE="Customers">Customers
</SELECT>
```

**When you are done:** You have customized the PKI Services template file (pkiserv.tpl) for this CA domain. Record your progress in Table 43 on page 164.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

## Subtask 6: Steps for updating the Web server configuration

**Before you begin:** This procedure requires you to be familiar with the information in the following topics where you find additional details:

- “Steps for adding application domains to the Web server configuration files” on page 160
- “Steps for updating the z/OS HTTP Server configuration files” on page 67.

Perform the following steps to customize the z/OS HTTP Server configuration files for this new CA domain.

1. Add this new CA domain (check Table 44 on page 166 for domain name and directory) following the instructions in “Steps for adding application domains to the Web server configuration files” on page 160.

2. (Optional) If you intend to have a dedicated set of administrators for each CA domain, repeat Step 1 for the administrative domain. (Check Table 44 on page 166 for domain name and directory.) Otherwise, skip to Step 3.

3. Update the HTTP Server environment variables. Edit the `httpd.envvars` file by entering the following command from the UNIX command line:

```
oedit /etc/httpd.envvars
```

4. Add the environment variable identifying the runtime directory of this CA domain. (Check Table 44 on page 166.)

**Example:**

```
_PKISERV_CONFIG_PATH_EMPLOYEES=/etc/pkiserv/employees
```

5. (Optional) If you intend to have a dedicated set of administrators for each CA domain, add the environment variable identifying the `pkiserv.tmp1` directory of this administrative CA domain.

**Example:**

```
_PKISERV_CONFIG_PATH_ADMEMPLOYEES=/etc/pkiserv/employees
```

**When you are done:** You have customized the z/OS HTTP Server configuration files for this CA domain. Record your progress in Table 43 on page 164.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

## Subtask 7: Steps for allocating VSAM data sets

**Before you begin:** This procedure requires you to be familiar with the information in Chapter 9, “Creating VSAM data sets,” on page 77. You will find additional details about the following steps there.

Perform the following steps to allocate the needed VSAM files for this new CA domain.

1. Locate the IKYCVSAM JCL job you originally customized for your initial CA domain and copy it to a data set that you can edit.

---
2. Change all occurrences PKISRV.D.VSAM to include corresponding VSAM data set qualifier from Table 45 on page 166. For example, if this new CA domain is Employees, then the VSAM data sets might be qualified as follows:  
**Example:**  
`PKISRV.D.EMPLOYEE.VSAM.data-set-suffix`

---
3. Submit the JCL job when your changes are complete.

---

**When you are done:** You have allocated the needed VSAM files for this CA domain. Record your progress in Table 43 on page 164.

Continue to the next subtask. **Guideline:** Complete all subtasks for this new CA domain and ensure that it operates properly before adding another CA domain.

### Subtask 8: Steps for starting PKI Services

**Before you begin:** This procedure requires you to be familiar with the information in Chapter 10, “Starting and stopping PKI Services,” on page 85. You will find additional details about the following steps there.

Perform the following steps to start a separate instance of PKI Services for this new CA domain.

1. Start the PKI Services daemon for this CA domain by entering the MVS console START command qualified with the appropriate runtime directory. (Check Table 44 on page 166.)  
**Example:**  
`S PKISRV.D,JOBNAME=EMPLOYEE,DIR='/etc/pkiserv/employees'`  
**Guideline:** To simplify your environment, give this instance of PKI Services a JOBNAME that matches or relates to this CA domain name. When you add additional CA domains, it will be easier to distinguish multiple jobs running PKI Services.

---
2. Restart the HTTP servers to enable the environment variables you changed for this CA domain. Optionally, you can wait to do this until after you have started all the new domain-specific daemons.  
`F IMWEBSRV,APPL=-restart`

---
3. Test that your new domain-specific PKI Services daemon is functioning properly. Go to your Web pages by entering the following URL from your browser:  
`http://<webserver-fully-qualified-domain-name>/<new-admin-domain-name>/public-cgi/camain.rexx`  
The *webserver-fully-qualified-domain-name* is the common name (CN) portion of the Web server’s distinguished name; see Table 11 on page 29.

You should be able to go through your Web pages to request, retrieve, and revoke an applicable certificate for this CA domain, possibly “PKI browser certificate for authenticating to z/OS”. Ensure you can do this before adding new CA domains.

**When you are done:** You have customized the z/OS HTTP Server configuration files for this CA domain. Record your progress in Table 43 on page 164.

Once your new CA domain works properly, proceed to add another CA domain, if needed. **Guideline:** Perform Subtasks 3–8 for each new CA domain and ensure that the new CA domain operates properly before proceeding to add another.

## Enabling Simple Certificate Enrollment Protocol (SCEP)

The Simple Certificate Enrollment Protocol (SCEP) allows you to securely issue certificates to large numbers of network devices using a primarily automatic enrollment technique. The network devices, usually IPSEC devices such as Cisco routers, must be SCEP-enabled and preregistered (to your CA domain) before they can successfully request certificates from you. To request a certificate, the preregistered SCEP client sends a message (the certificate request) to your CA using the HTTP protocol. (The message is a PKCS #10 request enveloped in a signed PKCS #7 structure.)

You can configure PKI Services to respond automatically to some (or all) SCEP certificate requests, or to submit some (or all) SCEP certificate requests to the PKI administrator for approval or rejection. When you enable automatic enrollment, certificate requests can be automatically approved and synchronously fulfilled, based on the requestor’s knowledge of a predetermined secret, the challenge passphrase.

## Overview of SCEP preregistration

To request certificates using SCEP, a SCEP requestor must be *preregistered* to PKI Services, your CA. You can preregister SCEP clients in batches using the `pkiprereg` utility (see “Using the `pkiprereg` utility” on page 309) or the PKI administrators can preregister individual SCEP clients (one client at a time) using the end-user Web pages.

When PKI administrators preregister a SCEP client, they do so by using the end-user Web page for requesting a certificate and selecting the SCEP (preregistration) certificate template called 5-Year SCEP Certificate – Preregistration. (See “Steps for preregistering a SCEP client” on page 214.) The PKI administrator fills out the request form by specifying the device or client name of the SCEP client, a passphrase for client authentication, and additional (optional) subject name and alternate name information. You can customize the **<CONSTANT>** section of the SCEP (preregistration) certificate template to supply the additional optional information.

When a PKI administrator submits the form for a SCEP (preregistration) certificate request, PKI Services creates a preregistration record—not an actual certificate request—in the VSAM ObjectStore data set (request database). The client name is translated to lowercase characters, truncated to 32 characters if longer, and saved as the Requestor to support searching of the ObjectStore. (Each preregistration record must have a client name that is unique in the first 32 characters, regardless of upper or lower case.)

## Advanced customization

The preregistration record contains the template nickname, passphrase, and additional (optional) subject name and alternate name values. Any other information (unrelated to the subject name or alternate name) specified on the request form is ignored.

When you customize the **<CONSTANT>** section of the SCEP template to supply additional (optional) values for the following variables, those values are not saved in the preregistration record. However, those values are processed when the preregistered client subsequently requests a certificate.

- AuthInfoAcc
- CertPolicies
- Critical
- ExtKeyUsage (not typically used in a SCEP request)
- KeyUsage (not typically used in a SCEP request)
- NotAfter
- NotBefore

## Overview of certificate request processing for preregistered SCEP clients

Following preregistration, when the preregistered SCEP client requests a certificate (sends a SCEP request), PKI Services searches for a preregistration record matching the client name. If found, PKI Services compares the values in the request to the challenge password and any subject name or alternate name information specified by the PKI administrator or supplied in the **<CONSTANT>** template section. (If not found, the SCEP request is automatically rejected.)

Based on the comparison of values in the request with those in the preregistration record, PKI Services considers the request to be in one of the following states:

### Authenticated

When the challenge password matches and all other preregistered values are included in the request

### Semiauthenticated

When the challenge password matches but some other preregistered values are missing from the request

### Unauthenticated

When the challenge password does not match or is missing.

Depending on how you customize the variables in the SCEP (preregistration) certificate template, a certificate request from an Authenticated SCEP client is either automatically approved and fulfilled synchronously or it is queued for administrator approval. Likewise, a certificate request from an Unauthenticated or Semiauthenticated SCEP client is either queued for administrator approval or it is automatically rejected.

## Variables used in the **<PREREGISTER>** section

These are the valid variables you can customize in the **<PREREGISTER>** section of the 5-Year SCEP Certificate – Preregistration template. Some variables *must* be present in your **<PREREGISTER>** section and they are labeled as *required* in the following list.

### AuthenticatedClient (*required*)

Specifies which action PKI Services takes when an authenticated SCEP client submits a certificate request for the first time. Valid values are:

**AutoApprove** *(default)*

Automatically approves certificate requests from authenticated first-time SCEP clients and automatically creates their certificates.

**AdminApprove**

Submits certificate requests from authenticated first-time SCEP clients to your PKI administrator for verification and approval.

**SemiauthenticatedClient** *(required)*

Specifies which action PKI Services takes when a semiauthenticated SCEP client submits a certificate request for the first time. Valid values are:

**AdminApprove** *(default)*

Submits certificate requests from semiauthenticated first-time SCEP clients to your PKI administrator for verification and approval.

**Reject** Automatically rejects certificate requests from semiauthenticated first-time SCEP clients.

**UnauthenticatedClient** *(required)*

Specifies which action PKI Services takes when an unauthenticated SCEP client submits a certificate request for the first time. Valid values are:

**AdminApprove**

Submits certificate requests from unauthenticated first-time SCEP clients to your PKI administrator for verification and approval.

**Reject** *(default)*

Automatically rejects certificate requests from unauthenticated first-time SCEP clients.

**SubsequentRequest** *(optional)*

Specifies which action PKI Services takes when a previously approved SCEP client submits an additional certificate request. If not set, PKI Services uses the **AuthenticatedClient** value. Valid values are:

**AutoApprove** *(default)*

Automatically approves certificate requests from previously approved SCEP clients and automatically creates their certificates.

**AdminApprove**

Submits certificate requests from previously approved SCEP clients to your PKI administrator for verification and approval.

**Reject** Automatically rejects SCEP requests from previously approved clients.

**RenewalRequest** *(optional)*

Specifies which action PKI Services takes when a previously approved SCEP client submits a certificate renewal request. If not set, PKI Services uses the **AuthenticatedClient** value. Valid values are:

**AutoApprove** *(default)*

Automatically approves certificate renewal requests from previously approved SCEP clients and automatically creates their certificates.

**AdminApprove**

Submits certificate renewal requests from previously approved SCEP clients to your PKI administrator for verification and approval.

**Reject** Automatically rejects certificate renewal requests from previously approved SCEP clients.



## Checking certificate fingerprints

There are two instances when the PKI administrator checks certificate *fingerprints* (the SHA1 and MD5 hashes) in support of certificate request processing for SCEP clients.

- Preregistered SCEP clients who request certificates from this CA domain must download the correct PKI Services CA certificate to their workstations before they issue their certificate requests. After the download, the client can use the SCEP client software to display the fingerprints of the downloaded CA certificate and then confirm with the PKI administrator of the CA domain that it is the correct CA certificate.

To match CA certificate fingerprints with a SCEP client, the PKI administrator can display the fingerprints of the CA certificate for this domain by issuing the following MODIFY (or **F**) console command:

```
F PKISERVD,DISPLAY
```

The result of this command is information message IKYP025I. **Sample output:**

```
10.37.39 STC00146: IKYP025I PKI SERVICES SETTINGS:
CA DOMAIN NAME: Customers
SUBCOMPONENT          MESSAGE LEVEL
LDAP                   ERROR MESSAGES AND HIGHER
SAF                    WARNING MESSAGES AND HIGHER
DB                     INFORMATIONAL MESSAGES AND HIGHER
CORE                   WARNING MESSAGES AND HIGHER
PKID                   VERBOSE DIAGNOSTIC MESSAGES AND HIGHER
POLICY                 WARNING MESSAGES AND HIGHER
TPOLICY                WARNING MESSAGES AND HIGHER
MESSAGE LOGGING SETTING: STDOUT_LOGGING
CONFIGURATION FILE IN USE:
/etc/pkiserv/pkiserv.conf
TEMPLATE FILE IN USE:
/etc/pkiserv/pkiserv.tpl
CA CERTIFICATE FINGERPRINTS:
SHA1:  B4 6F EB 2E E0 96 9D 05 32 84 E3 2E D0 3C 10 6F 98 F5 43 3C
MD5:   C2 15 14 AC 12 81 59 46 09 F6 35 ED FB B6 CF 31
```

- When the PKI administrator receives a certificate request from a preregistered SCEP client, the PKI administrator can confirm the integrity of the certificate request by viewing its fingerprints on the “Single Request” Web page. (See Figure 22 on page 223 for a sample.)

To ensure the integrity of the certificate request, the PKI administrator can contact the SCEP requestor to match the fingerprints in the received certificate request with the fingerprints in the original certificate request. (The certificate requestor can use the SCEP client software to view the fingerprints saved for the original request.)

## Steps for enabling Simple Certificate Enrollment Protocol (SCEP)

**Before you begin:** The commands in the steps that follow include several variables that are described in Table 47. Determine the values for these variables and record the information in the blank boxes:

Table 47. Information you need to enable Simple Certificate Enrollment Protocol (SCEP)

Information needed	Where to find this information	Record your value here
<i>ca_label</i> —The label of your CA certificate in RACF.	See Table 11 on page 29.	



Table 47. Information you need to enable Simple Certificate Enrollment Protocol (SCEP) (continued)

Information needed	Where to find this information	Record your value here
<i>ra_label</i> —The label of your RA certificate in RACF.	See Table 11 on page 29.	
<i>ca_ring</i> —The PKI Services SAF key ring.	See Table 17 on page 36.	
<i>ca_expires</i> —The date the PKI Services CA certificate expires.	See Table 17 on page 36.	
<i>daemon</i> —The user ID for the PKI daemon.	See Table 17 on page 36.	
<i>ra_backup_dsn</i> —The name of the encrypted data set containing the backup copy of your new RA certificate and private key.	See Table 17 on page 36.	
<i>ra_dn</i> —The RA's distinguished name.	See Table 17 on page 36.	

Perform the following steps to enable PKI Services to process Simple Certificate Enrollment Protocol (SCEP) requests:

- (Optional) Create your PKI Services RA certificate by following these steps, if you haven't done so already. (This is optionally done by IKYSETUP.) If you already created an RA certificate, skip to Step 2.
  - To create an RA certificate, execute the following RACF command from the TSO command line
 

```
RACDCERT ID(daemon) GENCERT SUBJECTSDN(ra_dn) KEYUSAGE(HANDSHAKE)
SIGNWITH(CERTAUTH LABEL('ca_label') NOTAFTER(DATE(ca_expires))
WITHLABEL('ra_label'))
```
  - Backup the new PKI Services RA certificate and private key to a password-encrypted data set (*ra\_backup\_dsn*). **Important:** Remember to record and store your encryption password in case you ever need to recover the certificate or private key.
 

```
RACDCERT ID(daemon) EXPORT(LABEL('ra_label')) DSN(ra_backup_dsn)
FORMAT(PKCS12DER) PASSWORD('encryption-pw')
```
  - Add the new RA certificate to the PKI Services key ring.
 

```
RACDCERT ID(daemon) CONNECT(LABEL('ra_label') RING(ca_ring))
```
- Edit the PKI Services configuration file (/etc/pkiserv.conf) and set the RALabel directive in the **SAF** section to specify the label (*ra\_label*) of your PKI Services RA certificate. (The default in IKYSETUP is Local PKI RA. For details, see “(Optional) Steps for updating the configuration file” on page 51.)
 

**Example:**

```
[SAF]
KeyRing=PKISRVD/CARing
# The label of the PKI Services RA certificate
RALabel=Local PKI RA
```
- Edit the PKI Services configuration file (/etc/pkiserv.conf) to change the EnableSCEP directive in the **CertPolicy** section setting from **F(False)** to **T(True)**.

```
[CertPolicy]
# Enable the Simple Certificate Enrollment Protocol, (T)rue or (F)alse
EnableSCEP=T
```

4. Edit the PKI Services template file (/etc/pkiserv.tmpl) and customize the **<PREREGISTER>** section of the 5-Year SCEP Certificate – Preregistration template as desired or create a new preregistration template. (Refer to the list in “Variables used in the <PREREGISTER> section” on page 176 for valid variables and values.

**Example: (defaults)**

```
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
```

5. Edit the **<CONTENT>** section of your preregistration template to allow the PKI administrator to specify subject distinguished name and alternate name fields that the SCEP client must provide to authenticate. Specify only subject distinguished name and alternate name fields here. All other fields are ignored. (For about customizing the end-user Web pages, see Chapter 11, “Customizing the end-user Web application,” on page 91.)

**Example: (defaults)**

```
%%SerialNumber (Optional)%%
%%UnstructAddr (Optional)%%
```

6. Edit the **<CONSTANT>** section of your preregistration template to supply any other desired value, such as MAIL or ORG, that must be included for every SCEP preregistration request. Any subject distinguished name and alternate name fields you specify here must match the information (in the subsequent certificate request) sent by the SCEP client to authenticate the certificate request.

**Example:**

```
%%Org=The Firm%%
```

7. Stop and restart PKI Services.

**When you are done:** You have enabled your CA domain to accept SCEP preregistration requests and process certificate requests from preregistered SCEP clients.

## Using the PKI exit

### Programming Interface information

For the end-user functions except VERIFY, the PKISERV Web application CGIs support calling an installation-provided exit routine. The exit routine can perform tasks such as the following:

- Provide additional authorization checking
- Validate and change parameters

- Capture certificates for further processing.

PKI Services provides the following files for the exit. Both files are, by default, located in: `/usr/lpp/pkiserv/samples/`.

*Table 48. Summary of information about important files for the exit routine*

File name	Description
<code>pkixit.c</code>	Code sample for the exit (in the C programming language). You probably need to update the exit code before using it.
<code>Makefile.pkixit</code>	Makefile for <code>pkixit.c</code> .

If the exit exists, it must be a UNIX executable residing in the file system, and it must have appropriate permission assigned. To specify the exit, the UNIX programmer sets the `_PKISERV_EXIT` environment variable. (See page 347.) On input it receives standard UNIX parameters (that is, `argc` and `argv[]`). It communicates back to PKISERV through the return code and by writing to `STDOUT`.

## Steps for updating the exit code sample

To update the exit code sample, `pkixit.c`, perform the following steps:

1. Copy the sample exit and makefile to the current directory by entering the following commands:

```
cp /usr/lpp/pkiserv/samples/pkixit.c pkixit.c
cp /usr/lpp/pkiserv/samples/Makefile.pkixit Makefile
```

2. Compile and link to produce the executable, `pkixit`, by entering the following command:

```
make
```

3. Move the executable to its execution directory and set the permissions by entering the following commands:

```
mv pkixit /full-directory-name
chmod 755 /full-directory-name/pkixit
```

4. Edit the Web server's environment variables file by entering the following command:

```
oedit /etc/httpd.envvars
```

and add the environment variable `_PKISERV_EXIT` by adding the following line to the file:

```
_PKISERV_EXIT=/full-directory-name/pkixit
```

## Using the exit for pre- and post-processing

The exit is called:

- For preprocessing before calling the IRRSPX00 SAF callable service
- For post-processing after returning from the callable service.

## Advanced customization

The following table summarizes the values of the first two arguments for pre- and post-processing. (Additional arguments vary, depending on the function to perform.)

*Table 49. Values of arguments for pre- and post-processing*

Time of processing	Argument 1	Argument 2
Preprocessing	0	The function number from the SAF callable service in EBCDIC: <b>1</b> GENCERT <b>2</b> EXPORT <b>9</b> REQCERT <b>11</b> REVOKE <b>12</b> GENRENEW <b>13</b> REQRENEW
Post-processing	1	

**Note:** The parameters that are input to the CGIs and the values resolved by the CGIs (argument 3...argument *n* for all functions) will vary based on how you have customized the templates.

### Return codes

The sections that follow contain tables of expected return codes. If calling the exit produces an unexpected return code, that is, one that is not listed, PKI Services treats it as a failure. Processing for the request stops and an error message is issued.

## GENCERT and GENRENEW—preprocessing

**Purpose:** Provide additional authorization checking and parameter validation and modification.

**Arguments:**

*argument 3...argument n*

The parameters as input to the CGI plus values resolved by the CGI in *name=value* form, for example, "CommonName=Sam Smith".

**Return codes:**

Return code	Meaning
0	Continue with the request with possible modifications.
4	Continue with the request with possible modifications, but change it to require administrator approval.
>=8 <50	Deny the request and return to the caller immediately.

**STDOUT:** Zero or more additional *CertPlist* parameters to add to the request in *name=value* form, one per line. For those fields defined as non-repeating (according to the documentation for the IRRSPX00 callable service, for example, *CommonName*), specifying the parameters here in effect replaces the CGI input values.

### GENCERT and GENRENEW—post-processing

**Purpose:** Capture the TransactionId or failing return codes for further processing.

**Arguments:**

*argument 3...argument n-3*

The final set of parameters as determined by the preprocessing exit in *name=value* form.

*argument n-2*

The RACF return code from the callable service.

*argument n-1*

The RACF reason code from the callable service.

*argument n*

The *TransactionId*. This is a string of undetermined value if the request was unsuccessful.

**Return codes:**

Return code	Meaning
0	Normal

**STDOUT:** Optional replacement *TransactionId*.

## REQCERT and REQRENEW—preprocessing

**Purpose:** Provide additional authorization checking and parameter validation and modification.

**Arguments:**

*argument 3...argument n*

The parameters as input to the CGI plus values resolved by the CGI in *name=value* form, for example, "CommonName=Sam Smith".

**Return codes:**

Return code	Meaning
0	Continue with the request with possible modifications.
4	Continue with the request with possible modifications, but change it to not require administrator approval.
>=8 <50	Deny the request and return to the caller immediately.

**STDOUT:** Zero or more additional *CertPlist* parameters to add to the request in *name=value* form, one per line. For those fields defined as non-repeating (according to the documentation for the IRRSPX00 callable service, for example, *CommonName*), specifying the parameters here in effect replaces the CGI input values.

### REQCERT and REQRENEW—post-processing

**Purpose:** Capture the *TransactionId* or failing return codes for further processing.

**Arguments:**

*argument 3...argument n-3*

The final set of parameters as determined by the preprocessing exit in *name=value* form.

*argument n-2*

The RACF return code from the callable service.

*argument n-1*

The RACF reason code from the callable service.

*argument n*

The *TransactionId*. This is a string of undetermined value if the request was unsuccessful.

**Return codes:**

Return code	Meaning
0	Normal

**STDOUT:** Optional replacement *TransactionId*.



## EXPORT—preprocessing

**Purpose:** Provide additional authorization checking and parameter validation and modification.

**Arguments:**

*argument 3...argument n*

The parameters as input to the CGI in *name=value* form, for example, "TransactionId=12345".

**Return codes:**

Return code	Meaning
0	Continue with the export.
>=8 <50	Deny the request and return to the caller immediately.

**STDOUT:** Optional replacement *TransactionId* and *ChallengePassPhrase* parameters in *name=value* form, one per line. If these values are provided, they replace the user-provided values on the call to the SAF callable service. If *TransactionId* is specified without *ChallengePassPhrase*, the user-provided *ChallengePassPhrase* is used. If *ChallengePassPhrase* is specified without *TransactionId*, the user-provided *TransactionId* is used.

### EXPORT—post-processing

**Purpose:** Capture the certificate or failing return codes for further processing.

**Arguments:**

*argument 3...argument n-3*

The parameters as input to the CGI in *name=value* form, followed by any modified value provided by the preprocessing exit, also in *name=value* form.

*argument n-2*

The RACF return code from the callable service.

*argument n-1*

The RACF reason code from the callable service.

*argument n*

The base64-encoded certificate with header and footer. This is a string of undetermined value if the request was unsuccessful.

**Return codes:**

Return code	Meaning
0	Normal

**STDOUT:** Non-applicable.

## REVOKE—preprocessing

**Purpose:** Provide additional authorization checking and parameter validation.

**Arguments:**

*argument 3...argument n*

The parameters as input to the CGI in *name=value* form, for example, "reason=1".

**Return codes:**

Return code	Meaning
0	Continue with the request.
>=8 <50	Deny the request and return to the caller immediately.

**STDOUT:** Non-applicable.

### REVOKE—post-processing

**Purpose:** Capture the certificate or failing return codes or both for further processing.

**Arguments:**

*argument 3...argument n-2*

The parameters as input to the CGI in *name=value* form, for example, "reason=1".

*argument n-1*

The RACF return code from the callable service.

*argument n*

The RACF reason code from the callable service.

**Return codes:**

Return code	Meaning
0	Normal

**STDOUT:** Non-applicable.

## Scenarios for using the PKI exit

The sample PKI exit supplied with PKI Services, `pkixit.c`, written in the C language. It is intended to demonstrate the power of the exit and to provide a guide for you to write your own exit. The main routine of the program determines which subroutine to call, based on the `R_PKIServ` function being called and whether this is a pre- or post-processing call. The individual subroutines in the program handle the following scenarios:

### Scenario 1: Allow only selected users to request PKI browser certificates for authenticating to z/OS

This scenario is for allowing only selected local z/OS users to request PKI browser certificates for authenticating to z/OS. Additionally, this is for providing a customized TITLE value for the subject's distinguished name based on the user's role in the organization. Permission and the user's role in the organization is indicated by access to the BPX.SERVER resource in the FACILITY class and by the user's level of access to FACILITY class resources called PROJ.MEMBER and PROJ.PARTNER. The access values are as follows:

<b>NONE</b>	No access for either resource. The user is not permitted to request this type of certificate. The certificate request is denied.
<b>READ to PROJ.MEMBER</b>	The user is a team member and is permitted to request the certificate. The TITLE value is set to Team Member. Certificate requests for team members are automatically approved. (No administrator approval is required.)
<b>UPDATE to PROJ.MEMBER</b>	The user is the team's leader and is permitted to request the certificate. The TITLE value is set to Team Leader. A certificate request by the team leader is automatically approved. (No administrator approval is required.)
<b>READ to PROJ.PARTNER</b>	The user is considered to be a general partner of the team, not an active team member. The user is allowed to request certificates, but the requests require administrator approval before being issued. The TITLE value is set to Team Partner.
<b>UPDATE to PROJ.PARTNER</b>	The user is considered to be a trusted partner of the team, not an active team member. The user is allowed to request certificates, and unlike requests of the general partner, the certificate request are automatically approved. The TITLE value is set to Team Trusted Partner.

The preprocessing exit call for the GENCERT and REQCERT functions (subroutine `preProcessGenReqCertExit`) handles the logic described in the preceding. Here are the steps:

- The request values are passed into the exit through *argv* in *field-name=field-value* pairs, and the subroutine looks for the `Template=` and `UserId=` in the input parameters.
- When the exit code finds a `Template=` value containing PKI Browser Certificate For Authenticating To z/OS, the `__check_resource_auth_np()` system function examines the user ID. This determines the user's access to the preceding profiles.

- If the user has no authority to either of these resources, return code 8 is set. This causes the request to be denied.
- Otherwise the user's TITLE is set by writing the `TITLE=title-value` string to STDOUT.

By default, administrator approval is not required for the PKI browser certificate for authenticating to z/OS.

- When the user has only READ access to PROJ.PARTNER, the function must be changed to require administrator approval. This is done by setting return code 4.
- For all other accesses the function does not need to be changed.

### Scenario 2: Maintain a customized certificate repository (database) independent of PKI Services

This scenario is for maintaining a customized certificate repository (database) that is independent of PKI Services. After a successful submission of a certificate request, PKI Services returns the transaction ID. This is saved in a new customer-provided database entry. An alias for this database entry is then returned to the end user as the transaction ID. Later, when the user wishes to pick up the certificate, the user-entered alias name is used to retrieve the actual PKI Services transaction ID. The retrieved certificate is saved in the database entry before being returned to the user.

Three different exit calls handle the preceding logic.

- Post-processing for the GENCERT or REQCERT functions (subroutine `postProcessGenReqCertExit`) returns a pretend alias entry name by suffixing the actual transaction ID with either SAF or PKI. This is where the database entry should be created. (Note that the exit performs no actual database calls because this would be too customer-specific.)
- Preprocessing for the EXPORT function (subroutine `preProcessExportExit`) reverts the transaction ID to its original value. This emulates retrieval from the database entry.
- Post-processing for the EXPORT function (subroutine `postProcessExportExit`) saves the returned certificate to a database entry. This is emulated by writing it to a file.

### Scenario 3: Mandate a policy for certificate renewal only within 30 days of expiration

This scenario is for mandating a policy that allows users to renew their certificates only when certificates are within 30 days of expiring. When the condition is met, you can change the expiration date for the renew request so that the new certificate's validity period is extended by the number of days specified by the `NotAfter` parameter. In other words, the new certificate should expire  $n$  days from the current date, where  $n = \text{number of days left in the old certificate's validity period} + \text{number of days specified by NotAfter}$ .

The preprocessing exit call for GENRENEW and REQRENEW functions (subroutine `preProcessGenReqRenewExit`) handles the preceding logic. Here are the steps:

- The user's certificate is extracted from the environment variable `HTTPS_CLIENT_CERT`.
- The `NotAfter` value is extracted from the input parameters (`argv`), converted to a number, and saved in the variable `RequisitePro`.

- Subroutine `determineExpiration` is called to extract the expiration date from the user's certificate. This subroutine calls several lower subroutines to base64 decode the certificate, DER decode the binary certificate, and convert the expiration date to a seconds value.
- Upon return from `determineExpiration`, the variable *timeBeforeExp* is the number of seconds from now that the certificate expires. This is compared against the number of seconds in 30 days ( $86400 \times 30$ ) to see if it is greater than 30 days.
  - If it is greater than 30, the request is rejected by setting return code 8.
  - If it is not greater than 30, the new *NotAfter* value is computed as  $timeBeforeExp/86400 + requestPeriod$ .
- This new *NotAfter* value is set by writing it to STDOUT.

\_\_\_\_\_ **End of Programming Interface information** \_\_\_\_\_





---

## Part 4. Using PKI Services

This part explains how to use the PKI Services Web pages.

- Chapter 14, “Using the end-user Web pages,” on page 197 shows the Web pages for the end user and explains how to perform tasks such as requesting a certificate, obtaining the certificate, and renewing or revoking a certificate.
- Chapter 15, “Using the administration Web pages,” on page 217 shows the administration Web pages and explains how to process certificate requests and certificates.



## Chapter 14. Using the end-user Web pages

This chapter describes how the end user can use the PKI Services Web pages.

**Note:** The PKI Services Web pages in this chapter may differ slightly from those on the Web. If you need to see the exact content, view the pages on the Web. Additionally, the pages may contain differences depending on the browser you are using. This chapter assumes you are using Internet Explorer.

By default, the end user can:

- Install a CA certificate into the browser
- Request a new certificate
- Pick up a previously requested certificate
- Renew or revoke a previously issued browser certificate

The following table lists the types of certificates you can request:

*Table 50. Types of certificates you can request*

Type of certificate	Use
One-year PKI SSL browser certificate	End-user client authentication using SSL
One-year PKI S/MIME browser certificate	Browser-based e-mail encryption
Two-year PKI browser certificate for authenticating to z/OS	End-user client authorization using SSL when logging onto z/OS
Two-year PKI Authenticode—code signing server certificate	Software signing
Two-year PKI Windows logon certificate	End-user client authentication for an Active Directory user logging in to a Windows desktop using a smart card
Five-year PKI SSL server certificate	SSL Web server certification
Five-year PKI IPSEC server (firewall) certificate	Firewall server identification and key exchange
Five-year PKI intermediate CA server certificate	Subordinate (non-self-signed) certificate-authority certification
Five-year SCEP certificate	Creation of a preregistration record for certificate requestors. (Certificate requestors using Simple Certificate Enrollment Protocol (SCEP) must be preregistered.)
	Unlike other templates, this template is intended for administration use only.
<i>n</i> -year PKI browser certificate for extensions demonstration	Demonstration of all extensions supported by PKI Services
One-year SAF browser certificate	End-user client authentication where the security product (RACF, not PKI Services) is the certificate provider
One-year SAF server certificate	Web server SSL certification where the security product (RACF, not PKI Services) is the certificate provider

## Using the end-user Web pages

**Note:** If your installation has not customized the certificate templates, the PKI Services Web pages in this chapter may still differ slightly from those on the Web; if your installation customized the templates, the Web pages in this chapter may differ greatly from those you view on the Web.

### Special consideration about using SAF templates:

The templates that control processing of the SAF certificates listed in Table 50 on page 197 perform only a subset of the function available natively in RACF through the RACDCERT TSO command or the ISPF panels. They are provided to enable a web interface for requesting certificates from RACF for browsers and off-platform servers. They are not intended to be a complete replacement for RACF certificate function.

**Restriction:** If you wish to generate a certificate for a server running on the local z/OS system (in other words, for a system using the RACF database where the signing certificate resides), do not use the “One-year SAF server certificate” template. Instead, use the RACDCERT TSO command or ISPF panels directly. Using the “One-year SAF server certificate” template may cause the loss of the private key if the authenticating user ID is not the same as the user ID specified when generating the certificate request in RACF.

---

## Steps for accessing the end-user Web pages

Perform the following preliminary steps to access the PKI Services Web pages:

1. Get your organization’s URL for accessing the PKI Services Web pages. Enter this URL in your browser. This takes you to the end-user “PKI Services Certificate Generation Application” Web page, shown in the following figure:

## PKI Services Certificate Generation Application

[Install the CA certificate to enable SSL sessions for PKI Services](#)

**Choose one of the following:**

- ♦ **Request a new certificate using a model**

Select the certificate template to use as a model 1-Year PKI SSL Browser Certificate ▼
- ♦ **Pick up a previously requested certificate**

Enter the assigned transaction ID

Select the certificate return type PKI Browser Certificate ▼
- ♦ **Renew or revoke a previously issued browser certificate**

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 6. PKI Services end-user home page for certificate generation

2. If this is the first time you have accessed the forms on these Web pages, you must install the CA certificate into your browser. Click the **Install the CA certificate** link and follow the directions.

The following is a sample of the directions to follow for installing the CA certificate on Internet Explorer:

- a. After you click the **Install the CA certificate** link, a popup window called "File download" appears. Make sure the "Open this file from its current location" radio button is selected (rather than "Save this file to disk"). Then click the **OK** button. The following is an example of the popup window you might see, depending on the CA certificate you have installed.



Figure 7. The certificate popup window for installing the CA certificate

- b. Click the **Install certificate** button. (This initiates a series of pop-ups in which you need to click **Next** buttons and finally the **Finish** button, culminating in a popup window that says “The import was successful”.)

---

You are now ready to perform tasks, such as:

- Requesting a new certificate
- Picking up a previously requested certificate
- Renewing or revoking a previously issued browser certificate

---

## Summary of fields

When you request certificates, you provide information for the fields in certificate request forms. The following table describes the fields in the end-user Web pages:

Table 51. Summary of fields in end-user Web pages

Field	Description
Base64-encoded PKCS #10 certificate request	<p>(This is for server or device enrollment only.) You create a certificate request on behalf of another server (which could be a z/OS server or other type of server) or device for which you are requesting a certificate. You use software specific to that server to generate the PKCS #10 request before going to the PKI Services Web site. Save the request in a file. Then open the file in a text editor such as Windows Notepad and copy and paste the contents into the text box on the enrollment form. A text area of 70 columns and 12 rows is allocated for this certificate request. Here is an example of the certificate request:</p> <pre> -----BEGIN NEW CERTIFICATE REQUEST----- MIIBiDCB8gIBADAZMRcwFQYDVQQDEw5Kb2huIFEuIFB1YmtpYzCBnzANBgkqhkiG 9w0BAQEFAAOBjQAwgYkCgYEASt1cJHAGPqi6QjAyL+xNbt8z5ngmvq02V003oYu /mEnQtRM96e+2jbmDCRo5tWVklG40Yf9ZVB5biURMJFLztfa4AVdEVtun8DH2pwc wiNIZZcC1Zym5adurUmyDk64PgiiIPMQS/t0ttG4c5U8uWSK0b1J4V4f7ps+t1aG t+cAwEAAaAwMC4GCSqGSIb3DQEJDDjEhMB8wHQYDVRO0BBYEFAIKTovBBvnFqDA0 1oIhtRinwRC9MA0GCSqGSIb3DQEBBQUAA4GBAIBCVpwYvppIX3HHmpKZPNY8Snsz AJrDsgAEH51W0IRGywhqKcLLxa9htoQai6cdc8RpFVTwk6UfdC0GxMn4aFb34Tk3 5WYdz0iHXg8MhHiB3EruwdWs+S7Fv3JhU3FLwU6lFLfAjbvi+35iEWQym0R6mE5W CathprmGfKR5DE5E -----END NEW CERTIFICATE REQUEST----- </pre> <p>For a sample of the enrollment form showing the text box for a PKCS #10 request, see Figure 9 on page 206.</p>
Challenge passphrase	This is the passphrase you entered when requesting a certificate. You type the same passphrase, exactly as you typed it on the request form. This is a case-sensitive text field of up to 32 characters.
Common name	<p>Your name, such as John Smith. (You can use your first and last name, in that order.) This is a text field of up to 64 characters. (See Note 1.)</p> <p>For SSL servers, the common name is the server's fully qualified domain name, for example, <a href="http://www.ibm.com">www.ibm.com</a>.</p>
Country	The country where your organization is located. This is a 2-character text field. (See Note 1.)
Cryptographic service provider	(This is for the Internet Explorer browser only.) The cryptographic service provider to generate your public/private key pair. You select a value from the drop-down list. Larger keys are more secure, but they also increase the time that is needed for connecting to a secure session.
Domain name	<p>Domain name for alternate name. This is the host name of the machine where a certificate will be installed. This is a text field of up to 100 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>
E-mail address for distinguished name	<p>E-mail address for the distinguished name. This is a text field of up to 64 characters. (See Note 1.)</p> <p><b>Restriction:</b> If you specify a value for this parameter and for Notification e-mail address, the two values <i>must</i> be the same.</p>
E-mail address for alternate name	<p>E-mail address for alternate name, including the @ character and any periods (.). This is a text field of up to 100 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>

## Using the end-user Web pages

Table 51. Summary of fields in end-user Web pages (continued)

Field	Description																														
Extended key usage	<p>This indicates the intended purpose of the certificate. Possible values are:</p> <table><tr><td><b>clientauth</b></td><td>Client side authentication</td></tr><tr><td><b>codesigning</b></td><td>Code signing</td></tr><tr><td><b>emailprotection</b></td><td>Email protection</td></tr><tr><td><b>mssmartcardlogon</b></td><td>Smart card logon for Microsoft Windows users</td></tr><tr><td><b>ocspsigning</b></td><td>OCSP response signing</td></tr><tr><td><b>serverauth</b></td><td>Server side authentication</td></tr><tr><td><b>timestamping</b></td><td>Digital timestamping</td></tr></table>	<b>clientauth</b>	Client side authentication	<b>codesigning</b>	Code signing	<b>emailprotection</b>	Email protection	<b>mssmartcardlogon</b>	Smart card logon for Microsoft Windows users	<b>ocspsigning</b>	OCSP response signing	<b>serverauth</b>	Server side authentication	<b>timestamping</b>	Digital timestamping																
<b>clientauth</b>	Client side authentication																														
<b>codesigning</b>	Code signing																														
<b>emailprotection</b>	Email protection																														
<b>mssmartcardlogon</b>	Smart card logon for Microsoft Windows users																														
<b>ocspsigning</b>	OCSP response signing																														
<b>serverauth</b>	Server side authentication																														
<b>timestamping</b>	Digital timestamping																														
HostIdMappings extension	<p>This is the user ID for authorization purposes in the format: subject-id@host-name</p> <p><b>Example:</b> DSmith@ibm.com</p> <p>This is a text field of up to 100 characters.</p>																														
IP address	<p>The IP address for the alternate name. This unique IP version 4 address specifies the location of each device or workstation on the Internet, for example, 9.67.97.103. (PKI Services supports only IP version 4 addresses.) The IP address is in dotted decimal format and is a text field of up to 15 characters.</p> <p><b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.</p>																														
Key protection	<p>(This is for the Internet Explorer browser only.) This asks if you want to enable private key protection. (The drop-down choices are <b>Yes</b> and <b>No</b>.)</p>																														
Key size	<p>(This is for the Netscape browser only.) This is the key size for your public/private key pair. Select a value from the drop-down list. Larger keys are more secure, but they also increase the time needed for connecting to a secure session.</p>																														
Key usage	<p>The intended purpose of the certificate. Each possible value is shown here with its intended purpose and possible PKIX bits:</p> <table><tr><th>KeyUsage value</th><th>Intended purpose</th><th>PKIX bits</th></tr><tr><td><b>certsign</b></td><td>Certificate and CRL signing</td><td>keyCertSign and cRLSign</td></tr><tr><td><b>crlsign</b></td><td>CRL signing</td><td>cRLSign</td></tr><tr><td><b>dataencrypt, dataencipherment, or dataenciph</b></td><td>Data encryption</td><td>dataEncipherment</td></tr><tr><td><b>digitalsig or digitalsignature</b></td><td>Authentication</td><td>digitalSignature</td></tr><tr><td><b>docsign or nonrepudiation</b></td><td>Document signing</td><td>nonRepudiation</td></tr><tr><td><b>handshake</b></td><td>Protocol handshaking (for example, SSL)</td><td>digitalSignature and keyEncipherment</td></tr><tr><td><b>keyagree or keyagreement</b></td><td>Key agreement</td><td>keyAgreement</td></tr><tr><td><b>keycertsign</b></td><td>Certificate signing</td><td>keyCertSign</td></tr><tr><td><b>keyencrypt, keyencipherment, or keyenciph</b></td><td>Key transport</td><td>keyEncipherment</td></tr></table>	KeyUsage value	Intended purpose	PKIX bits	<b>certsign</b>	Certificate and CRL signing	keyCertSign and cRLSign	<b>crlsign</b>	CRL signing	cRLSign	<b>dataencrypt, dataencipherment, or dataenciph</b>	Data encryption	dataEncipherment	<b>digitalsig or digitalsignature</b>	Authentication	digitalSignature	<b>docsign or nonrepudiation</b>	Document signing	nonRepudiation	<b>handshake</b>	Protocol handshaking (for example, SSL)	digitalSignature and keyEncipherment	<b>keyagree or keyagreement</b>	Key agreement	keyAgreement	<b>keycertsign</b>	Certificate signing	keyCertSign	<b>keyencrypt, keyencipherment, or keyenciph</b>	Key transport	keyEncipherment
KeyUsage value	Intended purpose	PKIX bits																													
<b>certsign</b>	Certificate and CRL signing	keyCertSign and cRLSign																													
<b>crlsign</b>	CRL signing	cRLSign																													
<b>dataencrypt, dataencipherment, or dataenciph</b>	Data encryption	dataEncipherment																													
<b>digitalsig or digitalsignature</b>	Authentication	digitalSignature																													
<b>docsign or nonrepudiation</b>	Document signing	nonRepudiation																													
<b>handshake</b>	Protocol handshaking (for example, SSL)	digitalSignature and keyEncipherment																													
<b>keyagree or keyagreement</b>	Key agreement	keyAgreement																													
<b>keycertsign</b>	Certificate signing	keyCertSign																													
<b>keyencrypt, keyencipherment, or keyenciph</b>	Key transport	keyEncipherment																													
Label	<p>The label assigned to the requested certificate. This is a text field of up to 32 characters.</p>																														
Locality	<p>The city or municipality where your organization is located, such as Pittsburgh or Paris. This is a text field of up to 64 characters. (See Note 1.)</p>																														
Not before (date)	<p>A number of days, added to the current date (by default, you can select either 0 or 30), before which the certificate is not valid.</p>																														
Not after (date)	<p>A number of days, added to the current date, after which the certificate expires. By default, you can select either one year or two years for the time at which the certificate expires.</p>																														



Table 51. Summary of fields in end-user Web pages (continued)

Field	Description
Notification e-mail address	E-mail address for notification purposes. This is a text field of up to 64 characters. <b>Note:</b> If you specify a value for this parameter and for E-mail address for distinguished name, the two values <b>must be</b> the same.
Organization	The legally registered name (or trademark name, for example, IBM) of your organization. This is a text field of up to 64 characters. (See Note 1.)
Organizational unit	The name of your division or department. (There can be more than one organizational unit field on a request form. For example, one could be for your department and another for your division.) This is a text field of up to 64 characters. (See Note 1.)
OtherName	Additional identifier for the alternate name. See your PKI Services administrator for information about this field.
Passphrase	You decide this value when requesting a certificate (and must later supply this value when retrieving the certificate). You enter and then reenter this when requesting a certificate. This is a case-sensitive text field of up to 32 characters. (There is no minimum number of characters, and you can use any characters, but alphanumeric characters (A–Z, a–z, and 0–9) are suggested.
Postal code	Your postal code or zip code. This is a text field of up to 64 characters. (See Note 1.)
State or Province	The state or province where your organization is located. Your registration policies determine whether you spell out the full name of the state or province or use an abbreviation. This is a text field of up to 64 characters. (See Note 1.)
Street	Your street address. This is a text field of up to 64 characters. (See Note 1.)
Title	Your job title. This is a text field of up to 64 characters. (See Note 1.)
Transaction ID	This is assigned after you request your certificate. When it is displayed, you need to record this number. This is a text field of up to 56 characters.
Uniform resource identifier (URI)	Uniform resource identifier for the alternate name. This is a name or address referring to an Internet resource; a URL is one kind of uniform resource identifier. This is a text field of up to 100 characters. <b>Note:</b> The value is one of the list of subject's alternate names that is saved in the subject alternate name extension in the certificate.
Your name	Your name (for tracking purposes). This can be in any format, for example, John Smith or John. J. Smith. This is a text field of up to 32 characters.

**Note:**

1. The value for this field is one of the relative distinguished names that is saved in the subject's distinguished name in the certificate.

For server certificates where a base64-encoded PKCS #10 certificate request is also supplied, specify this field only if you wish to change the distinguished name supplied in the PKCS #10 certificate request. In this case, you must respecify the entire distinguished name (all fields) as desired.

The subject's distinguished name specified in the PKCS #10 certificate request is ignored if a value is supplied for any of the following fields on the certificate request Web page:

- Common name
- Country
- E-mail address for distinguished name
- Locality
- Organization
- Organizational unit
- Postal code
- State or Province
- Street
- Title

---

## Steps for requesting a new certificate

To request a new certificate, first go to the PKI Services home page. (See Figure 6 on page 199.)

Perform the following steps to request a new certificate:

1. Click the down arrow to the right of the field beside Request a new certificate using a model. This displays a list of certificate templates from which you can select.

**For SCEP preregistration:** Do not follow these steps to request a SCEP (preregistration) certificate template. Instead, go to “Steps for preregistering a SCEP client” on page 214.

The following list shows the certificate templates that PKI Services provides by default. This list may differ from the certificate templates your installation provides because your installation can customize the certificate templates and Web pages.

- One-year SAF server certificate
- One-year SAF browser certificate
- One-year PKI SSL browser certificate (See Figure 8 on page 205 to see a sample of this Web page.)
- One-year PKI SSL S/MIME browser certificate
- Two-year PKI browser certificate for authenticating to z/OS
- Two-year PKI Authenticode—code signing server certificate
- Two-year PKI Windows logon certificate
- Five-year PKI SSL server certificate
- *n*-year PKI browser certificate for extensions demonstration
- Five-year SCEP certificate - Preregistration
- Five-year PKI IPSEC server (firewall) certificate
- Five-year PKI intermediate CA server certificate

2. Click one of the items in the list. The drop-down list then collapses so that only the certificate you selected appears in the field and is highlighted.

3. Click the **Request certificate** button. A form where you fill in information is displayed.

**Note:** You may need to click through some additional panels specific to your browser (for example, clicking **Next** on Netscape or answering Do you want to proceed? on Internet Explorer) before the certificate request form appears.

4. Fill in the necessary information in the certificate request form.

The form that appears depends on the certificate you are requesting and, in some instances, the fields that appear on the form depend on the browser you are using. **Example:** If you request a one-year SSL browser certificate, the following form appears. (See Figure 8 on page 205.)

## 1-Year SSL Browser Certificate

Choose one of the following:

- **Request a New Certificate**

Enter values for the following field(s)

Your name for tracking this request (optional)

Email address for distinguished name (optional)

Common Name

Email address for notification purposes (optional)

Pass phrase for securing this request. You will need to supply this value when retrieving your certificate

Reenter your pass phrase to confirm

Select the following key information

Cryptographic Service Provider

Enable strong private key protection?

- **Pick Up a Previously Issued Certificate**

email: [webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 8. One-year SSL browser certificate request form

**Note:** In the case of the one-year SSL browser certificate, fill in your common name. (See Table 51 on page 201 for descriptions of fields.) If you are using Netscape, select a key size from a drop-down list. Alternately, if you are using Internet Explorer, click the drop-down lists to select your cryptographic service provider and to specify whether to use strong private key protection.

5. If you are requesting a server or device certificate, you will need to supply a base64-encoded PKCS #10 certificate request. Use software specific to that server to generate the PKCS #10 request before going to the PKI Web site. Paste the request into the Web page as shown in Figure 9 on page 206.

## 5-Year PKI SSL Server Certificate

Choose one of the following:

- Request a New Certificate**

Enter values for the following field(s)

Your name for tracking this request (Optional)

Email address for distinguished name (Optional)

Common Name (Optional)

Organizational Unit (Optional)

Organizational Unit (Optional)

Organization (Optional)

Street address (Optional)

Locality (Optional)

State or Province (Optional)

Zipcode or postal code (Optional)

Country (Optional)

Email address for alternate name (Optional)

Domain name for alternate name (Optional)

Uniform Resource Identifier for alternate name (Optional)

IP address for alternate name in dotted decimal form (Optional)

Email address for notification purposes (Optional)

Pass phrase for securing this request. You will need to supply this value when retrieving your certificate

Reenter your pass phrase to confirm

Base64 encoded PKCS#10 certificate request

Submit certificate request

Clear
- Pick Up a Previously Issued Certificate**

Retrieve your certificate

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 9. Supplying the PKCS #10 certificate request for a server or device certificate

For server certificates where a base64-encoded PKCS #10 certificate request is also supplied, specify the following fields only if you wish to change the distinguished name supplied in the PKCS #10 certificate request. In this case, you must respecify the entire distinguished name (all fields) as desired. The name specified in the PKCS #10 certificate request is ignored if a value is supplied for any of the following fields on the certificate request Web page:

- Common name
- Country
- e-mail address for distinguished name
- Locality
- Organization
- Organizational unit
- Postal code
- State or Province
- Street
- Title

6. Fill in the passphrase on the certificate request form (twice). This is a value known only to you. Pick a value that you can easily remember because you will be challenged to supply the same passphrase when you pick up your certificate. Do not use a sensitive value such as your ATM pin or login password.
7. Fill in any optional information as desired. When you are satisfied with the information you have entered, click the **Submit certificate request** button. If the request is successful, you see a page like the one shown in Figure 10, which tells you your transaction ID.

### Request submitted successfully

Here's your transaction ID. You will need it to retrieve your certificate. Press 'Continue' to retrieve the certificate.

1jL+PdXDzuFOVknDWBrf3ls+

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 10. Successful request displays transaction ID

- a. Make a note of the transaction ID. (You can copy and paste the transaction ID to a file so that you have it for future reference, or you can write it in the box below. The reason for keeping a record of the transaction ID is that, depending on how you go to the Web page to retrieve your certificate (see Figure 11 on page 208), you may have to fill in the transaction ID on that Web page.)

<b>Transaction ID:</b>	
------------------------	--

- b. Click the **Continue** button. This displays the following Web page:

### Retrieve Your 1-Year PKI SSL Browser Certificate

**Please bookmark this page**

Since your certificate may not have been issued yet, we recommend that you create a bookmark to this location so that when you return to this bookmark, the browser will display your transaction ID. This is the easiest way to check your status.

Enter the assigned transaction ID

If you specified a pass phrase when submitting the certificate request, type it here, exactly as you typed it on the request form

**To check that your certificate installed properly, follow the procedure below:**

**Netscape V6** - Click Edit->Preferences, then Privacy and Security-> Certificates. Click the Manage Certificates button to start the Certificate Manager. Your new certificate should appear in the Your Certificates list. Select it then click View to see more information.

**Netscape V4** - Click the Security button, then Certificates-> Yours. Your certificate should appear in the list. Select it then click Verify.

**Internet Explorer V5** - Click Tools->Internet Options, then Content, Certificates. Your certificate should appear in the Personal list. Click Advanced to see additional information.

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 11. Web page to retrieve your certificate

- c. Bookmark this Web page.

**Notes:**

- 1) After you submit the request for a certificate, your PKI Services administrator may need to approve the request before you can pick up your certificate. The amount of time that this takes can vary from a few minutes to a few days, depending on your installation. You bookmark this Web page so that you can return to it at a later time.
  - 2) If your installation has enabled e-mail notification and you supplied a valid e-mail address when submitting your certificate request, then you will receive an e-mail message when your certificate is ready for pick-up or if the PKI Services rejects your certificate request.
- d. From this Web page, you can start the steps to retrieve your certificate (see "Steps for retrieving your certificate from the bookmarked Web page" on page 209) or you can return to the PKI Services home page (by clicking the **Home** button).

---

## Retrieving your certificate

You can retrieve your certificate:

- From Web page you bookmarked in Step 7c. (This Web page contains your transaction ID, so you do not have to enter it.) The steps that follow are for retrieving your certificate from the bookmarked Web page.

- From the PKI Services home page. (See Figure 6 on page 199 and “Steps for retrieving your certificate from the PKI Services home page” on page 210.)

If your company has enabled e-mail notification for non-SAF certificates and you supplied a valid e-mail address when submitting your certificate request, you will receive an e-mail to notify when your certificate is ready for retrieval (or if your certificate request has been rejected).

## Steps for retrieving your certificate from the bookmarked Web page

Perform the following steps to retrieve your certificate from the bookmarked Web page:

1. Go to the bookmarked Web page. (See Figure 11 on page 208.)
2. If you entered a passphrase when requesting your certificate, enter the passphrase.
3. Click the **Retrieve and install certificate** button. If you are using Netscape, go to Step 5 on page 210. If you are using Internet Explorer and the retrieval of a certificate is successful, this displays the Web page shown in Figure 12. (This is for a browser certificate. For a server certificate, Figure 13 shows an example of the Web page.)

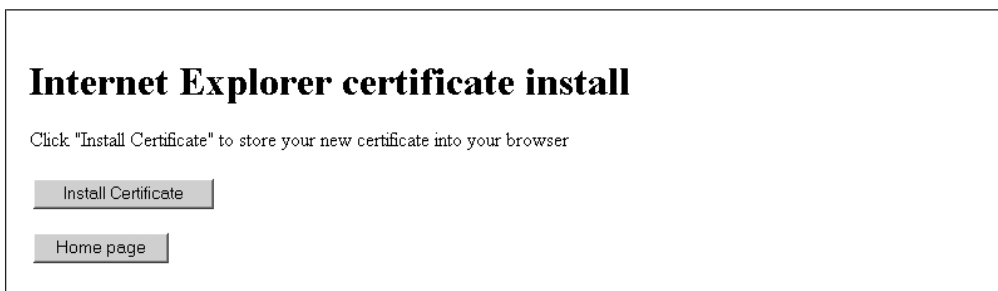


Figure 12. Browser certificate installation Web page



Figure 13. Server certificate installation Web page

4. Click the **Install certificate** button. If the certificate installs successfully, you get a popup window that says Your new certificate installed successfully.

- 
5. Check that your certificate installed correctly:
    - For Netscape, click the **Security** button, then Certificates → Yours. Your certificate should appear in the list. Select it and click Verify.
    - For Internet Explorer, Click Tools → Internet Options, then Content, **Certificates**. Your certificate should appear in the Personal list. Click Advanced to see additional information.
- 

## Steps for retrieving your certificate from the PKI Services home page

**Before you begin:** To retrieve your certificate from the PKI Services home page, you must first know your transaction ID. You should have recorded this when your certificate request was successful. (See Figure 10 on page 207.)

Perform the following steps to retrieve your certificate from the PKI Services home page:

1. Enter your transaction ID and select the certificate type using the drop-down. Then click the **Pick up certificate** button on the PKI Services home page. (See Figure 6 on page 199.) This displays the Web page that Figure 11 on page 208 displays.
  2. Enter your passphrase (this is the challenge passphrase) if you specified one when requesting your certificate.
  3. Click the **Retrieve and install certificate** button. If you are using Netscape, go to Step 5. If you are using Internet Explorer and the retrieval of the certificate is successful, this displays the Web page that Figure 12 on page 209 shows. (This is for a browser certificate. For a server certificate, Figure 13 on page 209 shows an example of the Web page.)
  4. Click the **Install certificate** button. If the certificate installs successfully, you get a popup window that says Your new certificate installed successfully.
  5. Check that your certificate installed correctly:
    - For Netscape, click the **Security** button, then Certificates → Yours. Your certificate should appear in the list. Select it and click Verify.
    - For Internet Explorer, Click Tools → Internet Options, then Content, Certificates. Your certificate should appear in the Personal list. Click Advanced to see additional information.
- 

## Steps for renewing a certificate

Perform the following steps to renew a certificate:

1. On the PKI Services home page (see Figure 6 on page 199), click the **Renew or revoke certificate** button. This displays a popup window with a list of certificates, such as the following figure shows:





Figure 14. Popup window listing certificates

2. The popup window may list more than one certificate. The certificates are listed by nickname in the order they are installed in the browser. Therefore, you may not be able to identify the PKI Services certificate you want to renew. Highlight the entry you think is the right one and click the **OK** button. If the certificate you selected is one that PKI Services issued and it is not expired or revoked, the following is an example of the Web page you might see, depending on the certificate you are renewing:

## Renew or Revoke a Browser Certificate

Here is the certificate you selected:

<b>Requestor:</b>	W.M. Fortey	<b>Created:</b>	2003/07/21
<b>Status:</b>	Active	<b>Modified:</b>	2003/07/21
<b>Template:</b>	1-Year PKI SSL Browser Certificate		
<b>Serial #:</b>	19093		

---

<b>Subject:</b>	MAIL=forteywm@somecompany.com,CN=W.M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm		
<b>Issuer:</b>	CN=Bank XYZ Identrus Certificate Authority,OU=Bank XYZ Identrus Authority,O=Bank XYZ		
<b>Validity:</b>	2003/07/21 00:00:00 - 2004/07/19 23:59:59		
<b>Usage:</b>	handshake(digitalSignature, keyEncipherment)		
<b>Extended Usage:</b>	clientauth		

If this is the correct certificate, choose one of the following:  
(otherwise you need to restart your browser to pick another certificate)

- Renew the above certificate**

Email address for notification purposes (optional)

Pass phrase for securing this request. You will need to supply this value when retrieving your certificate

Reenter your pass phrase to confirm
- Revoke the above certificate**
- Suspend the above certificate**

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 15. Renew or revoke a certificate Web page

**Notes:**

- If this is not the PKI Services certificate you want to renew, you need to close your browser (because the browser caches information) before again clicking the **Renew or revoke certificate** button as in Step 1 on page 210.
- If the certificate has the MAIL attribute in the subject's distinguished name, the value of NotifyEmail must match it.

- Under the Renew the above certificate section, enter your passphrase in the two fields requesting it.

4. Click the **Renew** button.

---

5. If the renewal request is successful, this displays a Web page that says Request submitted successfully and displays the transaction ID. Click the **Continue** button on this Web page.

---

6. This takes you the Web page from which you retrieve your certificate. (See Figure 11 on page 208 for an example of this Web page and “Steps for retrieving your certificate from the bookmarked Web page” on page 209 for the directions to follow.)

---

## Steps for revoking or suspending a certificate

Revoking or suspending a certificate means that you cannot continue to use the certificate. You might want to permanently *revoke* your certificate if you suspect your private key has been compromised. You might want to *suspend* (temporarily revoke) your certificate if you want to discontinue using it for a period of time (known as the suspension *grace period*).

If you suspend your certificate, the PKI administrator may *resume* (reactivate) the certificate, or permanently revoke it, if the certificate has not yet expired and the grace period has not elapsed. If the grace period has elapsed, the certificate is permanently revoked the next time the certificate revocation lists (CRLs) are issued.

Perform the following steps to revoke or suspend a certificate:

1. On the PKI Services home page (see Figure 6 on page 199), click the **Renew or revoke certificate** button. This displays a popup window with a list of certificates, as in Figure 14 on page 211.

---

2. The popup window may list more than one certificate. The certificates are listed by nickname in the order they are installed in the browser. You may not be able to identify the PKI Services certificate you want to revoke or suspend. Highlight the entry you think is the right one and click the **OK** button. If the certificate you selected is one that PKI Services issued and it is not expired or revoked, this displays the “Renew or revoke a browser certificate” Web page. (See “Steps for renewing a certificate” on page 210.)
 

**Note:** If this is not the PKI Services certificate you want to revoke or suspend, you need to close your browser before again clicking the **Renew or revoke certificate** button as in Step 1 on page 210.

---

3. Make sure the certificate you want is the one described at the top of the Web page. Click the **Revoke** or **Suspend** button. Note that when you revoke a certificate, you can click the drop-down list (of reasons) to select a reason if you wish.

---

4. This displays a Web page that says Request submitted successfully. You can click the **Home page** button to return to the PKI Services home page.

---

**When you are done:** You will no longer be able to use the certificate. If you suspended the certificate, contact your PKI administrator when you wish to have it resumed.

---

### Steps for preregistering a SCEP client

These steps are performed by the PKI administrator using the end-user Web pages. To preregister a SCEP client, first go to the PKI Services home page. (See Figure 6 on page 199.)

Perform the following steps to complete a preregistration request and preregister a SCEP client:

1. Click the down arrow to the right of the field beside Request a new certificate using a model. (This displays a list of certificate templates from which you can select.) Select 5-Year SCEP Certificate - Preregistration from the list. The drop-down list then collapses so that only the preregistration template appears in the field and is highlighted.
2. Click the **Request certificate** button. A form where you fill in information is displayed.
3. The preregistration form appears. (See Figure 16.) Fill in the necessary information in the preregistration form for this SCEP client. The form that appears depends how this template is customized for your CA domain.

**5-Year SCEP Certificate - Preregistration**

- Enter values the client must provide to authenticate

The name of the person or device that the certificate represents

Pass phrase for securing this request. You will need to supply this value when retrieving your certificate

Reenter your pass phrase to confirm

Device serial number (Optional)

Unstructured device address (Optional)


email: [webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 16. SCEP preregistration request form

4. Fill in the passphrase on the certificate request form (twice). This SCEP client must match this passphrase to successfully request a certificate. Do not use a sensitive value such as your ATM pin or login password.

5. Fill in any optional information as needed. This SCEP client must match all subject and alternate name information you enter to successfully request a certificate. When you are satisfied with the information you have entered, click the **Submit certificate request** button.

If your preregistration request is successful, you see a page like the one shown in Figure 17, which tells you your temporary transaction ID.



**Preregistration successful**

Here's the temporary transaction ID so you may locate the preregistration record: 1j5NqaAnB1q+VknudWBrf3kI+

[Examine Preregistration Record](#)

Press 'Preregister' to preregister another client using the same template.

[Preregister](#)

[Administration Home Page](#)

[Home Page](#)

email: webmaster@your-company.com

Figure 17. Successful preregistration request

6. (Optional) Click the **Examine Preregistration Record** button.
7. (Optional) Click the **Preregister** button to preregister another client using the same template.

**When you are done:** You have successfully preregistered a SCEP client. Return to the PKI Services home page (by clicking the **Home Page** button) or return to the administration home page (by clicking the **Administration Home Page** button).



---

## Chapter 15. Using the administration Web pages

This chapter presents background information about certificate requests and certificates and explains how the administrator can use the administration Web pages to perform the following tasks:

- Process a certificate request
  - Approve a request without making changes
  - Approve a request with changes
  - Reject a request
  - Delete a request
- Process a preregistration record
  - Delete a certificate request from a preregistered client  
(For information about preregistering clients, see “Steps for preregistering a SCEP client” on page 214.)
- Process a certificate
  - Revoke a certificate
  - Suspend a certificate
  - Resume a certificate
  - Delete a certificate
- Perform searches for certificate requests, certificates, and preregistration records

**Note:** The PKI Services Web pages in this chapter may differ slightly from those on the Web. If you need to see the exact content, view the pages on the Web. Additionally, the pages may contain differences depending on the browser you are using. This chapter assumes you are using Internet Explorer.

---

### Steps for accessing the administration home page

Perform the following preliminary steps to access the administration home page:

1. Get your organization’s URL for accessing the PKI Services Web pages. Enter this URL in your browser. This takes you to the PKI Services administration start page.  
  
The administration start page contains a button for the administration home page (see the default “PKI Services Administration” home page in Figure 21 on page 222) and a button for each end-user home page (see the default “PKI Administrators Start Page” in Figure 18 on page 218).

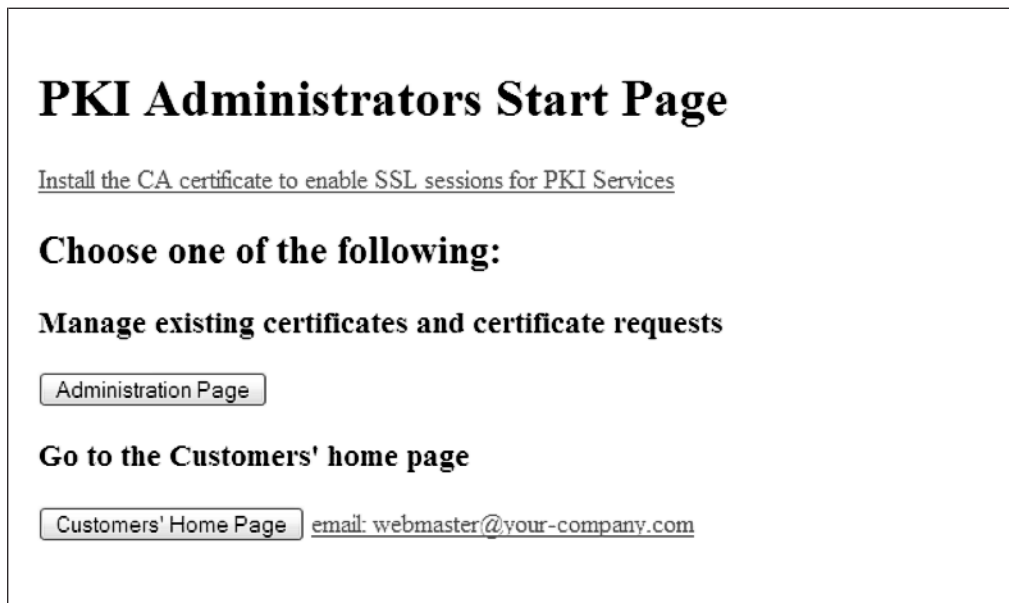


Figure 18. PKI Services administration start page

2. If this is the first time you have accessed these Web pages, you must install the CA certificate into your browser. Click the **Install the CA certificate** link and follow the directions.  
The following is a sample of the directions to follow for installing the CA certificate on Internet Explorer:
  - a. After you click the **Install the CA certificate** link, a popup window called "File download" appears. Make sure the "Open this file from its current location" radio button is selected (rather than "Save this file to disk"). Then click the **OK** button. The following is an example of the popup window you might see, depending on the CA certificate you have installed:



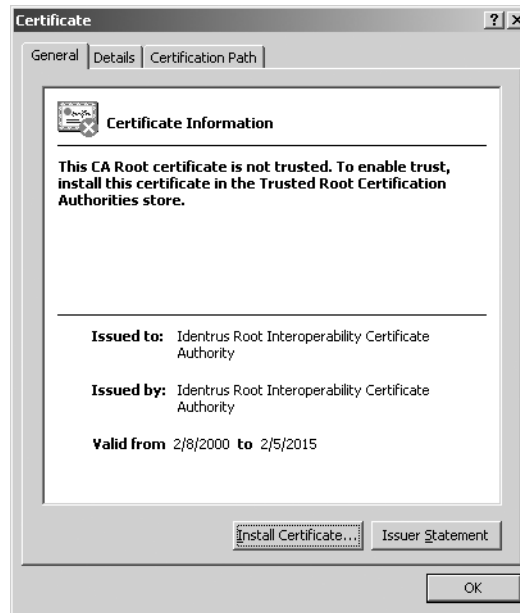


Figure 19. The certificate popup window for installing the CA certificate

- b. Click the **Install certificate** button. (This initiates a series of pop-ups in which you need to click **Next** buttons and finally the **Finish** button, culminating in a popup window that says “The import was successful”).
- 
3. Click the **Go to administration page** button.
- 
4. You will be prompted to authenticate, as shown in the following figure. Provide the necessary information:



Figure 20. Entering your user ID and password

- a. Fill in your z/OS user ID and password.
- b. If you want to eliminate having to reenter your user ID and password each time you access the administration pages, check the check box.
- c. Click **OK**.  
This calls up the “PKI Services Administration” Web page. (See Figure 21 on page 222.)

### Notes:

## Using the administration Web pages

- Your Web server programmer might provide you with an alternate URL for accessing the administration home page. You might also have to authenticate using a certificate instead of a user ID and password.
- Your browser caches the authentication information that you provide. Therefore, if you need to change this information, you first must close all instances of your browser. Then open the browser and, when the panel shown in Figure 20 on page 219 appears, enter the correct information.

---

## Fields in the administration Web pages

When you process certificates requests and certificates, you provide information for various fields in the Web pages. The following table describes the fields in the administration Web pages:

*Table 52. Summary of fields in the administration pages*

Field	Description
Recent activity	This specifies a time range for searches. Possible values include: <ul style="list-style-type: none"><li>• Not selected</li><li>• Within the past day</li><li>• Within the past week</li><li>• Within the past month</li><li>• Within the past six months</li></ul>
Requestor name	The name of the person requesting the certificate, as it appears in the common name field of the certificate request form.
Serial number	PKI Services assigns this number to a certificate when you approve it.
Transaction ID	PKI Services assigns this number to a request when a user requests it. This is a text field of up to 56 characters.

---

## Processing certificate requests

Before you can use the Web page to process certificate requests, you need to understand the statuses of certificate requests and the actions you can perform on these certificate requests.

## Status of certificate requests

Requests for certificates are kept in a request database while they are active. This is from the moment they are created until an event occurs that causes them to be deleted. The following table summarizes possible statuses. During the time period when a certificate request is active, it can have only one of the following statuses at a time:

*Table 53. Statuses of certificate requests*

Status	Meaning
Pending Approval	The request requires administrative approval. No action has been taken on the request yet.
Approved	The administrator explicitly approved the request or it was submitted as an auto-approved certificate request. The actual certificate may or may not have been created at this point.
Completed	The certificate has been issued and the requestor has retrieved it. This is a final state.

Table 53. *Statuses of certificate requests (continued)*

Status	Meaning
Preregistered	The certificate request is from a preregistered Simple Certificate Enrollment Protocol (SCEP) client.
Rejected	The administrator rejected the request, and the requestor has <i>not</i> been informed of this action (because the user has not tried to retrieve the certificate).
Rejected, User notified	The administrator rejected the request and the requestor has been informed of this action when attempting to retrieve the certificate. This is a final state.

A request is deleted from the request database when the administrator explicitly deletes it or when the request expires. This expiration time period is configurable and varies depending on whether the request was finalized or not.

## Actions on certificate requests

The following table summarizes actions on certificate requests and the required status for each of these actions:

Table 54. *Summary of actions to perform on requests and required status*

Action	Required status of request
Approve	Pending Approval
Approve with modifications	Pending Approval
Reject	Pending Approval
Delete	All statuses (Pending Approval, Approved, Completed, Preregistered, Rejected, or Rejected, Notified)

## Using the PKI Services administration home page

The following figure shows the “PKI Services Administration” home page:

**PKI Services Administration**

**Choose one of the following:**

- **Work with a single certificate request**  
Enter the Transaction ID:
- **Work with a single issued certificate**  
Enter the Serial Number:
- **Specify search criteria for certificates and certificate requests**

Certificate Requests	Issued Certificates
<input type="radio"/> Show all requests	<input type="radio"/> Show all issued certificates
<input checked="" type="radio"/> Show requests pending approval	<input type="radio"/> Show revoked certificates
<input type="radio"/> Show approved requests	<input type="radio"/> Show suspended certificates
<input type="radio"/> Show completed requests	<input type="radio"/> Show expired certificates
<input type="radio"/> Show rejected requests	<input type="radio"/> Show active certificates (not expired, not revoked, not suspended)
<input type="radio"/> Show rejections in which the client has been notified	<input type="radio"/> Show disabled certificates (suspended or revoked, not expired)
<input type="radio"/> Show preregistered requests	

**Additional search criteria (Optional)**

Requestor's name

Show recent activity only

email: [webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 21. PKI Services administration home page

This Web page allows you to:

- Process a single certificate request (by specifying its transaction ID)
- Process a single certificate (by specifying its serial number)
- Search for groups of certificate requests or certificates by status and additional search criteria so that you can process them

You can process a single certificate request or a single preregistered (certificate) request if you know its transaction ID. Otherwise, you can perform a search to display all certificate requests of a particular status.

### Steps for processing a single request

To process a single request, perform the following steps:

1. On the PKI Services administration home page (see Figure 21), enter the transaction ID in the field provided for it, and click the **Process request** button. This displays the “Single Request” Web page shown in the example figure (Figure 22 on page 223).

## Single Request

<b>Requestor:</b>	W. M. Fortey	<b>Created:</b>	2006/06/06
<b>Status:</b>	Pending Approval	<b>Modified:</b>	2006/06/06
<b>Transaction Id:</b>	1j5OI+j2b4GN2SHV++++++	<b>Passphrase:</b>	PICK9FAN
<b>Template:</b>	1-Year PKI SSL Browser Certificate	<b>NotifyEmail:</b>	wmfortey@wmfortey.com

---

**Subject:** MAIL=wmfortey@wmfortey.com,CN=W. M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm  
**Issuer:** OU=Master CA,O=IBM,C=US  
**Validity:** 2006/06/06 00:00:00 - 2007/06/05 23:59:59  
**Usage:** handshake(digitalSignature, keyEncipherment)  
**Extended Usage:** clientauth  
**Fingerprints:**  
**SHA1:** 3F:9B:86:4D:06:66:57:15:04:8B:4D:40:0A:5B:55:AA:7B:2F:C5:62  
**MD5:** 5D:BE:48:DF:8D:4E:24:82:D3:DE:97:A3:A6:89:9C:92

### Action to take:

Action Comment (Optional)

email: [webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 22. Single request approval Web page

2. Make sure the request is the correct one by reviewing the information in the top part of the Web page.

**Guideline:** If this is a preregistered certificate request, examine the *fingerprints* (the SHA1 and MD5 hashes) and contact the certificate requestor to confirm that the fingerprints in this received request match the fingerprints in the original request. These actions ensure the integrity of the request. (The requestor can use the SCEP client software to display the fingerprints saved for the original request.)

3. Optionally insert a comment.

4. Click one of the buttons on the “Single Request” Web page to process the request.
  - The **Approve the request as is** button
  - The **Approve the request with modifications** button
  - The **Reject request** button
  - The **Delete request** button

## Using the administration Web pages

The buttons on the “Single Request” Web page appear based on the status of the request. For example:

- If the status of a request is Pending Approval, only the first three buttons in the preceding list appear.
- If the administrator has already processed the request or if the request is Preregistered, only the **Delete request** button appears.

a. When you click the **Approve the request as is** button and processing is successful, this displays a Web page that says you that “Processing is successful”, such as the following. (Otherwise, the Web page says “Processing is not successful”.)



Figure 23. Processing successful Web page

From these Web pages, you can then click the **Process more request(s)** button to return to the PKI Services administration home page (Figure 21 on page 222).

b. When you click the **Approve the request with modifications** button, this displays a “Modify and Approve Request” Web page similar to the one shown in Figure 25 on page 225.

If the subject's distinguished name contained in the current request is not in the proper format for RACF processing, you will see the note (Figure 24) on the “Modify and Approve Request” Web page.

**Note - the existing subject's name is not in a format that can be recreated by PKI Services. Therefore, specifying any subject's name field below will cause the existing name to be deleted and completely replaced.**

Figure 24. Restriction note on the modify and approve request Web page

**Restriction:** If you receive the note shown in Figure 24, you cannot change any field of the subject's distinguished name (the common name, organizational unit, or organization field) without causing PKI Services to delete the entire subject's distinguished name and replace it with your changed values. (This is because the subject's distinguished name is not in the proper format for RACF processing.)

## Modify and Approve Request

Requestor	Request Information	Dates
W. M. Fortey	Trans ID:1j5OI+j2b4GN2SHV++++++ Template:1-Year PKI SSL Browser Certificate Subject:MAIL=wmfortey@wmfortey.com,CN=W. M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm	Created: 2006/06/06 Modified:2006/06/06

You may modify the following fields by providing new values. To remove a field simply blank it out or de-select it.

### • Subject Distinguished Name:

Common Name (optional)  
W. M. Fortey

Email address for distinguished name MAIL= attribute  
wmfortey@wmfortey.com

Organizational Unit (optional)  
Class 1 Internet Certificate CA

Organizational Unit (optional)

Organization (optional)  
The Firm

### • Extensions:

Indicate the key usage for the certificate (optional)  
Protocol handshaking, e.g. SSL (digitalSignature, keyEncipherment)  
Certificate and CRL signing (keyCertSign, cRLSign)  
Document signing (nonRepudiation)  
Data encryption (dataEncipherment)  
Authentication (digitalSignature)  
Key Transport (keyEncipherment)  
Key agreement (keyAgreement)  
Certificate signing (keyCertSign)  
CRL signing (cRLSign)

Indicate the extended key usage the certificate  
Server side authentication (serverAuth)  
Client side authentication (clientAuth)  
Code signing (codeSigning)  
Email protection (emailProtection)  
Digital time stamping (timeStamping)  
OCSP response signing (OCSPSigning)  
Microsoft Smart Card Logon (msSmartCardLogon)

HostIdMappings Extension value(s) in subject-id@host-name form (optional)

HostIdMappings Extension value(s) in subject-id@host-name form (optional)

HostIdMappings Extension value(s) in subject-id@host-name form (optional)

HostIdMappings Extension value(s) in subject-id@host-name form (optional)

### • Validity Period:

Date certificate becomes valid    Date certificate expires (at end of day)  
2006 6 6    2007 6 5

Action Comment (Optional)

email: webmaster@your-company.com

Figure 25. Modifying the request Web page

## Using the administration Web pages

On the “Modify and Approve Request” Web page, you can change the following fields.

- Common name, unless noted as in Figure 24 on page 224
- Organizational unit (this can be multiple fields), unless noted as in Figure 24 on page 224
- Organization, unless noted as in Figure 24 on page 224
- e-mail address

**Note:** If you change the value of the e-mail address field (Email) and if the original request included the notification e-mail address field (NotifyEmail), the value of the latter field is changed to match the changed e-mail address value.

- Street
- Postal code
- Certificate purpose
- Date certificate becomes valid
- Date certificate expires
- HostIdMappings extensions (This can be multiple fields.)
- Optional comment about action you perform on the certificate.

When you are satisfied with the changes you have made, click the **Approve with specified modifications** button; or, if you change your mind, you can click **Reset modified fields**. Alternately, you can click **Home page** to go to the PKI Services home page. (See Figure 6 on page 199.)

- C. When you click the **Reject request** button, this displays a Web page that informs you that “Processing is successful” or that “Processing is not successful”. From these Web pages, click the **Process more request(s)** button to return to the PKI Services administration home page. (See Figure 21 on page 222.)
- d. When you click the **Delete request** button, this displays a Web page that informs you that “Processing is successful” or that “Processing is not successful”. On these Web pages, click the **Process more request(s)** button to return to the PKI Services administration home page. (See Figure 21 on page 222.)

---

### Steps for processing requests by performing searches

The administrator can use the Web page to search for certificate requests of various statuses. The following table summarizes the searches listed on the Web page and the certificate requests that are displayed as a result:



Table 55. Searches to display certificate requests

Search criteria	Results
Show all requests	Displays all certificate requests (all statuses (Pending Approval, Approved, Completed, Preregistered, Rejected, or Rejected, User Notified).
Show requests pending approval	Displays only certificate requests whose status is Pending Approval.
Show approved requests	Displays certificate requests whose status is Approved or Completed.
Show completed requests	Displays certificate requests whose status is Completed.
Show preregistered requests	Display certificate requests whose status is Preregistered.
Show all rejected requests	Displays certificate requests whose status is Rejected, or Rejected, User Notified.
Show rejections in which the client has been notified	Displays certificate requests whose status is Rejected, User Notified.

To process requests by performing a search for requests of a particular status, perform the following steps:

1. On the PKI Services administration home page (see Figure 21 on page 222), select one of the searches by clicking the appropriate radio button under “Certificate Requests”. (The preceding table describes these searches.) You can optionally fill in additional search criteria (“Requestor’s name” and “Show recent activity only”).

**Guideline:** Queries against the request database may time out if the database contains a large number of records. The performance of the query can be vastly improved by supplying “Requestor’s name” as additional search criteria if the saved requestor data is meaningful to your organization and it is recallable. In this case, a PKI exit can be used to supply a meaningful value, such as a Lotus Notes® short name or customer account number.

2. Click **Find certificates or certificate requests** button. This displays the following Web page:

## Certificate Requests

The following certificate requests matched the search criteria specified:

All <input checked="" type="checkbox"/>	Requestor	Certificate Request Information	Status	Dates
<input checked="" type="checkbox"/>	W. M. Fortey	Trans ID: <a href="#">1j5OH+i2b4GN2SHV+++++</a> Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=wmfortey@wmfortey.com,CN=W. M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm	Completed Serial #: 4	Created: 2006/06/06 Modified: 2006/06/06
<input checked="" type="checkbox"/>	Jean Fortey	Trans ID: <a href="#">1j5Op7JhK1Lz2SHV+++++</a> Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=jfortey@wmfortey.com,CN=Jean Fortey,OU=Class 1 Internet Certificate CA,O=The Firm	Pending Approval	Created: 2006/06/06 Modified: 2006/06/06
<input checked="" type="checkbox"/>	scep client 1234	Trans ID: <a href="#">1j5Oqa31dFqz2SHV+++++</a> Template: 5-Year SCEP Certificate - Preregistration Subject: SCEP client 1234	Preregistered	Created: 2006/06/06 Modified: 2006/06/06

Choose one of the following:

- Click on a transaction ID to see more information or to modify, approve, reject, or delete requests individually
- Select and take action against multiple requests at once

Action Comment (Optional)

- Approve without modification all requests selected above that are "Pending Approval"

- Reject all requests selected above that are "Pending Approval"

- Delete all requests selected above

Figure 26. Processing requests after searching

**Note:** The table at the top of the Web page shows the certificate requests that match your search criteria. (If multiple certificates requests match the search criteria, up to ten appear on a Web page, and a button at the bottom of the Web page allows you to view the next set.)

### 3. You can use this Web page:

- To process a single certificate request
- To perform the same action on all of the certificate requests that are listed
- To process selected requests.

To process a single certificate request:

- Click on its transaction ID in the table at the top of the Web page. This transfers you to the single request Web page; see Figure 22 on page 223.
- From the "Single Request" Web page, you can perform the steps in the preceding section, starting with Step 2 on page 223).

To perform the same action on all the certificate requests that are listed:

- Optionally enter a comment.
- Click one of the action buttons below the comment field to perform that action on all listed requests:

<b>Approve</b>	Approves without modification all requests that are pending approval.
<b>Reject</b>	Rejects all requests that are pending approval.
<b>Delete</b>	Deletes all requests.

**Note:** The **Approve** and **Reject** buttons appear only if certificate requests are pending approval. Otherwise, only the **Delete** button appears.

To process selected certificate requests:

- Uncheck the check box beside the **Select** column header. (When the check box beside **Select** is checked, all the individual check boxes in the body of the table are checked. This means all these certificate requests are selected. Unchecking the box in the header unchecks all the boxes in the body of the table.)
- Check the check boxes of all the certificate requests for which you want to perform a particular action.
- Optionally enter a comment.
- Click one of the action buttons below the comment field to perform that action on all listed requests. The action buttons include the following:

<b>Approve</b>	Approves without modification all requests that are pending approval.
<b>Reject</b>	Rejects all requests that are pending approval.
<b>Delete</b>	Deletes all requests.

**Note:** The **Approve** and **Reject** buttons appear only if certificate requests are pending approval. Otherwise, only the **Delete** button appears.

**Tip:** If you select the **Show all requests** radio button (see Figure 21 on page 222) and click the **Approve** button on this Web page, only the certificate requests whose status is Pending Approval are approved.

Instead of processing one or more certificate requests, you can click the **Respecify your search criteria Web page** button to return to the PKI Services administration home page (see Figure 21 on page 222) or the **Home page** button to return to the PKI Services home page (see Figure 6 on page 199).

#### 4. After you click an action button, the next Web page is one of the following:

- “Processing successful” (see Figure 27 on page 230)
- “Processing was not successful” (see Figure 28 on page 230)
- “Processing partially successful” (see Figure 29 on page 231)

If “Processing was not successful”, you can click on the transaction ID to display the “Single Request” Web page; see Figure 22 on page 223. Processing can be unsuccessful because requests do not have the status required for the action you selected; see Table 54 on page 221.

If you get “Processing partially successful”, you can click on the transaction ID to display the “Single Request” Web page; see Figure 22 on page 223. This message can occur when your organization has more than one administrator and involves the following sequence:

- One administrator performs a search.
- Another administrator performs a search before the first administrator has approved requests displayed in the search results.
- One of the administrators approves only some of the requests.

## Using the administration Web pages

- d. The other administrator tries to approve requests including at least one the preceding administrator has already approved and one that the preceding administrator has not already approved.



Figure 27. Request processing was successful Web page



Figure 28. Request processing was not successful Web page



Figure 29. Request processing was partially successful Web page

5. After approving requests as appropriate, you can:

- Click **Process more request(s)** to return to Figure 26 on page 228.
- Click **Administration home page** to return to Figure 21 on page 222.
- Click **Home page** to return to Figure 6 on page 199.

## Processing certificates

Before you can use the Web page to process certificates, you need to understand the statuses of certificates and actions you can perform on certificates.

### Status of certificates

Certificates that have been created from requests are maintained permanently in an issued certificate database. Another name for this is the issued certificate list (ICL). Issued certificates are also published in an LDAP directory.

A certificate can have only one of the following states (statuses) at a time:

Table 56. Status of certificates

Active	The certificate has not yet expired, has not been revoked, and is not currently suspended.
Expired	The certificate's validity period expired while it was active.
Revoked	The certificate has not expired but it has been revoked. Such certificates are published on the next certificate revocation list (CRL).
Revoked, Expired	The certificate was either revoked or suspended, and time has elapsed so that it is now also expired. Such certificates are not published on the next CRL.
Suspended	The certificate has not expired but it is currently suspended. Such certificates are published on the next certificate revocation list (CRL).

## Using the administration Web pages

The administrator must approve a request for the certificate to have a status (as enumerated in the preceding list) or for the administrator to delete the certificate from the ICL. (An administrator can delete a certificate from the ICL, but this would not be a normal situation.) Alternately, the administrator can reject a request or delete the request from the request database (RDB). If the administrator does not approve the request, it is never listed in the ICL.

## Actions for certificates

The following table summarizes actions on certificates and the required status to perform these actions:

*Table 57. Summary of actions to perform and required status to do so*

Action	Required status of certificate	Who performs action
Renew	Active	End user
Resume	Suspended	Administrator
Revoke	Active or Suspended	End user or administrator
Suspend	Active	End user or administrator
Delete	All statuses (Active, Expired, Suspended, Revoked, or Revoked, Expired)	Administrator

**Note:** You may resume (reactivate) a suspended certificate, or permanently revoke it, if the certificate has not yet expired and the suspension grace period has not elapsed. If the grace period has elapsed, the certificate will be permanently revoked the next time certificate revocation lists (CRLs) are issued.

## Steps for processing a single certificate

To process a single certificate, perform the following steps:

1. On the PKI Services administration home page (see Figure 21 on page 222), enter the serial number of the certificate you want to process in the field provided for it. The following is an example of the Web page that is displayed:

## Single Issued Certificate

<b>Requestor:</b>	W. M. Fortey	<b>Created:</b>	2006/06/06
<b>Status:</b>	Active	<b>Modified:</b>	2006/06/06
<b>Template:</b>	1-Year PKI SSL Browser Certificate	<b>PassPhrase:</b>	PICK9FAN
<b>Serial #:</b>	4		
<b>Previous Action Comment:</b>	Issued certificate		

<b>Subject:</b>	MAIL=wmfortey@wmfortey.com,CN=W. M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm
<b>Issuer:</b>	OU=Master CA,O=IBM,C=US
<b>Validity:</b>	2006/06/06 00:00:00 - 2007/06/05 23:59:59
<b>Usage:</b>	handshake(digitalSignature, keyEncipherment)
<b>Extended Usage:</b>	clientauth

### Action to take:

Action Comment (Optional)

Revoke Certificate No Reason

[email: webmaster@your-company.com](mailto:webmaster@your-company.com)

Figure 30. Processing a certificate from the single certificate Web page

- Make sure the certificate is the correct one by reviewing the information in the top part of the Web page.
- If you are going to process a certificate from this Web page, you can optionally insert a comment.
- Click one of the following buttons to process the certificate:
 

<b>Revoke certificate</b>	Revokes the certificate.
<b>Suspend certificate</b>	Suspends the certificate.
<b>Resume certificate</b>	Resumes the certificate.

## Using the administration Web pages

### Delete certificate

Deletes the certificate. (This is for cleanup purposes.)

#### Notes:

- a. The **Suspend** and **Revoke** buttons appear only if the status of the certificate is **Active**.
- b. The **Resume** button appears only if the status of the certificate is **Suspended**.

## Steps for processing certificates by performing searches

The administrator can use the Web page to search for certificates of various statuses. The following table summarizes the searches listed on the Web page and the certificates that are displayed as a result:

Table 58. Searches to display certificates

Searches	Results
Show all issued certificates	Displays all certificates (can be any status—Active, Expired, Suspended, Revoked, or Revoked, Expired).
Show revoked certificates	Displays certificates whose status is Revoked or Revoked, Expired.
Show suspended certificates	Displays certificates whose status is Suspended.
Show expired certificates	Displays certificates whose status is Expired or Revoked, Expired.
Show active certificates (not expired, not revoked, not suspended)	Displays certificates whose status is Active.
Show disabled certificates (suspended or revoked, not expired)	Displays certificate requests whose status is Suspended or Revoked.

To process certificates by performing a search for certificates of a particular status, perform the following steps:

1. On the PKI Services administration home page (see Figure 21 on page 222), select one of the searches by clicking the appropriate radio button under “Issued Certificates”. (The preceding table describes these searches.) You can optionally fill in additional search criteria (“Requestor’s name” and “Show recent activity only”).
2. Click the **Find certificates or certificate requests** button. This displays the following Web page.



## Issued Certificates

The following issued certificates matched the search criteria specified:

All <input checked="" type="checkbox"/>	Requestor	Certificate Information	Status	Dates
<input checked="" type="checkbox"/>	Jim Renew	Serial #: <a href="#">19092</a> Template: 1-Year PKI SSL Browser Certificate Subject: CN=Jim Renew,OU=Class 1 Internet Certificate CA,O=The Firm	Active	Created: 2003/07/17 Modified: 2003/07/22
<input checked="" type="checkbox"/>	W.M. Fortey	Serial #: <a href="#">19093</a> Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=forteywm@somecompany.com,CN=W.M. Fortey,OU=Class 1 Internet Certificate CA,O=The Firm	Active	Created: 2003/07/21 Modified: 2003/07/22
<input checked="" type="checkbox"/>	W. M. Fortey Jr.	Serial #: <a href="#">19094</a> Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=forteywmjr@somecompany.com,CN=W. M. Fortey Jr.,OU=Class 1 Internet Certificate CA,O=The Firm	Active	Created: 2003/07/21 Modified: 2003/07/22
<input checked="" type="checkbox"/>	Doollee Dorbie	Serial #: <a href="#">19095</a> Template: 1-Year PKI SSL Browser Certificate Subject: MAIL=dorbie@somecompany.com,CN=Doollee Dorbie,OU=Class 1 Internet Certificate CA,O=The Firm	Active	Created: 2003/07/21 Modified: 2003/07/22

Choose one of the following:

- Click on a serial number to see more information or to revoke or delete certificates individually
- Select and take action against multiple requests at once

Action Comment (Optional)

Revoke

No Reason



- Revoke all selected active or suspended certificates

Suspend

- Suspend all selected active certificates

Delete

- Delete all selected certificates

Respecify Your Search Criteria

Home Page

[email\\_webmaster@your-company.com](mailto:email_webmaster@your-company.com)

Figure 31. Processing certificates using searches

**Note:** The table at the top of the Web page shows the certificates that match your search criteria. (If multiple certificates match the search criteria, up to ten appear on a Web page, and a button at the bottom of the Web page allows you to view the next set.)

3. You can use this Web page:

- To process a single certificate
- To perform the same action on all of the certificates that are listed
- To process selected certificates

To process a single certificates:

## Using the administration Web pages

- a. Click on its serial number in the table at the top of the Web page. This transfers you to the single certificate Web page; see Figure 30 on page 233.

- b. From the single certificate Web page, you can perform the steps in the preceding section, starting with Step 2 on page 233).

To perform the same action on all the certificates that are listed:

- a. Optionally enter a comment.
- b. Click one of the action buttons below the comment field to perform that action on all listed certificates:

<b>Revoke</b>	Revokes all selected active certificates.
<b>Suspend</b>	Suspends all selected active certificates.
<b>Resume</b>	Resumes all selected suspended certificates.
<b>Delete</b>	Deletes all selected certificates.

### Notes:

- 1) The **Suspend** and **Revoke** buttons appear only when your search matches at least one certificate whose status is Active.
- 2) The **Resume** button appears only when your search matches at least one certificate whose status is Suspended.

To process selected certificates:

- a. Uncheck the check box beside the **Select** column header. (When the check box beside **Select** is checked, all the individual check boxes in the body of the table are checked. This means all these certificates are selected. Unchecking the box in the header unchecks all the boxes in the body of the table.)
- b. Check the check boxes of all the certificates for which you want to perform a particular action.
- c. Optionally enter a comment.
- d. Click one of the action buttons below the comment field to perform that action on all listed requests. The action buttons include:

<b>Revoke</b>	Revokes all selected active certificates.
<b>Suspend</b>	Suspends all selected active certificates.
<b>Resume</b>	Resumes all selected suspended certificates.
<b>Delete</b>	Deletes all selected certificates.

### Notes:

- 1) The **Suspend** and **Revoke** buttons appear only when your search matches at least one certificate whose status is Active.
- 2) The **Resume** button appears only when your search matches at least one certificate whose status is Suspended.

Instead of processing one or more certificates, you can click the **Respecify your search criteria Web page** button to return to the PKI Services administration home page. (See Figure 21 on page 222) or the **Home page** button to return to the PKI Services home page (see Figure 6 on page 199.)

- 
4. After you click an action button, the next Web page tells you:

- “Processing was successful” (see Figure 32 on page 237)
- “Processing was not successful” (see Figure 33 on page 237)
- “Processing partially successful” (see Figure 34 on page 238)

If “Processing was not successful”, you can click on a serial number to display the “Single Certificate” Web page; see Figure 30 on page 233. Processing can

be unsuccessful because certificates do not have the status required for the action you selected; see Table 57 on page 232.

If you get “Processing partially successful”, you can click on the serial number to display the “Single Certificate” Web page; see Figure 30 on page 233. The “Processing partially successful” message can occur when your organization has more than one administrator and involves the following sequence:

- a. One administrator performs a search.
- b. Another administrator performs a search before the first administrator has revoked or deleted certificates displayed in the search results.
- c. One of the administrators revokes or deletes some of the certificates.
- d. The other administrator tries to revoke or delete certificates including at least one the preceding administrator has already revoked or deleted and at least one the preceding administrator has not already revoked or deleted.

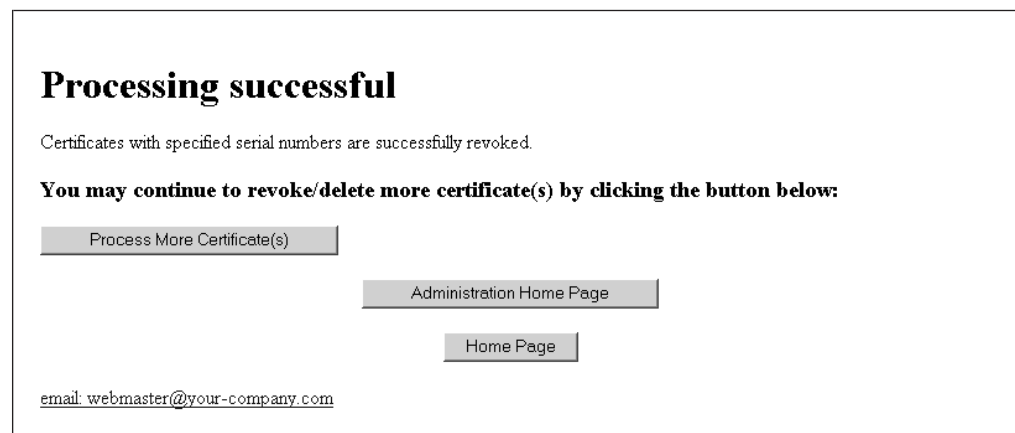


Figure 32. Processing of certificate was successful Web page

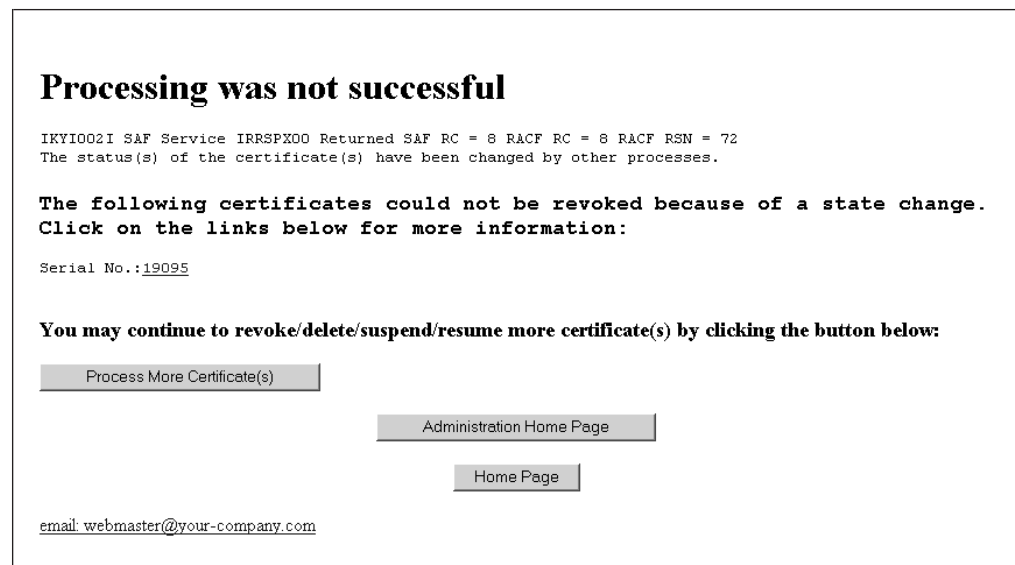


Figure 33. Request processing was not successful Web page



Figure 34. Request processing was partially successful Web page

You can click the **Home page** button to return you to the PKI Services home page. (See Figure 6 on page 199.)

## Relationship between certificate requests and matching certificates

PKI Services maintains two databases:

- The request database (RDB) also called the ObjectStore
- The issued certificate list (ICL)

RDB records are temporary in nature. They exist only to track active requests. PKI Services automatically removes these records when they are complete or go inactive. ICL records are permanent. Requests for certificates (both new and renewal) are stored in the RDB. Once approved, a matching certificate is created from the request and stored in the ICL. (Note, the creation of the certificate may not be instantaneous.) At this point, the two database records, though related, exist independently of each other.

- After a request is approved, there is no way for you to *un*-approve a request. If you mistakenly approve a request that you meant to reject, you should immediately delete the RDB entry. This prevents the user from retrieving the certificate. You should then search the issued certificates to see if the certificate has been issued. If it has, you should revoke it in case the user has already picked it up.
- Revoking a certificate (an ICL action) has no effect on its matching RDB entry. If you revoke a certificate, you should also delete its matching RDB entry if it exists. This prevents the user from retrieving the certificate, if the user has not already done so.
- You can delete RDB entries any time after they have been completed to save space in the database if desired.
- Under normal circumstances, ICL entries should not be deleted. If you delete an ICL entry, you will no longer be able to revoke or renew the certificate.
- You can delete entries in any state in either database to clean up error conditions.

---

## Part 5. Administering security for PKI Services

This part explains how to administer security for PKI Services.

- Chapter 16, “RACF administration for PKI Services,” on page 241 describes how to use RACF to administer security for PKI Services.

The following tasks are covered:

- “Authorizing users for the PKI Services administration group” on page 241
- “Authorizing users for inquiry access” on page 241
- “Administering HostIdMappings extensions” on page 242
- “Locating your PKI Services certificates and key ring” on page 244
- “Establishing PKI Services as an intermediate CA” on page 246
- “Renewing your PKI Services CA and RA certificates” on page 248
- “Recovering a CA certificate profile” on page 251
- “Retiring and replacing the PKI Services CA private key” on page 254
- “R\_PKIServ (IRRSPX00) callable service” on page 257
- “Using encrypted passwords for LDAP servers” on page 260.



---

## Chapter 16. RACF administration for PKI Services

This chapter describes the tasks that the RACF administrator performs after PKI Services has been set up and customized.

The following tasks are covered:

- “Authorizing users for the PKI Services administration group”
- “Authorizing users for inquiry access”
- “Administering HostIdMappings extensions” on page 242
- “Locating your PKI Services certificates and key ring” on page 244
- “Establishing PKI Services as an intermediate CA” on page 246
- “Renewing your PKI Services CA and RA certificates” on page 248
- “Recovering a CA certificate profile” on page 251
- “Retiring and replacing the PKI Services CA private key” on page 254
- “R\_PKIServ (IRRSPX00) callable service” on page 257
- “Using encrypted passwords for LDAP servers” on page 260.

For more information about the RACF commands shown in this chapter, see *z/OS Security Server RACF Command Language Reference*.

---

### Authorizing users for the PKI Services administration group

You need to know how to add and delete members from the PKI Services administration group (by default, PKIGRP).

#### Connecting members to the group

The PKI Services administration group is a RACF group containing the list of user IDs that are authorized to use PKI Services administration functions. To connect a member to the group, execute the following command, replacing *pkigroup\_mem* with the member's user ID and *pkigroup* with the name of the PKI Services administration group (PKIGRP by default). (See Table 17 on page 36 for more information.)

```
CONNECT pkigroup_mem GROUP(pkigroup)
```

**Note:** You need to enter this command for each user ID in turn.

#### Deleting members from groups

To remove a user from a group, execute the following command, replacing *pkigroup\_mem* with the user ID of the member you want to delete and *pkigroup* with the name of the PKI Services administration group (PKIGRP by default).

```
REMOVE pkigroup_mem GROUP(pkigroup)
```

---

### Authorizing users for inquiry access

You can add groups of users who do not need the full administrative authority of users in the PKIGRP group. You can use the following procedure to authorize a new group for inquiry abilities, such as a help desk might require. The commands shown include variables whose names are appropriate for this scenario.

### Steps for authorizing users for inquiry access

**Before you begin:** You need to know the high-level VSAM data set qualifier used for the IKYSETUP variable *vsamhlq* value, in case your installation did not use the PKISRV default. (See Table 17 on page 36.)

Perform the following steps to add and administer a group that needs authority to query PKI Services information.

1. Add the new group.

**Example:**

```
ADDGROUP HELPDESK OMVS(GID(197312))
```

---

2. Connect each member to the new group. Repeat for each user ID you need to connect.

**Example:**

```
CONNECT OPER17 GROUP(HELPDESK)
```

---

3. Authorize the new group for READ access to the resources of PKI Services. Replace your installation's value for the data set's high-level qualifier if your installation did not use the PKISRV default.

**Example:**

```
PERMIT 'PKISRV.**' ID(HELPDESK) ACCESS(READ)
PERMIT IRR.RPKISERV.PKIADMIN CLASS(FACILITY)
      ID(HELPDESK) ACCESS(READ)
SETROPTS GENERIC(DATASET) REFRESH
SETROPTS RACLIST(FACILITY) REFRESH
```

The SETROPTS commands activate the profiles that authorize READ access.

---

4. If necessary, you can remove a user from the group. The following example removes the user you connected in Step 2.

**Example:**

```
REMOVE OPER17 GROUP(HELPDESK)
```

---

5. If necessary, you can delete the group. The following example deletes the group you created in Step 1.

**Example:**

```
DELGROUP(HELPDESK)
```

---

### Administering HostIdMappings extensions

You can add a HostIdMappings extension to certificates you create for certain users, allowing you to specify the user IDs that each user will be able to use for login to particular servers (or hosts). Controlling an identity used for login purposes is a very important security objective. Therefore, you must exercise administrative control in the following areas by authorizing:

- PKI Services as a highly trusted certificate authority whose certificates will be honored when they contain HostIdMappings extensions



- Particular servers to accept logins from clients whose certificates contain HostIdMappings extensions

## Steps for administering HostIdMappings extensions

Perform the following steps to allow the Web server to accept logins from clients who have been issued PKI Services certificates with HostIdMappings extensions:

1. Determine if PKI Services is defined as a highly trusted certificate authority on your system by listing its certificate authority definition by using the RACDCERT CERTAUTH LIST command.

### Example:

```
RACDCERT CERTAUTH LIST(LABEL('Local PKI CA'))
```

Check the Status information near the top of the output listing for the HIGHTRUST attribute.

2. If not already defined, add the HIGHTRUST attribute to the certificate authority definition for PKI Services.

### Example:

```
RACDCERT CERTAUTH ALTER(LABEL('Local PKI CA')) HIGHTRUST
```

3. Define a resource in the SERVAUTH class for each server (host) name you want your Web server to honor when accepting logins for certificates containing HostIdMappings extensions. The resource name follows the format: IRR.HOST.*hostname*. The *hostname* is the value of the HostIdMappings extension entry pertaining to the z/OS host system you are administering (without the subject ID portion). This is usually a domain name, such as plpsc.pok.ibm.com. The following example shows defining a resource.

### Example:

```
RDEFINE SERVAUTH IRR.HOST.PLPSC.POK.IBM.COM UACC(NONE)
```

4. Permit your Web server to access this resource with READ authority. Be sure the Web server is defined as a RACF user.

### Example:

```
PERMIT IRR.HOST.PLPSC.POK.IBM.COM CLASS(SERVAUTH) ID(WEBSRV) ACCESS(READ)
```

5. Activate the SERVAUTH class, if not already active.

### Example:

```
SETROPTS CLASSACT(SERVAUTH)
```

If already active, refresh the SERVAUTH class.

### Example:

```
SETROPTS CLASSACT(SERVAUTH) REFRESH
```

**Note:** On a z/OS system, a HostIdMappings extension is not honored if the target user ID was created after the start of the validity period for the certificate containing the HostIdMappings extension. Therefore, if you are creating user IDs specifically for certificates with HostIdMappings extensions, make sure that you create the user IDs before the certificate requests are submitted.

Alternately, when approving the certificate, you can modify the date the certificate becomes valid so that it is not earlier than the date the user ID was created. For renewed certificates, all the original information is replicated in the new certificate, including the date the certificate becomes valid and any HostIdMappings. If you want to change a HostIdMappings extension when approving the renewed certificate, you must also modify the date the certificate becomes valid so that it is not earlier than the date the user ID was created.

See *z/OS Security Server RACF Command Language Reference* for details about syntax and authorization required for using the RACDCERT command.

### Locating your PKI Services certificates and key ring

The IKYSETUP exec sets up the RACF environment for PKI Services. After the setup is complete, you may need to go back and locate the PKI Services CA certificate, key ring, or the optional RA certificate, possibly to diagnose error conditions. You can do this by using various RACF TSO commands.

**Before you begin:** You need to determine the following setup information:

Table 59. Information you need for locating your PKI Services certificates and key ring

Information needed	Where to find this information	Record your value here
<i>ca_label</i> —The label of your CA certificate in RACF	See Table 11 on page 29.	
<i>ra_label</i> —The label of your RA certificate in RACF	See Table 11 on page 29.	
<i>ca_ring</i> —The PKI Services SAF key ring	See Table 17 on page 36.	
<i>daemon</i> —The user ID for the PKI Services daemon	See Table 17 on page 36.	
<i>log_dsn</i> —The data set name of the IKYSETUP log	See Table 17 on page 36.	
<i>export_dsn</i> —The data set name of your CA certificate as exported from RACF	See Table 17 on page 36.	

### Steps for locating the PKI Services certificates and key ring

Perform the following steps to locate the PKI Services CA certificate, key ring, and the optional RA certificate:

1. Locate the CA certificate using one of the following two methods (Step 1a or Step 1b) and examine its information.
  - a. Locate the CA certificate using the name of its export data set. (Get the export data set name from *export\_dsn* in Table 59.) Display its information by executing the following RACF command from a TSO command prompt:  
RACDCERT CHECKCERT(*export\_dsn*)

**Sample output:**

```
Digital certificate information for CERTAUTH:
Label: Local PKI CA
Certificate ID: 2QiJmZmDhZmjgdOWg4GTQnFSyUDDwUBA
Status: HIGHTRUST
Start Date: 2001/06/04 23:00:00
End Date: 2020/01/01 22:59:59
```

```

Serial Number:
>00<
Issuer's Name:
>OU=Human Resources Certificate Authority.O=IBM.C=US<
Subject's Name:
>OU=Human Resources Certificate Authority.O=IBM.C=US<
Key Usage: CERTSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024

```

- b. Alternately, locate the CA certificate using its certificate label. (Get the label name from *ca\_label* in Table 59 on page 244.) Display its information by entering the following RACF command from a TSO command prompt.

```
RACDCERT CERTAUTH LIST(LABEL('ca_label'))
```

The RACDCERT CERTAUTH command produces the same output as the RACDCERT CHECKCERT (shown in Step 1a on page 244) with the addition of information about any ring associations. For example:

## Sample output:

```

Ring Associations:
Ring Owner: PKISRVD
Ring:
>CAring<

```

- c. Examine the CA certificate information. If you are diagnosing errors, note the following:
  - The first line must indicate that this is a CERTAUTH certificate.
  - Label must match your *ca\_label* value (as in the preceding table).
  - If Serial Number is not equal to 00, this indicates that the certificate has been renewed or was issued by another certificate authority.
  - If Issuer's Name differs from Subject's Name, this indicates that the certificate was issued by another certificate authority.
  - Subject's Name must match the original value recorded for the PKI Services SUBJECTSDN in the IKYSETUP log.
  - Private Key Type and Private Key Size must be present.
  - Private Key Type indicates whether the key is a non-ICSF key (software key) or an ICSF key (ICSF or PCICC key).
  - If Ring Associations are listed, ensure that an association is displayed for the daemon user ID as ring owner and your *ca\_ring* value (from Table 59 on page 244) as ring name.

## 2. Locate the CA key ring and examine its information.

- a. Get the ring name from *ca\_ring* in Table 59 on page 244 and display its information by executing the following RACF command from a TSO command prompt:

```
RACDCERT ID(daemon) LISTRING(ca_ring)
```

## Sample output:

Digital ring information for user PKISRVD:

```

Ring:
>CAring<

```

Certificate Label Name	Cert Owner	USAGE	DEFAULT
Local PKI CA	CERTAUTH	PERSONAL	YES
Local PKI RA	PKISRVD	PERSONAL	NO

- b. Examine the key ring information. If you are diagnosing errors, note the following:
  - The entry for the PKI Services CA certificate must have USAGE PERSONAL and DEFAULT YES.
  - If you use an optional RA certificate, you will see the second line. If present, the entry for the PKI Services RA certificate must have USAGE PERSONAL and DEFAULT NO.

3. If you use an optional RA certificate, locate it and examine its information.
  - a. Locate the RA certificate using its certificate label. (Get the RA's certificate label from *ra\_label* in Table 59 on page 244 or from the RACDCERT LISTRING output shown in Step 2a.) Display the RA certificate information by executing the following RACF command from a TSO command prompt:  
RACDCERT ID(*certificate-owner*) LIST(LABEL('certificate-label-name'))

### Sample output:

Digital certificate information for PKISRVD:

```
Label: Local PKI RA
Certificate ID: 2QiJmZmDhZmjgdOWg4GTQNfSyUDDwUBA
Status: TRUST
Start Date: 2001/06/04 23:00:00
End Date: 2020/01/01 22:59:59
Serial Number:
>01<
Issuer's Name:
>OU=Human Resources Certificate Authority.0=IBM.C=US<
Subject's Name:
>CN=Registration Authority.0U=Human Resources Certif<
>icate Authority.0=IBM.C=US<
Key Usage: HANDSHAKE
Private Key Type: Non-ICSF
Private Key Size: 1024
```

- b. Examine the RA certificate information. If you are diagnosing errors, note the following:
  - The user ID of the certificate owner (indicated in the first line) must match the user ID of the PKI Services daemon.
  - Issuer's Name must match the Subject's Name of the CA certificate.
  - Private Key Type and Private Key Size must be present.
  - Private Key Type indicates whether the key is a non-ICSF key (software key) or an ICSF key (ICSF or PCICC key).

---

## Establishing PKI Services as an intermediate CA

The default setup for PKI Services establishes the PKI Services certificate authority as a root CA, also known as a self-signed CA. Because there is no established trust hierarchy leading to a self-signed certificate, it is impossible to verify that a self-signed certificate is genuine. Accordingly, any person or application that wishes to process certificates issued by a root authority must explicitly trust the authenticity of the self-signed CA certificate.

Alternately, you can establish the PKI Services certificate authority as an intermediate (subordinate) certificate authority. An intermediate certificate authority is one whose certificate is signed by another higher certificate authority. This higher

certificate authority may be a root CA or another intermediate CA. If the root CA certificate has previously been trusted, you can verify any lower intermediate CA certificate using the higher certificate.

In the following steps, you will be replacing the self-signed CA certificate created by IKYSETUP with one signed by another authority

## Steps for establishing PKI Services as an intermediate CA

### Before you begin:

1. This procedure assumes that the PKI Services CA certificate is issued by a root, or self-signed, CA.
2. The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 60. Information you need for establishing PKI Services as an intermediate CA

Information needed	Where to find this information	Record your value here
<i>ca_label</i> —The label of your CA certificate in RACF	See Table 11 on page 29.	
<i>cacert_dsn</i> —The name of the data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	
<i>export_dsn</i> —The data set name of your CA certificate as exported from RACF.	See Table 17 on page 36.	

Perform the following steps to establish PKI Services as an intermediate certificate authority:

1. If you have not yet configured your system for PKI Services, then perform all required steps in Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27.
2. Determine what certificate authority will be acting as a higher authority for PKI Services. (This could be a public certificate authority, such as VeriSign, or a local, internal certificate authority, perhaps even another instance of PKI Services.)
3. Create a new certificate request from your existing self-signed CA certificate by entering the following RACF command from a TSO command prompt:  

```
RACDCERT CERTAUTH GENREQ(LABEL('ca_label')) DSN(cacert_dsn)
```
4. Send the certificate request to the higher certificate authority, following the procedures that higher authority requires.
5. If the root CA is not one that is already know by RACF, then add the root CA to RACF as a certificate authority. To do this, receive the root CA certificate and place it into the certificate data set (*cacert\_dsn*).

**Note:** The procedure for doing this can vary greatly depending on how the higher certificate authority delivered the certificate:

- If the certificate is delivered as base64 encoded text, the easiest way to deposit the certificate into the data set is to edit the certificate data set:
  - a. Delete all existing lines in *cacert\_dsn*.
  - b. Copy the base64 encoded text.
  - c. Paste this into the ISPF edit window.
  - d. Save.
- If the certificate is delivered as binary data (also called DER encoded), the easiest way to deposit the certificate into the data set is to use binary FTP.

- 
6. Add the new PKI Services CA certificate into the RACF data base by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(cacert_dsn) WITHLABEL('label-for-root-CA')
```

---

7. Repeat Step 5 to add the PKI Services CA to RACF as a certificate authority by receiving the PKI Services CA certificate and placing it into the certificate data set (*cacert\_dsn*).
- 

8. Receive the PKI Services CA certificate back into the RACF data base by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(cacert_dsn)
```

---

9. Export the root CA certificate in DER format to the export data set by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH EXPORT(LABEL('label-for-root-CA')) DSN(export_dsn)  
FORMAT(CERTDER)
```

---

10. Make your new root CA certificate available to your clients. To do this, set up the var directory by performing Step 2 through Step 4 in “Steps for setting up the var directory” on page 63.

**Note:** Make sure that the root CA certificate, not your intermediate CA certificate, is stored in */var/pkiserv/cacert.der*.

---

---

## Renewing your PKI Services CA and RA certificates

Eventually, your PKI Services CA and RA certificates will expire. To avoid complications related to an expired CA or RA certificate, renew those certificate before they actually expires. (You will receive MVS console message IKYP026E as the expiration date approaches.)

This topic contains these procedures:

- “Steps for renewing your PKI Services CA certificate” on page 249
- “Steps for renewing your PKI Services RA certificate” on page 250

## Steps for renewing your PKI Services CA certificate

**Before you begin:** The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 61. Information you need for renewing your PKI Services certificate authority certificate

Information needed	Where to find this information	Record your value here
<i>ca_label</i> —The label of your CA certificate in RACF	See Table 11 on page 29.	
<i>cacert_dsn</i> —The data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	
<i>export_dsn</i> —The data set name of your CA certificate as exported from RACF.	See Table 17 on page 36.	

Perform the following steps to renew your PKI Services CA certificate:

1. Create a new certificate request from your self-signed CA certificate by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH GENREQ(LABEL('ca_label')) DSN(cacert_dsn)
```

Make note of the CA certificate's new expiration date and record it here:

<b>CA certificate's new expiration date:</b>	
--	--

2. If your PKI Services certificate authority is a root CA (that is, it has a self-signed certificate, which is the default), then generate the self-signed renewal certificate by entering the following RACF command from a TSO command prompt. (Use the expiration date you recorded in Step 1 as the value for the *ca\_expires* variable.)

```
RACDCERT CERTAUTH GENCERT(cacert_dsn) NOTAFTER(DATE(ca_expires))
SIGNWITH(CERTAUTH LABEL('ca_label'))
```

3. Alternately, if your PKI Services certificate authority is an intermediate certificate authority, perform the following steps:

- a. Send the certificate request to the higher (external) CA, following the procedures that higher authority requires. If your CA retains original certificate signing requests (CSR), you may not need to create and store a new request based on the expiring certificate. You may be able to request a renewal using the original CSR.
- b. After the certificate has been issued, receive the certificate back into the certificate data set (*cacert\_dsn*).

**Note:** The procedure for doing this can vary greatly depending on how the higher certificate authority delivers the new certificate:

- If the certificate is delivered as base64 encoded text, the easiest way to deposit the certificate into the data set is to edit the certificate data set:
  - 1) Delete all existing lines in *cacert\_dsn*.
  - 2) Copy the base64 encoded text.



## RACF administration for PKI Services

- 3) Paste this into the ISPF edit window.
- 4) Save.
- If the certificate is delivered as binary data (also called DER encoded), the easiest way to deposit the certificate into the data set is to use binary FTP.

- C. Receive the certificate back into the RACF data base by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH ADD(cacert_dsn)
```

- 
4. Export the certificate in DER format to the export data set by entering the following RACF command from a TSO command prompt:

```
RACDCERT CERTAUTH EXPORT(LABEL('ca_label')) DSN(export_dsn) FORMAT(CERTDER)
```

- 
5. To make your new certificate available to your clients, set up the /var/pkiserv directory by performing Step 2 through Step 4 in “Steps for setting up the var directory” on page 63.

- 
6. Stop and restart PKI Services.
- 

## Steps for renewing your PKI Services RA certificate

**Before you begin:** The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 62. Information you need for renewing your PKI Services RA certificate

Information needed	Where to find this information	Record your value here
<i>ca_label</i> —The label of your CA certificate in RACF	See Table 11 on page 29.	
<i>daemon</i> —The user ID of the PKI Services daemon.	See Table 17 on page 36.	
<i>racert_dsn</i> —The name of the data set to contain your new certificate request.	You decide this based on local data set naming conventions.	
<i>ra_label</i> —The label of your RA certificate in RACF.	See Table 11 on page 29.	

Perform the following steps to renew your PKI Services RA certificate:

1. Create a new certificate request from your existing RA certificate by entering the following RACF command from a TSO command prompt:

```
RACDCERT ID(daemon) GENREQ(LABEL('ra_label')) DSN(racert_dsn)
```

Make note of the RA certificate's new expiration date and record it here:

<b>RA certificate's new expiration date:</b>	
--	--



2. Create the renewed PKI Services certified RA certificate by entering the following RACF command from a TSO command prompt. (Use the expiration date you recorded in Step 1 on page 250 as the value for the *ra\_expires* variable.)

```
RACDCERT ID(daemon) GENCERT(racert_dsn) NOTAFTER(DATE(ra_expires))
SIGNWITH(CERTAUTH LABEL('ca_label'))
```

3. Stop and restart PKI Services.

## Recovering a CA certificate profile

Unless you change the IKYSETUP REXX exec to disable the function, IKYSETUP automatically backs up the PKI Services CA certificate and private key to a passphrase-encrypted data set that has PKCS #12 format. If the CA certificate profile in RACF is accidentally deleted, you can recover it from the backup data set.

## Steps for recovering a CA certificate profile

**Before you begin:** The commands in the steps that follow include several variables. The following table describes these variables. Determine the values for these variables and record the information in the blank boxes:

Table 63. Information you need for recovering a CA certificate profile

Information needed	Where to find this information	Record your value here
<i>backup_dsn</i> —The name of the data set containing the backup copy of your private key.	See Table 17 on page 36.	
<i>ca_label</i> —The label of your CA certificate in RACF	See Table 11 on page 29.	
<i>ca_ring</i> —The PKI Services SAF key ring.	See Table 17 on page 36.	
<i>cacert_dsn</i> —The name of the data set to contain your new certificate request and returned certificate.	You decide this based on local data set naming conventions.	
<i>daemon</i> —The user ID for the PKI daemon.	See Table 17 on page 36.	
<i>export_dsn</i> —The data set name of your CA certificate as exported from RACF.	See Table 17 on page 36.	
<i>your-passphrase</i> —The passphrase you used when backing up the private key.	You specified this when running IKYSETUP.	

Perform the following steps to recover a CA certificate profile:

1. Execute the following TSO commands:

```
RACDCERT CERTAUTH ADD(backup_dsn) PASSWORD(your-passphrase)
WITHLABEL('ca_label') ICSF
RACDCERT CERTAUTH ADD(export_dsn)
RACDCERT ID(daemon) CONNECT(CERTAUTH LABEL('ca_label'))
RING(ca_ring) USAGE(PERSONAL) DEFAULT)
```

**Note:** If you are not using ICSF, omit the ICSF keyword on the ADD.

- 
2. Perform the following steps to update the RACF profile with the serial number of the last certificate PKI Services issued. (You need to restore the certificate serial number incrementer value that is stored in the profile because otherwise, PKI Services resumes issuing certificates starting from serial number 1.)

- a. Make sure PKI Services is stopped. (See “Stopping the PKI Services daemon” on page 86 for details on how to do this.)
- b. Execute the following command from the UNIX command line to run the iclview utility:

```
iclview \'pkisrzd.vsam.icl\'
```

Record the serial number displayed (in hex) of the last certificate listed:

<b>Serial number (in hex) of last certificate:</b>	
--	--

- c. To determine your CA certificate’s profile name, execute the following command to perform an *unsuccessful* ADD:

```
RACDCERT CERTAUTH ADD(export_dsn) WITHLABEL('*** Bad Label ***')
```

The unsuccessful ADD displays an error message including the profile name. Record the profile name:

<b>Profile name:</b>	
----------------------	--

- d. Create the ICHEINTY ALTER job shown in Figure 35 on page 253 in your own JCL data set, replacing the highlighted values based on the information you recorded in the previous steps.
- e. Submit the job and check its return code.
-

```

//SAMPIC JOB 'xxxxxxx',NOTIFY=xxxxxx,
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=4M
//*****
//ASM EXEC ASMHCL,PARM.C='OBJECT,DECK,TEST',
// PARM.L='MAP,LET,LIST,NCAL,AC(1)'
//C.SYSIN DD *
SAMPICHE CSECT
SAMPICHE AMODE 31
SAMPICHE RMODE ANY
        STM R14,R12,12(R13) Save registers
        BALR R12,0
        USING *,R12 R12 = base register
        B DOIT
*****
* Update the following declares with your certificate info *
*****
ENTRY EQU * Your CA certificate profile
ENTBLEN DC H'54' Length of cert profile name
ENTALEN DC H'54' Length of cert profile name
        DC CL43'00.00=HumanResourcesCertificateAuthority'
        DC CL11'.0=IBM.C=US'
LSER EQU * CERTLSER is an 8 byte field
LSERHIGH DC X'00000000' High word - set to zero
LSERLOW DC X'000000FF' Set to your last serial # (hex)
*****
* Establish standard linkage *
*****
DOIT ST R13,SAVE+4 Save caller's save area address
        LA R15,SAVE Get the next save area address
        ST R15,8(R13) Link the save areas
        LR R13,R15 R13 points to next save area
        ICHEINTY ALTER,TYPE='GEN',ENTRYX=ENTRY,RELEASE=1.9, X
        SEGMENT='CERTDATA',CLASS=DIGTCLAS, X
        ACTIONS=(A_LSER)
CLOSEUP EQU *
        L R13,SAVE+4 Get caller's save area address
        ST R15,16(R13) Save ICHEINTY RC
        LM R14,R12,12(R13) Restore registers except R13
        BR R14 Back to invoker
*****
* CONSTANTS, SAVE AREAS, ETC *
*****
SAVE DS 18F
DIGTCLAS DC CL8'DIGTCERT'
        ORG
A_LSER ICHEACTN FIELD=CERTLSER,FLDATA=(8,LSER),RELEASE=1.9,MF=L
*****
* General Equates *
*****
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
        END SAMPICHE
//C.SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
// DD DSN=SYS1.MODGEN,DISP=SHR
//C.SYSPRINT DD SYSOUT=*
//L.SYSLMOD DD DSN=SYS1.LINKLIB(SAMPICHE),DISP=SHR
//L.SYSPRINT DD SYSOUT=*
//L.SYSIN DD *
        NAME SAMPICHE(R)
/*
//RUNIT EXEC PGM=SAMPICHE
//*

```

Figure 35. Sample JCL data set for restoring the certificate serial number incrementer value

## Retiring and replacing the PKI Services CA private key

For certificates that are associated with private keys, such as the PKI Services CA certificate, you should periodically retire the private keys and replace them with new ones. This process is commonly called *certificate rekeying* or *key rollover*. Do this to prevent private keys from being overused. (The more a key is used, the more susceptible it is to being broken and recovered by an unintended party.)

To rekey and rollover the PKI Services private key, use the REKEY and ROLLOVER operands of the RACF RACDCERT command. The REKEY operand makes a self-signed copy of the original certificate with a new public-private key pair. The ROLLOVER operand finalizes the rekey operation by replacing the use of the original certificate with the new certificate in every key ring to which the original certificate is connected. It also destroys the original private key and copies over information about its serial number base so the new certificate can be used to sign new certificates.

A retired CA certificate can not be used to sign new certificates. However, until it expires, it can be used to verify previously signed certificates.

## Steps to retire and replace the PKI Services CA private key for the PKI templates

The commands used in this procedure are examples based on the following scenario:

### Assumptions:

- The certificate you are rekeying is a CERTAUTH certificate with label 'Local PKI CA'. It was issued by a commercial CA and is being used by PKI Services for the PKI templates as a certificate authority (CA) certificate, making the PKI Services CA a subordinate CA.
- The PCI cryptographic coprocessor will to be used to generate the new key-pair.
- The size of the new private key will be 1024 bits (RACF default size).

Perform the following procedure to rekey and replace the private key.

1. Initiate the rekeying by executing the following RACF command:

```
RACDCERT CERTAUTH REKEY(LABEL('Local PKI CA')) WITHLABEL('Local PKI CA-2') PCICC
```

2. Create a request for a commercial CA to sign the new public key and reissue the certificate. To create a certificate request for the new key and store it in MVS data set 'SYSADM.CERT.REQ', execute the following command:

```
RACDCERT CERTAUTH GENREQ(LABEL('Local PKI CA-2')) DSN('SYSADM.CERT.REQ')
```

**Restriction:** The certificate request data contained in the data set must be sent to, and received from, the commercial CA using the process defined by the CA. Those steps are not included.

3. Receive the newly signed and reissued certificate back from the commercial CA into MVS data set 'SYSADM.CERT.B64'.

4. Add the newly signed certificate into RACF and replace the self-signed rekeyed one by executing the following command:

```
RACDCERT CERTAUTH ADD('SYSADM.CERT.B64')
```

---

5. You are now ready to retire the original certificate and must stop all use of the original private key. Stop the PKI Services daemon.

**Note:** At this point, the original certificate and its private key exist in RACF with label 'Local PKI CA'. The new certificate and its private key exist in a separate entry in RACF with label 'Local PKI CA-2'. You can proceed to rollover the key.

---

6. Finalize the rollover by entering the following command:

```
RACDCERT CERTAUTH ROLLOVER(LABEL('Local PKI CA')) NEWLABEL('Local PKI CA-2')
```

---

7. Restart the PKI Services daemon.
- 

**When you are done:** You have retired and replaced the old PKI Services CA certificate. All the information for the original certificate is updated to reflect the new certificate, including the key ring connections. You can now begin to use the new certificate and its private key. You can continue to use the old certificate for signature verification purposes until it expires. However, you cannot use the old certificate to sign new certificates. Additionally, do not connect the old certificate to any key rings as the default certificate.

## Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 1

The commands used in this procedure are examples based on the following scenario:

### Assumptions:

- The certificate you are rekeying is a CERTAUTH certificate with label 'taca'.
- It was issued by a local CA certificate labeled 'Local RACF CA' that was generated by RACF and is being used by PKI Services for the SAF templates as a certificate authority (CA) certificate.

Perform the following procedure to rekey and replace the private key.

1. Initiate the rekeying by executing the following RACF command:

```
RACDCERT CERTAUTH REKEY(LABEL('taca')) WITHLABEL('taca-2')
```

---

2. Generate a certificate request based on the new self-signed certificate and store it in MVS data set 'SYSADM.CERT.REQ' by executing the following command:

```
RACDCERT CERTAUTH GENREQ(LABEL('taca-2')) DSN('SYSADM.CERT.REQ')
```

---

3. Execute the following command to sign the new certificate:

```
RACDCERT CERTAUTH GENCERT('SYSADM.CERT.REQ')  
SIGNWITH(CERTAUTH LABEL('Local RACF CA'))
```

At this point, the original certificate and its private key exist in RACF with the label 'taca'. The new certificate and its private key exist in a separate entry in RACF with the label 'taca-2'. You can proceed to rollover the key.

---

4. Finalize the rollover by entering the following command:

```
RACDCERT CERTAUTH ROLLOVER(LABEL('taca')) NEWLABEL('taca-2')
```

---

5. Change the certificate label used in the SIGNWITH field in the SAF templates to the new label name.
- 

**When you are done:** You have retired and replaced the old certificate. All the information for the original certificate is updated to reflect the new certificate, including the key ring connections. You can now begin to use the new certificate and its private key. You can continue to use the old certificate for signature verification purposes until it expires. However, you cannot use the old certificate to sign new certificates. Additionally, do not connect the old certificate to any key rings as the default certificate.

## Steps to retire and replace the PKI Services CA private key for the SAF templates: Scenario 2

The commands used in this procedure are examples based on the following scenario:

### Assumptions:

- The certificate you are rekeying is a CERTAUTH certificate with label 'taca'.
- It was a self-signed certificate in RACF and is being used by PKI Services for the SAF templates as a certificate authority (CA) certificate.

Perform the following procedure to rekey and replace the private key.

1. Initiate the rekeying by executing the following RACF command:

```
RACDCERT CERTAUTH REKEY(LABEL('taca'))  
WITHLABEL('taca-2')
```

At this point, the original certificate and its private key exist in RACF with the label 'taca'. The new certificate and its private key exist in a separate entry in RACF with the label 'taca-2'. You can proceed to rollover the key.

---

2. Finalize the rollover by entering the following command:

```
RACDCERT CERTAUTH ROLLOVER(LABEL('taca')) NEWLABEL('taca-2')
```

---

3. Change the certificate label used in the SIGNWITH field in the SAF templates to the new label name.
- 

**When you are done:** You have retired and replaced the old certificate. All the information for the original certificate is updated to reflect the new certificate, including the key ring connections. You can now begin to use the new certificate and its private key. You can continue to use the old certificate for signature verification purposes until it expires. However, you cannot use the old certificate to sign new certificates. Additionally, do not connect the old certificate to any key rings as the default certificate.

## R\_PKIServ (IRRSPX00) callable service

Authorized applications, such as servers, that invoke the R\_PKIServ callable service (IRRSPX00) can request the generation, retrieval, and administration of PKIX-compliant X.509 Version 3 certificates and certificate requests. Applications can request end-user functions or administrative functions related to these requests. You authorize these applications by administering RACF resources in the FACILITY class, based on whether the application requests end-user functions or administrative functions.

See *z/OS Security Server RACF Callable Services* for the details of invoking IRRSPX00.

### Authorizing end-user functions

The end-user functions are:

	<b>EXPORT</b>	Retrieves (exports) a previously requested certificate, or retrieves (exports) the PKI Services registration authority (RA) certificate or the certificate authority (CA) certificate.
	<b>GENCERT</b>	Generates an auto-approved certificate.
	<b>GENRENEW</b>	Generates an auto-approved renewal certificate. (The request submitted is automatically approved.)
	<b>REQCERT</b>	Requests a certificate that an administrator must approve before it is created.
	<b>REQRENEW</b>	Requests certificate renewal. The administrator needs to approve the request before the certificate is renewed.
	<b>RESPOND</b>	Invokes the PKI OCSP responder.
	<b>REVOKE</b>	Revokes a certificate that was previously issued.
	<b>SCEPREQ</b>	Generates a certificate request using Simple Certificate Enrollment Protocol (SCEP).
	<b>VERIFY</b>	Confirms that a given user certificate was issued by this certificate authority and, if so, returns the certificate fields.

For end-user functions, FACILITY class resources protect this interface. Access authority is based on the user ID for the application (the user ID from the ACEE associated with the address space). To determine the user ID for the application, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

The form for the FACILITY class resources is:

	IRR.RPKISERV.function[.ca_domain]	
	function	
	Specifies one of the following end-user function names in the preceding list.	
	<b>NONE</b>	Access is denied.
	<b>READ</b>	Access is permitted based on subsequent access checks against the caller's user ID.
	<b>UPDATE</b>	Access is permitted based on subsequent access checks against the application's user ID.

### CONTROL (or user ID has RACF SPECIAL)

Access is permitted, and no subsequent access checks are made.

#### *ca\_domain*

Optionally specifies the PKI Services certificate authority (CA) domain name. Use this when your installation has established multiple PKI Services CAs and the *CA\_domain* parameter is provided with IRRSPX00.

**Restriction:** If the name of your initial CA domain is longer than 8 characters, you must truncate it to exactly 8 characters when you define the resource name in the FACILITY class.

**Example:** For the GENCERT function, when the *ca\_domain* is named Customers and the *CA\_domain* parameter is provided with IRRSPX00, then the FACILITY class resource controlling the function is IRR.RPKISERV.GENCERT.CUSTOMER. (The name Customers was truncated to CUSTOMER. See the restriction for the *ca\_domain* parameter.) When the *CA\_domain* parameter is not provided with IRRSPX00, the FACILITY class resource is IRR.RPKISERV.GENCERT.

For SAF GENCERT and EXPORT requests where the application has READ and UPDATE access, subsequent access checks are performed against the IRR.DIGTCERT.*function* FACILITY resources. These are identical to the checks the RACDCERT TSO command makes. See *z/OS Security Server RACF Command Language Reference* for more information.

For PKI Services EXPORT, GENCERT, GENRENEW, REQCERT, REQRENEW, RESPOND, REVOKE, SCEPREQ, and VERIFY requests in which the application has READ and UPDATE access, subsequent access checks are performed against the IRR.DIGTCERT.*function* FACILITY resources.

The following table summarizes the access requirements for the user ID whose access is checked.

Table 64. Summary of access authorities required for PKI Services requests

Request	Access
EXPORT	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.EXPORT <ul style="list-style-type: none"> <li>– READ access if PassPhrase is specified or if CertID is specified as PKICACERT.</li> <li>– UPDATE access if the PassPhrase parameter is not specified with IRRSPX00.</li> <li>– CONTROL access if you want to export a PKCS #7 certificate.</li> </ul> </li> </ul>
GENCERT	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.GENCERT — CONTROL access</li> <li>IRR.DIGTCERT.ADD <ul style="list-style-type: none"> <li>– UPDATE access if any hostIdMappings information is specified in the certificate request parameter list or the UserId field in the certificate request parameter list indicates the certificate is being requested for another user other than the caller</li> <li>– READ access otherwise</li> </ul> </li> </ul>
GENRENEW	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.GENRENEW — READ access</li> <li>IRR.DIGTCERT.GENCERT — CONTROL access</li> </ul> <p><b>Note:</b> It is assumed that the calling application has already verified the input certificate using the VERIFY function.</p>



Table 64. Summary of access authorities required for PKI Services requests (continued)

Request	Access
REQCERT	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.REQCERT — READ access</li> </ul>
REQRENEW	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.REQRENEW — READ access</li> </ul> <p><b>Note:</b> It is assumed that the calling application has already verified the input certificate using the VERIFY function.</p>
RESPOND	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.RESPOND — READ access</li> </ul>
REVOKE	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.REVOKE — READ access</li> </ul> <p><b>Note:</b> It is assumed that the calling application has already verified the target certificate using the VERIFY function.</p>
SCEPREQ	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.SCEPREQ — READ access</li> </ul>
VERIFY	<ul style="list-style-type: none"> <li>IRR.DIGTCERT.VERIFY — READ access</li> </ul> <p><b>Note:</b> It is assumed that the calling application has already verified that the end user possesses the private key that correlates to the input certificate.</p>

## Authorizing administrative functions

The administrative functions are:

<b>CERTDETAILS</b>	Get detailed information about one PKI Services issued certificate.
<b>MODIFYCERTS</b>	Change PKI Services issued certificates.
<b>MODIFYREQS</b>	Change PKI Services certificate requests.
<b>QUERYCERTS</b>	Query PKI Services issued certificates.
<b>QUERYREQS</b>	Query PKI Services about certificate requests.
<b>PREREGISTER</b>	Preregister clients who use Simple Certificate Enrollment Protocol (SCEP).
<b>REQDETAILS</b>	Get detailed information about one PKI Services certificate request.

For the all administrative functions, the following single FACILITY class resource protects this interface.

IRR.RPKISERV.PKIADMIN[.ca\_domain]

ca\_domain

Optionally specifies the PKI Services certificate authority (CA) domain name. Use this when your installation has established multiple PKI Services CAs and the CA\_domain parameter is provided with IRRSPX00.

**Restriction:** If the name of your initial CA domain is longer than 8 characters, you must truncate it to exactly 8 characters when you define the resource name in the FACILITY class.

- If the caller is RACF SPECIAL, no further access is necessary.
- Otherwise, the caller needs:
  - READ access to perform read operations (QUERYREQS, QUERYCERTS, REQDETAILS, and CERTDETAILS)

- UPDATE access for the action operations (PREREGISTER, MODIFYREQS and MODIFYCERTS).

**Example:** For administrative functions, when the *ca\_domain* is named Customers and the CA\_domain parameter is provided with IRRSPX00, the FACILITY class resource controlling this interface is IRR.RPKISERV.PKIADMIN.CUSTOMER. (The name Customers was truncated to CUSTOMER. See the restriction for the *ca\_domain* value.) When the CA\_domain parameter is not provided with IRRSPX00, IRR.RPKISERV.PKIADMIN is the name of the FACILITY class resource.

To determine the appropriate access level of the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

**Attention:** UPDATE access to the IRR.RPKISERV.PKIADMIN[*ca\_domain*] resource also controls who can act as PKI Services administrators. PKI Services administrators play a very powerful role in your organization. The decisions they make when managing certificates and certificate requests determine who will access your computer systems and what privileges they will have when doing so.

**Guideline:** Give UPDATE authority to only highly trusted individuals, but avoid allowing these same individuals to have direct access to the end-user functions of the R\_PKIServ callable service described in “Authorizing end-user functions” on page 257. This helps to maintain a secure separation of duties.

---

## Using encrypted passwords for LDAP servers

PKI Services uses an LDAP directory to store certificates. LDAP requires authenticating (binding) to the directory. You can do this by using a distinguished name and passwords. Passwords for binding (to multiple LDAP directories) can be encrypted or in clear text. The UNIX programmer or LDAP programmer or both determine whether or not to use encrypted LDAP bind passwords. You store information about passwords in the PKI Services configuration file, *pkiserv.conf*.

If you do not need the bind password for the LDAP server to be encrypted, you specify the values for Server1, AuthName1 and AuthPwd1 in the *pkiserv.conf* configuration file. If you want the bind password for the LDAP server to be encrypted, you can use either one of the following profiles:

- A profile named IRR.PROXY.DEFAULTS in the FACILITY class (This profile stores default binding information. It is the profile where PKI Services looks when there is no binding information.)
- A profile (you select the name) in the LDAPBIND class. (You can name this profile whatever you want as long as it matches the BindProfile1 value specified in the *pkiserv.conf* configuration file. (See Step 3 on page 75.)

Before creating either of the preceding profiles, the RACF administrator defines the LDAP.BINDPW.KEY profile in the KEYSMSTR class. This profile contains a SSIGNON segment, which holds either the masked or encrypted value for the key that encrypts passwords stored in the RACF database. Then the RACF administrator creates either of the preceding profiles with a PROXY segment that stores the binding information—the server name, bind distinguished name, and password.

## Steps for using encrypted passwords

Perform the following steps to use encrypted LDAP bind passwords:

1. Define a RACF KEYSMSTR class profile by entering the following command, replacing the highlighted value with your own key:

**Example:**

```
RDEFINE KEYSMSTR LDAP.BINDPW.KEY SSIGNON(KEYENCRYPTED(0023528875DECFA))
```

In this example:

- LDAP BIND passwords are masked by using a key saved in the KEYSMSTR class, LDAP.BINDPW.KEY.
- The key is **0023528875DECFA**. (Replace this with your own key.)
- KEYENCRYPTED is specified (rather than KEYMASKED) because ICSF is active. (If ICSF is not active, replace KEYENCRYPTED with KEYMASKED.)

2. Activate the KEYSMSTR class by entering the following command:

```
SETROPTS CLASSACT(KEYSMSTR)
```

3. If you intend to use the LDAPBIND class, for each LDAP directory, create a RACF LDAPBIND class profile by entering the following command:

```
RDEFINE LDAPBIND MY.LDAP.SERVER1
  PROXY(LDAPHOST(ldap://some.ldap.host:389)
  BINDDN('CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US') BINDPW('MYPASS1')
```

Replace the highlighted parameters as follows:

- a. Optionally, replace *MY.LDAP.SERVER1* with the profile name you want to use.
- b. Replace *ldap://some.ldap.host:389* with your LDAP server URL.
- c. Replace *CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US* with the bind DN.
- d. Replace *MYPASS1* with the bind password.

**Note:** All bind DN qualifiers and the bind password are case-sensitive.

4. If you intend to use IRR.PROXY.DEFAULTS instead of the LDAPBIND class for encrypted LDAP bind passwords, execute the following command to create the profile:

```
RDEFINE FACILITY IRR.PROXY.DEFAULTS
  PROXY(LDAPHOST(ldap://some.ldap.host:389)
  BINDDN('CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US') BINDPW('MYPASS1')
```

Replace the highlighted parameters as follows:

- a. Replace *ldap://some.ldap.host:389* with your LDAP server URL.
- b. Replace *CN=JOE USER,OU=POUGHKEEPSIE,O=IBM,C=US* with the bind DN.
- c. Replace *MYPASS1* with the bind password.

**Note:** All bind DN qualifiers and the bind password are case-sensitive.

5. Optionally, check your work by listing the segment with the RLIST command. If you are using the LDAPBIND class, execute the following:

```
RLIST LDAPBIND MY.LDAP.SERVER1 PROXY NORACF
```

## RACF administration for PKI Services

Replace MY.LDAP.SERVER1 with the profile name you used.

**Results:** This command displays information like the following:

CLASS	NAME
LDAPBIND	MY.LDAP.SERVER1

PROXY INFORMATION

LDAPHOST= LDAP://SOME.LDAP.HOST:389

BINDDN= CN=LDAP ADMINISTRATOR,OU=POUGHKEEPSIE,O=IBM,C=US

BINDPW= YES

If you are using the IRR.PROXY.DEFAULTS profile of the FACILITY class, execute the following command:

RLIST FACILITY IRR.PROXY.DEFAULTS PROXY NORACF

**Results:** This command displays information like the following:

CLASS	NAME
FACILITY	IRR.PROXY.DEFAULTS

PROXY INFORMATION

LDAPHOST= LDAP://SOME.LDAP.HOST:389

BINDDN= CN=LDAP ADMINISTRATOR,OU=POUGHKEEPSIE,O=IBM,C=US

BINDPW= YES

---

---

## Part 6. Using the certificate validation service

This part explains how to implement the PKI Services Trust Policy (PKITP) plug-in for OCSF.

- Chapter 17, “PKI Services Trust Policy (PKITP),” on page 265 describes the certificate validation service. It gives an overview of the OCSF plug-in PKITP, describes certificate policies and extensions, and explains additional configuration needed for PKITP and using the Trust Policy API, `CSSM_TP_PassThrough`.



---

## Chapter 17. PKI Services Trust Policy (PKITP)

This chapter:

- Provides an overview of PKITP, the PKI Services Trust Policy plug-in for OCSF
- Describes:
  - Certificate policies
  - Revoke status checking
  - Certificate extensions
  - CRL extensions and CRL entry extensions
- Explains how to perform additional OCEP configuration needed for PKITP
- Describes CSSM\_TP\_PassThrough (the Trust Policy API).

---

### Overview of PKITP

The PKI Services Trust Policy (PKITP) is an OCSF plug-in to perform certificate validation. It supports the following two functions through the implementation of CSSM\_TP\_PassThrough:

- **CertGroupVerify**
- **FreeEvidence**

Server applications running on z/OS can use this function to verify certificates that other network entities (users, other servers, and so forth) present. PKI Services or other certificate authorities may have issued these certificates.

Before using this plug-in, the server administrator must create a SAF key ring containing trusted CA or site certificates (called *anchor* certificates) or use a *virtual* key ring of either CERTAUTH or SITE certificates. See *z/OS Security Server RACF Command Language Reference* for information about creating a SAF key ring using the RACDCERT ADDRING command. See *z/OS Security Server RACF Callable Services* for information about using a virtual key ring with the R\_data1ib callable service.

The server application must attach to and open this key ring using the OCEP DL plug-in. (See *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming* for more information on OCEP and the use of SAF key rings.) The server application must also bind to any needed LDAP directories by attaching to and opening these directories using the OCSF LDAPDL plug-in. These LDAP directories can be internal corporate directories, directories of extranet business partners, directories of public certificate authorities, or combinations of these.

The following figure illustrates this diversity. The uppercase letter boxes are certificate authorities, and the lowercase letter boxes are end-entity certificates.

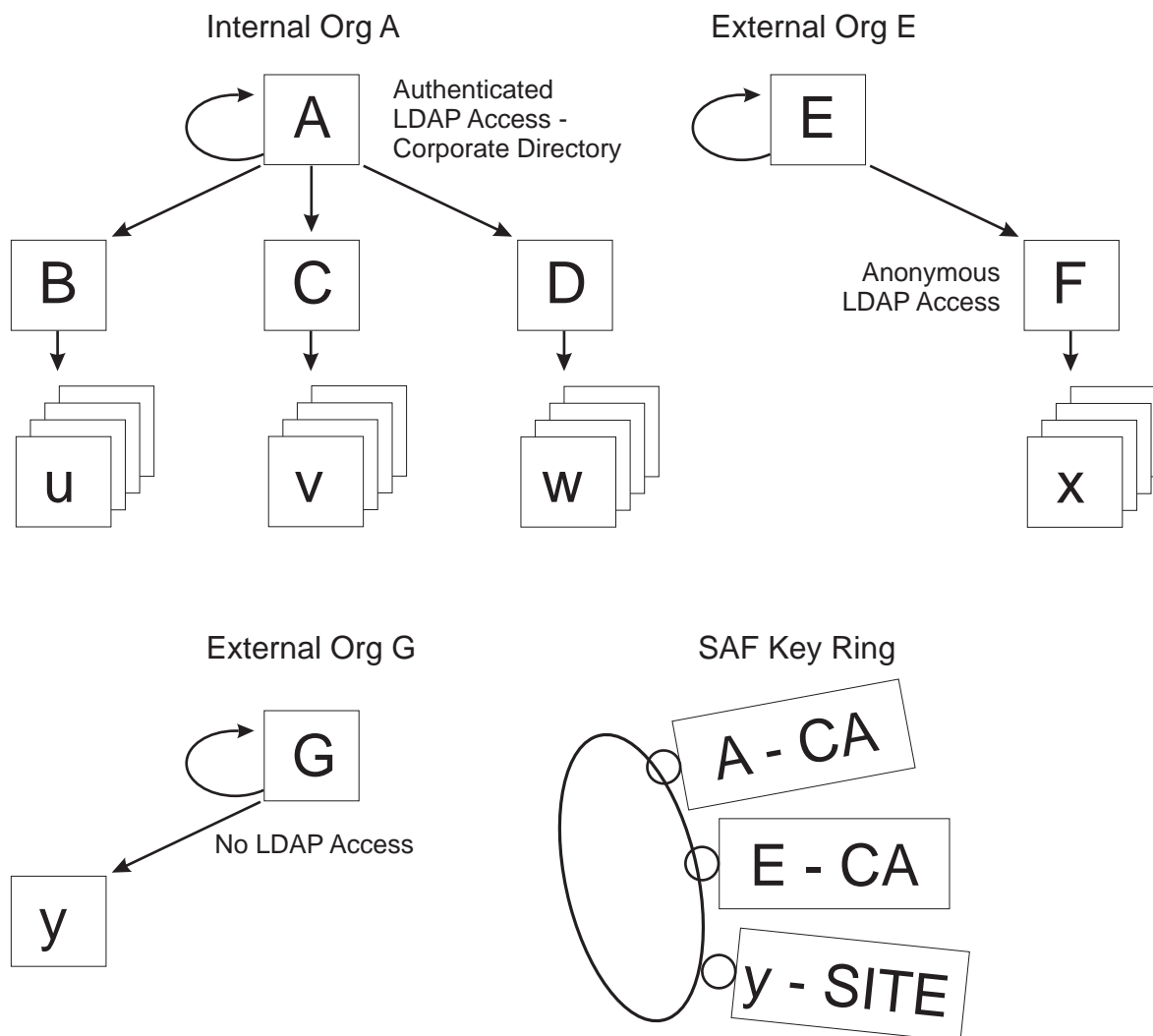


Figure 36. Examples of organizations, certificates, and chains

Organization A represents the local (corporate) certificate hierarchy. It contains one self-signed root certificate "A". Perhaps RACF or the Tivoli® PKI created this. "B", "C", and "D" are intermediate CAs. They could be separate instances of PKI Services. Certificates issued within this hierarchy are stored in an LDAP directory accessible to corporate server applications.

Organization E represents a public or business partner's certificate hierarchy with an LDAP directory that allows anonymous access. Organization G represents some other certificate hierarchy, in which either the directory does not exist or it is not accessible. The key ring contains three anchor certificates. Certificates "A" and "E" are trusted CAs, and there is a business need to trust end-entity certificate "y", even though it cannot be verified.

If each of these CAs has posted current CRLs to either their default LDAP locations or to distribution point CRLs in LDAP and all certificate chains to be verified are genuine, the PKITP **CertGroupVerify** function can validate the following input chains:

- Single certificates x, u, v, or w (PKITP can extract the missing links from the directories.)



- Chains u-B, v-C, w-D, u-B-A, v-C-A, w-D-A, x-F, or x-F-E (These chains have no missing links.)
- Any chain beginning with certificate "y" (As Figure 36 on page 266 shows, "y" is in the key ring as a SITE. Site certificates are trusted regardless.)

Note that, as with the OCEP Trust Policy, non-self-signed (intermediate) CA certificates can be connected to the key ring to shorten the validation path. Doing so has the following consequences:

- Certificate revocation list (CRL) checking is not performed for the anchor certificate in the chain, even if this happens to be an intermediate CA certificate. If the intermediate CA certificate is revoked, PKITP does not detect it.
- A chain containing the parent chain of the intermediate CA cannot be verified.

**Guideline:** When an intermediate CA certificate is connected to the key ring, the certificates that make up its parent chain should be connected as well. This ensures that all chains originating from the intermediate CA or higher can be verified.

---

## Certificate policies

PKITP supports CA and server application-defined certificate policies. CAs can and, in most cases, do establish their own policies for issuing certificates. These policies are declared within issued certificates through the CertificatePolicies extension. When this extension exists and is *not* marked critical, the extension is for informational purposes only—for example, specifying the URL for locating the CA's certificate practice statement (CPS). When this extension exists and *is* marked critical, the policies identified in the extension restrict the use of the certificate. These restrictions apply to subordinate CA certificates and to end-entity certificates. (For information about how PKI Services support the CertificatePolicies extension, see "Using certificate policies" on page 146.)

Similarly, a server application can be a general application that wishes to verify certificates for no specific policy or can be an application that was written for a specific purpose and wishes to verify certificates issued for that purpose (policy).

If the server application specifies an explicit set of policies, then at least one of these policies must be present in each certificate of the certification path (chain). Additionally, PKITP extracts the certificate policies marked critical from each certificate in the chain to determine the intersection—that is, only policies listed in every critically marked CertificatePolicies extension are retained. The server application must indicate that it supports at least one of these policies. If any of these tests is unsuccessful, certificate validation fails.

---

## Checking certificate status with PKITP

PKITP checks the revocation status of a certificate by retrieving certificate revocation lists (CRLs) or, when specified in the certificate, by invoking an online validation service that uses the online certificate status protocol (OCSP).

PKITP certificate revocation checking is performed when useCRLS is set higher than 0. It follows the sequence of validation stages shown in Table 65 on page 268.

Table 65. Sequence of validation stages for PKITP certificate revocation checking

Validation stage	Description
OCSP responder	<p>The trust policy invokes the OCSP responder specified in the AuthInfoAccess extension.</p> <p>If none is specified or if the trust policy fails to receive certificate status from the OCSP responder, it proceeds to the next stage.</p>
DP CRL, using the URI format	<p>The trust policy searches for the DP CRL using the directories, if any, listed in URI format in the CRLDistributionPoints extension in the order they appear.</p> <p>If the DP CRL is found, it is used to determine if the certificate is revoked. If the trust policy fails to find the DP CRL using the URI formats, it proceeds to the next stage.</p>
DP CRL, using the distinguished-name format	<p>The trust policy searches for the DP CRL in the LDAP directories attached through the distinguished name specified, if any, in the CRLDistributionPoints extension.</p> <p>If the trust policy fails to find the DP CRL using the distinguished name and the extension is <i>not</i> marked critical, it proceeds to the next stage.</p> <p>If the trust policy fails to find the DP CRL and the extension is marked critical, the validation fails and error code 8029 (CRL not found) is returned.</p> <p>If DP CRL processing is not to be performed (useCRLS is set to 0) and the target certificate contains a CRLDistributionPoints extension marked critical, validation fails and error code 8029 is returned. No attempt is made to locate the DP CRL.</p>
Global revocation list	<p>The trust policy uses the global CRL to find revocation status information for the certificate.</p>

## Certificate extensions

PKITP supports the following certificate extensions:

<b>AuthorityInformationAccess</b>	Checked for form only.
<b>AuthorityKeyIdentifier</b>	Checked for form only.
<b>BasicConstraints</b>	For CA certificate, cA flag must be on. Also checked for certification path length.
<b>CertificatePolicies</b>	See “Certificate policies” on page 267.
<b>CRLDistributionPoints</b>	See “Checking certificate status with PKITP” on page 267.
<b>HostIdMappings</b>	Checked for form only.
<b>IssuerAltName</b>	Checked for form only. Must be marked critical if the issuer DN is empty.
<b>KeyUsage</b>	For CA certificates, the key CertSign flag must be on.
<b>SubjectAltName</b>	Checked for form only. Must be marked critical if the subject DN is empty.
<b>SubjectKeyIdentifier</b>	Checked for form only.

All other extensions are ignored if they are not marked critical. Unsupported critical extensions prevent certificate validation.

---

## CRL extensions and CRL entry extensions

PKITP supports the following CRL and CRL entry extensions, which are checked for form only:

### CRL extensions:

- AuthorityKeyIdentifier
- CRLNumber
- IssuerAltName
- IssuingDistributionPoint

### CRL entry extensions:

- CertificateIssuer
- CRLReason
- HoldInstructionCode
- InvalidityDate

All other extensions are ignored if they are not marked critical. Unsupported critical extensions prevent certificate validation.

---

## Files for PKITP

The following table lists files for PKITP:

*Table 66. Summary of information about important files for PKITP*

File	Description	Source location (default)
Makefile.pkitsamp	Makefile for pkitsamp.c.	/usr/lpp/pkiserv/samples/
install_pkitsp	Program that registers the PKI Services Trust Policy plug-in with OCSF.	/usr/lpp/pkiserv/bin
pkitsp_ivp	This program verifies that the plug-in installed successfully.	/usr/lpp/pkiserv/bin
pkitsp.h	Contains #define definitions for applications calling the PKI Services OCSF Trust Policy.	/usr/lpp/pkiserv/include/
pkitsp.so	This is the OCSF Trust Policy plug-in for PKI Services.	/usr/lpp/pkiserv/lib
pkitsamp.c	Sample application program (in the C language) to call the PKI Trust Policy plug-in.	/usr/lpp/pkiserv/samples

---

## Configuring and getting started with PKITP

If you have not already installed and configured OCEP, you need to do so now. To install OCEP, you need to follow all of the configuration instructions in *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming* and then perform the following post-installation instructions.

The PKITP must be registered with OCSF before being used.

Steps for configuring PKITP

**Before you begin:** If you have not already done so, run the OCSF and OCEP install and verification scripts.

Perform the following steps to install and configure PKITP:

- 1. Run the PKITP post installation script by entering the following command:

```
/usr/lpp/pkiserv/bin/install_pkitp
```

The program prompts you for certain information. Assuming PKI Services has been installed in its default location, answer the prompts as follows:

Prompt	Response
addin directory?	/usr/lpp/pkiserv/lib
addin filename?	pkitp.so
action? [install   uninstall]	install

You know you are done and that the installation was successful when you see the following:

**Result:**

```
Installing IBMPKITP...
Addin successfully installed.
```

- 2. Update your C/C++ environment variable `_CEE_RUNOPTS` to include `XPLINK(ON)` if it does not already include it. For example, execute the following command from a UNIX shell.

**Example:**

```
export _CEE_RUNOPTS=$_CEE_RUNOPTS' XPLINK(ON) '
```

- 3. To verify that the installation was successful, run the verification program (`/usr/lpp/pkiserv/bin/pkitp_ivp`).

You know you are done and that the verification program ran successfully when you see the following:

```
Starting pkitp IVP
Initializing CSSM
CSSM Initialized
Attaching pkitp
Attach successful, Detaching pkitp
Detach of pkitp successful
Completed pkitp IVP
```

Trust Policy API

Programming Interface information

PKITP supports only one API, `CSSM_TP_PassThrough`. The globally unique identifier (GUID) for this plug-in is: `{01EBC8AC-CC6F-450c-83B4-F0BE0FBE78F9}`. (Before an application can use a module, an installation application must register the module's name, location, and description with OCSF. The name given to a module includes both a logical name and a GUID. The logical name is a string the module developer chooses to describe the module. The GUID is a structure used to differentiate between service provider modules in the OCSF registry.)

## CSSM\_TP\_PassThrough

### Purpose

This function lets applications call TP module-specific operations that have been exported. For PKITP, the module-specific operations support certificate chain validation, based on the CA and SITE certificates that are contained within a key ring.

### Format

```
void * CSSMAPI CSSM_TP_PassThrough
    (CSSM_TP_HANDLE TPHandle,
     CSSM_CL_HANDLE CLHandle,
     CSSM_DL_HANDLE DLHandle,
     CSSM_DB_HANDLE DBHandle,
     CSSM_CC_HANDLE CCHandle,
     uint32 PassThroughId,
     const void *InputParams)
```

### Parameters

#### *TPHandle*

Handle to this Trust Policy module (PKITP).

#### *CLHandle*

Not used. PKITP ignores this.

#### *DLHandle*

Not used. PKITP ignores this.

#### *DBHandle*

Not used. PKITP ignores this.

#### *CCHandle*

Not used. PKITP ignores this.

#### *PassThroughId*

Used to indicate the pass-through service requested. Two services are provided:

- Service 1 **CertGroupVerify** (TP\_VERIFY\_PASSTHROUGH)
- Service 2 **FreeEvidence** (TP\_FREE\_EVIDENCE\_PASSTHROUGH)

#### *InputParams*

Pointer to the API-caller-provided input parameter structure. The same structure is used for both pass-through functions. It is declared in `pkitp.h` as follows:

```
typedef struct tp_verify_extra {
    /* similar parameters as TP_CertGroupVerify */
    CSSM_CL_HANDLE CLHandle;
    CSSM_DL_DB_LIST_PTR DBList;
    unsigned int reserved; //@L1C
    CSSM_TP_STOP_ON VerificationAbortOn;
    CSSM_CERTGROUP_PTR CertToBeVerified;

    /* extra parameters: input */
    TP_INITIALPOLICY_PTR InitialPolicy;
    time_t CurrentTime;
    time_t ValidationTime;

    /* extra parameters: output */
    CSSM_BOOL result;
    uint32 DLStatusCode;           // Status code from DL failures
    uint32 DLindex;              // Index (from 0) into DBList
    TP_EVIDENCE_PTR Evidence;
} TP_VERIFY_EXTRA, *TP_VERIFY_EXTRA_PTR;
```

## The DB list

This DBList contains one or more handles to open DB stores. The last entry in this list must be a handle to an OCEPDL DB (a real or *virtual* SAF key ring). The key ring is used to declare the list of trusted CA and SITE certificates. Like the OCEP Trust Policy, certificate chains to verify must originate from one of these trusted CAs (anchors) or the end-entity certificate must be one of the SITE certificates. Also like the OCEP Trust Policy, if the security product (SAF) marks any certificate in the candidate chain NOTRUST, the certificate chain fails validation.

The other entries in the list are used for LDAPDL DB stores. PKITP runs through these to locate CRLs and intermediate CA certificates. For each item PKITP requests, the LDAPDLs are queried in the order in which they appear in the list. The search stops the first time an LDAPDL returns an item or when the OCEPDL is reached. No query is made to the OCEPDL to locate CRLs or intermediate CA certificates.

## The initial policy

The following optional, caller-provided and initialized structure defines InitialPolicy. PKITP uses the default values if the structure is not provided:

```
typedef struct tp_initialpolicy {

    /* initial-policy-set */
    uint32 NumberOfPolicyIdentifiers; // number of application specific
                                     // policy OIDs (defaults to 0)
    CSSM_OID_PTR PolicyIdentifiers;  // Address of array of policy OIDs
                                     // or 0
    uint32 useCRLs;                  // 0 - no CRL processing
                                     // 1 - Check CRLs only if current CRLs found
                                     // 2 - Strong CRL checking (default)

    /* initial-explicit-policy indicator */
    CSSM_BOOL initialExplicitPolicy; // If true, indicates PKITP should
                                     // consider policy set critical
                                     // defaults to false

    /* initial-policy-mapping-inhibit indicator */
    CSSM_BOOL initialPolicyMappingInhibit; // not used, ignored

} TP_INITIALPOLICY, *TP_INITIALPOLICY_PTR;
```

## The evidence

The following optional, caller-provided structure defines the evidence. This structure is used to return information relative to the validation decision PKITP makes. The caller must free the data areas returned. (The **FreeEvidence** pass-through function is provided for this.)

```
typedef struct tp_evidence {

    /* valid certification path if validation succeeds */
    CSSM_CERTGROUP_PTR CompleteCertGroup;

    /* relevant CRL if validation fails */
    CSSM_DATA_PTR CRL;

    /* relevant certificate if validation fails */
    CSSM_DATA_PTR Cert;

    /* authority-constrained-policy */
    CSSM_BOOL authAnyPolicy;
    uint32 NumberOfAuthCertPolicyIdentifiers;
    CSSM_OID_PTR AuthCertPolicyIdentifiers;

    /* list of policy mappings that occurred */
```

```

uint32 NumberOfMappedPolicies;
TP_CSSM_OID_PAIR_PTR mappedPolicies;

} TP_EVIDENCE, *TP_EVIDENCE_PTR;

```

## Error codes

Table 67 lists the error codes that are unique to PKI Services OCSF Trust Policy (PKITP).

Table 67. PKI Services OCSF Trust Policy (PKITP) error codes

Decimal value	Error description
8001	Certificate encoding error. Incorrect CertificatePolicies extension.
8002	Certificate policies violation.
8003	Incorrect certificate distinguished name chaining.
8004	Certificate encoding error. Subject name missing.
8006	Incorrect certificate BasicConstraints extension—CA flag off in signing certificate.
8008	Incorrect certificate KeyUsage extension—keyCertSign flag off in signing certificate.
8010	Unsupported AltName form in certificate.
8013	Certificate or CRL encoding error. Signature algorithm mismatch.
8014	Certificate encoding error. Incorrect version.
8015	CRL encoding error. Incorrect version.
8016	Unsupported critical extension in certificate.
8017	Unsupported critical extension in CRL.
8018	Unsupported critical entry extension in CRL.
8019	Certificate encoding error. Duplicate extension.
8020	CRL encoding error. Duplicate extension.
8021	Certificate signature failed verification.
8022	CRL signature failed verification.
8023	Incorrect date range in certificate or CRL. NotAfter earlier than NotBefore.
8024	Certificate's date range is in the future.
8025	Certificate has expired.
8026	CRL's date range is in the future.
8027	CRL has expired.
8028	DBList incorrect, no LDAPDL DBs or non-LDAPDL specified.
8029	CRL not found.
8030	Certificate is revoked.
8031	Unable to build certificate chain.
8033	Certificate not trusted.
8034	Incorrect CRLDistributionPoints extension in certificate.
8501	Unexpected status code returned from accessing LDAPDL.
8502	Unexpected status code returned from accessing OCEPDL.
8503	DBList incorrect, no OCEPDL DB or DB empty.

## Building the sample application to invoke the certificate validation service

To perform certificate validation, your server application calls the CSSM\_TP\_PassThrough API (see CSSM\_TP\_PassThrough on page 271), passing it the certificate chain to verify. The API returns a Boolean value indicating success or failure, along with additional information about the certificate chain. The `pkitpsamp.c` code sample that follows on page “Code sample of the PKITP program (`pkitpsamp.c`)” on page 275 is provided as an aid for developing your own server application. By default, you can find this file in the `/usr/lpp/pkiserv/samples` directory.

### Steps for building the sample application

Perform the following steps to build the sample application:

1. Copy the `pkitpsamp.c` program and `Makefile.pkitpsamp` to the current directory by entering the following commands:

```
cp /usr/lpp/pkiserv/samples/pkitpsamp.c pkitpsamp.c
cp /usr/lpp/pkiserv/samples/Makefile.pkitpsamp Makefile
```

2. Before compiling `pkitpsamp.c`, you need to edit some data (for example, information about how you want the Trust Policy to operate and where your LDAP is located). In the `pkitpsamp.c` code (see “Code sample of the PKITP program (`pkitpsamp.c`)” on page 275), find the section that begins with a block comment that says `// Start of application specific options`. Update the code as necessary up to the block comment that says `// End of application specific options`:

- a. If the number of LDAP servers is not 1, change `NUM_LDAPS`.
- b. Update `ldap_info` by specifying your LDAP server and port (`myldap.mycompany.com:389` in the sample program). If you have more than one LDAP server, you need to provide this information for each LDAP server.
- c. Replace the `"@USERID@/@KEYRINGNAME@"` default value for the `char ringname[ ]` variable in the code sample. Specify either the name of the real SAF key ring containing your trusted CA or site certificates, or the name of the *virtual* key ring that points to all your trusted CA or site certificates.
  - If using a real SAF key ring, specify the owning user ID and ring name of the real SAF key ring. **Example:** `patelusr/ring01`
  - If using a virtual key ring, replace the default value with either `*AUTH/*` or `*SITE/*` to point to all your trusted CA or site certificates, respectively. (The name of a virtual key ring is always an asterisk.)
- d. If necessary, change the value of `useCRLS`:
  - 0 This means using no CRL processing. (You must specify 0 if you have no LDAP servers.)
  - 1 This means querying LDAP for CRLs and processing those found. This is the value in the sample.
  - 2 This means using strong CRL checking. (With strong CRL checking, a valid CRL must be found for each CA certificate in the chain.)
- e. If necessary, change `NUM_POLICIES`, the policies that the application calling PKITP uses. In the sample, this is 2. For each policy, specify the DER-encoded policy information.



- f. If necessary, change `INITIAExplicitPolicy` from the default of `FALSE` to `TRUE` if you want PKITP to require all certificates in the chain to have at least one policydata in the preceding list.
- 
3. Compile and link to produce the executable, `pkitpsamp`, by entering the following command:  

```
make
```
- 
4. Export `LIBPATH` to include `/usr/lpp/pkiserv/lib`.  
**Example:**  

```
export LIBPATH=$LIBPATH:/usr/lpp/pkiserv/lib
```
- 
5. Enable program control by setting the extended attribute for `pkitpsamp`.  
**Example:**  

```
extattr +p pkitpsamp
```

**Restriction:** To execute the `extattr` command with the `+p` option, you must have at least `READ` access to the `BPX.FILEATTR.PROGCTL` resource in `FACILITY` class.
- 
6. Update your C/C++ environment variable `_CEE_RUNOPTS` to include `XPLINK(ON)` if it does not already include it. For example, execute the following command from a UNIX shell.  
**Example:**  

```
export _CEE_RUNOPTS=$_CEE_RUNOPTS' XPLINK(ON)'
```
- 
7. Run the `pkitpsamp.c` in your own directory by entering the following command:  

```
pkitpsamp
```
- 

## Code sample of the PKITP program (pkitpsamp.c)

**Note:** The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the `/usr/lpp/pkiserv/samples` directory.

```

/*****
/* This file contains sample code. IBM provides this code on an
/* 'as is' basis without warranty of any kind, either express or
/* implied, including but not limited to, the implied warranties
/* of merchantability or fitness for a particular purpose.
*****/
/*****
/*
/* Licensed Materials - Property of IBM
/* 5694-A01
/* (C) Copyright IBM Corp. 2001, 2005
/* Status = HKY7720
/*
*****/
/* Sample use of IBM PKITP program
/*
/* Purpose: Program attaches needed CSSM modules, then prompts
/* the user for filename(s) containing DER encoded
/* certificates. The certificate(s) are read from the
/* file, then passed to PKITP for verification.
/* A summary of the results are printed to stdout.
/*
*****/

```

```

/* Caution: In order to run this sample program, modification MUST */
/* BE MADE to several values assigned to the following */
/* variables that are defined between the block comment */
/* containing the text "Start of application specific */
/* options" and the block comment containing the text */
/* "End of application specific options"(without the */
/* quotation marks): */
/* */
/* #define NUM_LDAPS 1 */
/* Define the number of LDAP servers that PKITP should */
/* query for certificates, CRLs and ARLs. This can be 0, */
/* if entire certificate chain will be passed as input to */
/* PKITP AND caller requests to NOT process CRLs/ARLs (see */
/* useCRLs option below). */
/* */
/* struct ldap_info ldapserver[NUM_LDAPS] = */
/* { "@LDAPSERVERNAME:PORTNUMBER@", */
/*   "@LDAPUSER@", */
/*   "@LDAPUSERPASSWORD@" }; */
/* If NUM_LDAPS > 0, then ldapserver array should define */
/* the LDAP server:port, user and password for each LDAP */
/* server. Replace @LDAPSERVERNAME:PORTNUMBER@ with the */
/* appropriate ldap server name and port number (e.g */
/* myldap.mycompany.com:389 ). Replace @LDAPUSER@ with the */
/* appropriate ldap admin user name (e.g cn=root) and */
/* @LDAPUSERPASSWORD@ with the password for the specified */
/* ldap user name (e.g rootpw) */
/* */
/* char keyring[] = "@USERID@/@KEYRINGNAME@"; */
/* Define the SAF keyring containing trusted CA and/or */
/* site certificates. Format is "USERID/keyname". Replace */
/* @USERID@ with the userid of the keyring owner and */
/* @KEYRINGNAME@ with the name of the keyring. (e.g */
/* IBMUSER/CARing) Note that the userid and the keyring */
/* names are case sensitive so the userid is all */
/* uppercase and the keyring name is mixed case in this */
/* example. */
/* */
/* #define USECRLS 1 */
/* Define how the useCRLs option should be set. */
/* Set to 0 if no CRL processing is to be performed */
/* Set to 1, if LDAP is to be queried for CRLs and */
/* process the CRLs found. */
/* Set to 2, for strong CRL checking (With strong CRL */
/* checking, a valid CRL must be found for each CA */
/* certificate in the chain.) */
/* */
/* #define NUM_POLICIES 2 */
/* static unsigned char my_policy1[5] = */
/* {0x06,0x03,0x2a,0x03,0x04}; // DER encoded 2.3.4 */
/* static unsigned char my_policy2[7] = */
/* {0x06,0x05,0x2a,0x03,0x03,0x02,0x01}; // DER 2.3.3.2.1 */
/* CSSM_DATA policydata[NUM_POLICIES] = */
/* {{sizeof(my_policy1),(unsigned char *)my_policy1}, */
/* {sizeof(my_policy2),(unsigned char *)my_policy2}}; */
/* Define the policies that the application calling PKITP */
/* uses. These become important if a certificate in the */
/* certificate chain has a critically marked policy */
/* extension. At least one policy that is listed in such */
/* a critically marked policy extension, must appear in */
/* the list defined here or PKITP will return certificate */
/* policy error. */
/* */
/* #define INITIAExplicitPolicy FALSE */
/* Set to true if you want PKITP to require that all */
/* certificates in chain to have at least one policy */
/* listed by the policydata defined above. */
/* */
/***** */
/* */
/*Change-Activity: */
/* $D0=MG00545, HKY7708, 020306, BRW: Initialize @D0A*/
/* TP_EVIDENCE fields for printEvidence @D0A*/
/* */
/* $D1=MG00547, HKY7708, 020306, BRW: Change to allow compile @D1A*/
/* if NUM_LDAPS or NUM_POLICIES is zero @D1A*/
/* $D2=MG01368, HKY7708, 021030, BRW: Delete attaching of CSP @D2A*/
/* $D3=MG04177, HKY7720, 040614, TCG: Fix memory/init errors @D3A*/
/***** */
#pragma runopts("XPLINK(ON)")
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <cssm.h>
#include <ibmocepd1.h>
#include <ibmswcsp.h>
#include <cssmapi.h>
#include <cssmtype.h>
#include <pkitp.h>
#include <ldapd1.h>

struct ldap_info

```

```

{
char * ldapserver;
char * ldapauthuser;
char * ldapauthpass;
};

//-----
// storage function definitions needed to talk to CSSM
//-----

#ifdef _cplusplus
extern "C"
#endif
void * OurMalloc(size_t size, void * allocRef)
{
    return malloc(size);
}

#ifdef _cplusplus
extern "C"
#endif
void OurFree(void* memPtr, void * allocRef)
{
    free(memPtr);
}

#ifdef _cplusplus
extern "C"
#endif
void * OurRealloc(void * memPtr,
                  size_t size,
                  void * allocRef)
{
    return realloc(memPtr, size);
}

#ifdef _cplusplus
extern "C"
#endif
void * OurCalloc(size_t num,
                 size_t size,
                 void* allocRef)
{
    return calloc(num, size);
}

static CSSM_API_MEMORY_FUNCS memoryFuncs; // used to pass function addresses to CSSM

//-----
// internal function declarations
//-----
int connectTP(char * ringname,
             int number_ldap,
             CSSM_DL_DB_LIST *,
             CSSM_TP_HANDLE *); // @D1C

void disconnectTP(CSSM_DL_DB_LIST *, CSSM_TP_HANDLE);

int buildCertGroup(CSSM_CERTGROUP *, char * [], uint32);
void verifyCertGroup(CSSM_CERTGROUP certgroup,
                    CSSM_DL_DB_LIST * datasources_ptr,
                    CSSM_TP_HANDLE tphandle);

void
reportCertGroupVerify
    (TP_VERIFY_EXTRA extraVerifyInfo);

void printEvidence(TP_EVIDENCE_PTR evidence_ptr);

void freeCertGroup(CSSM_CERTGROUP * certGroupPtr);
//-----
//
// Start of application specific options
//
// The defines and declarations that follow should be altered to fit the
// particular application calling PKITP.
//
//-----
//-----
// Define the number of LDAP servers that PKITP should query for certificates,
// CRLs and ARLs. This can be 0, if entire certificate chain will be passed as
// input to PKITP AND caller requests to NOT process CRLs/ARLs (see useCRLs
// option below).
//
// If NUM_LDAPS > 0, then ldapserver array should define the LDAP server:port,
// user and password for each LDAP server, as this example shows.
//-----
#define NUM_LDAPS 1

#if NUM_LDAPS != 0 // @D1A
struct ldap_info ldapserver[NUM_LDAPS] =

```

```

        { "LDAPSERVERNAME:PORTNUMBER@", // LDAP server:port
          "@LDAPUSER@",                // user
          "@LDAPUSERPASSWORD@" };      // password
    #endif // @D1A

    //////////////////////////////////////
    // Define the SAF keyring containing trusted CA and/or site certificates.
    // Format is "USERID/keyname"
    //////////////////////////////////////
    char keyring[] = "@USERID@/KEYRINGNAME@";

    //////////////////////////////////////
    // Define how the useCRLs option should be set.
    // Set to 0 if no CRL processing to be done
    // Set to 1, if we are to query LDAP for CRLs and process those found
    // Set to 2, for strong CRL checking -- must find CRLs in LDAP.
    //////////////////////////////////////

    #define USECRLS 1

    //////////////////////////////////////
    // Define the policies that the application calling PKITP uses.
    //
    // These become important if a certificate in the certificate chain has a
    // critically marked policy extension. At least one policy
    // that is listed in such a critically marked policy extension, must appear
    // in the list defined here or PKITP will return certificate policy error.
    //////////////////////////////////////

    #define NUM_POLICIES 2

    #if NUM_POLICIES != 0 // @D1A
    static unsigned char my_policy1[5] = {0x06,0x03,0x2a,0x03,0x04}; // DER encoded 2.3.4
    static unsigned char my_policy2[7] = {0x06,0x05,0x2a,0x03,0x03,0x02,0x01}; // DER 2.3.3.2.1

    CSSM_DATA policydata[NUM_POLICIES] = {{sizeof(my_policy1),(unsigned char *)my_policy1},
                                           {sizeof(my_policy2),(unsigned char *)my_policy2}};
    #endif // @D1A

    #define INITIALExplicitPolicy FALSE // Set to true if you want PKITP to require that all
                                        // certificates in chain have at least one policy
                                        // listed by our policydata defined above

    //////////////////////////////////////
    //
    // End of application specific options
    //
    //////////////////////////////////////

    //-----
    // main
    //-----
    int
    main(int argc, char* argv[])
    {

        CSSM_DL_DB_LIST datasources;
        CSSM_TP_HANDLE tphandle = 0;
        CSSM_CERTGROUP certGroup;
        int repeating = 1;
        char buffer[1024];
        int num_certs = 0;
        char * cert_files[25];
        char * next_file;
        char * input;

        int rc;

        rc = connectTP(keyring,NUM_LDAPS, &datasources, &tphandle); // @D1C
        if (rc == 0)
        {
            //////////////////////////////////////
            // prompt for certificates to verify
            //////////////////////////////////////
            do
            {
                num_certs = 0;
                printf("Enter filename(s) of certificate(s). (List EE first). ");
                printf("Blank line to quit.\n");

                if ((input = gets(buffer)) != NULL) // get input line
                {
                    next_file = strtok(input," ");
                    while ((next_file != NULL) && (num_certs < 25)) // tokenize it
                    {
                        cert_files[num_certs] = next_file;
                        num_certs++;
                        next_file = strtok(NULL," ");
                    }
                }
            }
        }
    }

```

```

////////////////////////////////////////
// If we were given a list of files containing certificates, input them to TP
////////////////////////////////////////
if (num_certs > 0)
{
    rc = buildCertGroup(&certGroup, cert_files, num_certs);
    if (rc == 0)
    {
        verifyCertGroup(certGroup, &datasources, tphandle);
        freeCertGroup(&certGroup);
    }
} while (num_certs > 0);
}
disconnectTP(&datasources, tphandle);
}

//-----
// connectTP
//
// Purpose: connect to the datasources PKITP needs
// then connect to the PKITP
//
// Input: ringname - string containing "USERID/ringname" of SAF
//         keyring containing trusted CA and/or SITE certificates
//         number_ldap - number of ldap servers
//         ldapservers - array of ldap_info structures
//
// Output: The CSSM_DL_DB_LIST structure addressed by datasources will have
//         been initialized with the various handles that CSSM_ModuleAttach
//         and CSSM_DL_DbOpen calls have returned
//         The CSSM_TP_HANDLE addressed by tphandle_ptr will have been initialised.
//         int returned will be 0 if successful, -1 if not successful.
//-----

int connectTP(char * ringname,
              int number_ldap,
              CSSM_DL_DB_LIST * datasources_ptr,
              CSSM_TP_HANDLE * tphandle_ptr)    //@D1C
{
    uint32 status = 0;
    int z;
    CSSM_VERSION cssm_version = {CSSM_MAJOR, CSSM_MINOR};
    CSSM_DB_ACCESS_TYPE access = { CSSM_TRUE,
                                   CSSM_FALSE,
                                   CSSM_FALSE,
                                   CSSM_FALSE};

    CSSM_VERSION DL_version;
    CSSM_DL_HANDLE LDAP_dhhandle;
    CSSM_MODULE_INFO* moduleInfoPtr;
    void * voidpPtr;
    CSSM_DB_ACCESS_TYPE accessRequest = { CSSM_TRUE,    // ReadAccess
                                         CSSM_TRUE,    // WriteAccess
                                         CSSM_FALSE,    // PrivilegedMode
                                         CSSM_FALSE }; // Asynchronous

    memoryFuncs.malloc_func = OurMalloc;
    memoryFuncs.free_func = OurFree;
    memoryFuncs.realloc_func = OurRealloc;
    memoryFuncs calloc_func = OurCalloc;
    memoryFuncs.AllocRef = NULL;

    DL_version.Major = IBMOCEPDL_MAJOR_VERSION;
    DL_version.Minor = IBMOCEPDL_MINOR_VERSION;

    datasources_ptr->NumHandles = number_ldap + 1;
    voidpPtr = malloc(sizeof(CSSM_DL_DB_HANDLE)*(number_ldap + 1)); // get storage for DBlist
    if (voidpPtr == NULL) { // @D3A
        printf("connectTP unable to obtain memory: line %d\n", __LINE__); // @D3A
        return -1; // @D3A
    }

    memset(voidpPtr, 0, (sizeof(CSSM_DL_DB_HANDLE)*(number_ldap + 1))); // zero it
    datasources_ptr->DLDBHandle = (CSSM_DL_DB_HANDLE *)voidpPtr;

    if (CSSM_Init(&cssm_version, &memoryFuncs, NULL) != CSSM_OK)
    {
        printf("Failed CSSM_Init: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        return -1;
    }

    //////////////////////////////////////////
    // attach to LDAP and open each LDAP DB
    //////////////////////////////////////////
    #if NUM_LDAPS != 0 // @D1A
    if (number_ldap > 0) // if we have any LDAP sources
    {
        moduleInfoPtr = CSSM_GetModuleInfo((CSSM_GUID*)&LDAPDL_GUID,
                                           CSSM_SERVICE_DL,
                                           CSSM_ALL_SUBSERVICES,
                                           CSSM_INFO_LEVEL_ALL_ATTR);
    }
}

```

```

if (!moduleInfoPtr)
{
    printf("Failed CSSM_GetModuleInfo: %d, line %d\n", CSSM_GetError()->error, __LINE__);
    return -1;
}

LDAP_dhandle = CSSM_ModuleAttach((CSSM_GUID*)&LDAPDL_GUID,
                                &moduleInfoPtr->Version,
                                &memoryFuncs,
                                0,
                                0,
                                0,
                                NULL,
                                NULL);

if (!LDAP_dhandle)
{
    printf("Failed CSSM_ModuleAttach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
    return -1;
}

// connect to multiple database instances

//-----
// fill in LDAP DL authentication information:
// necessary only if user is supplying a name and password
//-----
for (z = 0; z < number_ldap; z++)          // for each LDAP source
{
    LDAP_BIND_PARMS bindParms;
    CSSM_USER_AUTHENTICATION userAuthentication = {0,0};
    CSSM_DATA userCredential = {0,0};
    CSSM_USER_AUTHENTICATION_PTR userAuthenticationPtr = 0;

    datasources_ptr->DLDBHandle[z].DLHandle = LDAP_dhandle;
    if (ldapserver[z].ldapauthuser && ldapserver[z].ldapauthpass) //@D1C
    {
        //-----
        // fill in LDAP DL specific data structure: LDAP_BIND_PARMS
        //-----
        bindParms.DN = ldapserver[z].ldapauthuser;                //@D1C
        bindParms.SASL = 0;
        bindParms.credentials.Data = (uint8 *)ldapserver[z].ldapauthpass; //@D1C
        bindParms.credentials.Length = strlen(ldapserver[z].ldapauthpass)+1; //@D1C
        userCredential.Length = sizeof(LDAP_BIND_PARMS);
        userCredential.Data = (unsigned char*)&bindParms
        userAuthentication.Credential = &userCredential
        userAuthenticationPtr = &userAuthentication
    }

    //-----
    // Open LDAP DL Database
    //-----
    datasources_ptr->DLDBHandle[z].DBHandle = CSSM_DL_DbOpen(LDAP_dhandle,
                                                            ldapserver[z].ldapserver, //@D1C
                                                            &accessRequest,
                                                            userAuthenticationPtr,
                                                            (void *)0);

    if (!datasources_ptr->DLDBHandle[z].DBHandle)
    {
        printf("Failed CSSM_DL_DbOpen %d, line %d\n", CSSM_GetError()->error, __LINE__);
        return -1;
    }

} // end of for each each LDAP source

if (CSSM_FreeModuleInfo(moduleInfoPtr) == CSSM_FAIL)
{
    printf("Failed CSSM_FreeModuleInfo, line %d, error %d\n", __LINE__,
        CSSM_GetError()->error);
    // This is not a catastrophic error, we'll continue
}

} // end if we have any LDAP sources
#endif //@D1A

////////////////////////////////////
// Attach to OCEP DL (to access RACF keyring)
////////////////////////////////////

datasources_ptr->DLDBHandle[number_ldap].DLHandle =
    CSSM_ModuleAttach(&IBMOCEPDL_GUID,
                    &DL_version,
                    &memoryFuncs,
                    0,
                    0,
                    0,
                    NULL,
                    NULL);

if (!datasources_ptr->DLDBHandle[number_ldap].DLHandle)
{
    printf("Failed CSSM_ModuleAttach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
}

```

```

        return -1;
    }

    datasources_ptr->DLDBHandle[number_ldap].DBHandle =
        CSSM_DL_DbOpen(datasources_ptr->DLDBHandle[number_ldap].DLHandle,
                        ringname,
                        &access,
                        NULL,
                        NULL);

    if (!(datasources_ptr->DLDBHandle[number_ldap].DBHandle))
    {
        printf("Failed CSSM_DL_DbOpen %d, line %d\n", CSSM_GetError()->error, __LINE__);
        return -1;
    }

    ////////////////////////////////////////////////////
    // Attach to PKITP
    ////////////////////////////////////////////////////
    moduleInfoPtr = CSSM_GetModuleInfo((CSSM_GUID*)&PKITP_GUID,
                                       CSSM_SERVICE_TP,
                                       CSSM_ALL_SUBSERVICES,
                                       CSSM_INFO_LEVEL_ALL_ATTR);

    if (!moduleInfoPtr)
    {
        printf("Failed CSSM_GetModuleInfo: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        return -1;
    }

    *(tphandle_ptr) = CSSM_ModuleAttach((CSSM_GUID*)&PKITP_GUID,
                                       &moduleInfoPtr->Version,
                                       &memoryFuncs,
                                       0,
                                       0,
                                       0,
                                       NULL,
                                       NULL);

    if (!(*tphandle_ptr))
    {
        printf("Failed CSSM_ModuleAttach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        return -1;
    }

    if (CSSM_FreeModuleInfo(moduleInfoPtr) == CSSM_FAIL)
    {
        printf("Failed CSSM_FreeModuleInfo, line %d, error %d\n", __LINE__,
              CSSM_GetError()->error);
        // This is not a catastrophic error, we'll continue
    }

    return 0;
}

//-----
// disconnectTP
//
// Purpose: to close any open databases and detach any CSSM modules
//          that connectTP attached
//
// Input: The CSSM_DL_DB_LIST structure, CSSM_TP_HANDLE,
//        that were initialized by connectTP.
//
// Output: None
//-----

void disconnectTP(CSSM_DL_DB_LIST * datasources_ptr, CSSM_TP_HANDLE tphandle)
{
    int x;
    int status;

    ////////////////////////////////////////////////////
    // Sever ties to LDAP
    // For each LDAP database opened -- call CSSM_DL_DbClose
    ////////////////////////////////////////////////////

    #if NUM_LDAPS != 0 //@D1A
    for (x = 0; x < datasources_ptr->NumHandles - 1; x++)
    {
        // we close each ldap database separately
        if (datasources_ptr->DLDBHandle[x].DBHandle) // if we opened database
        {
            status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[x]);
            if (status != 0)
            {
                printf("Failed CSSM_DL_DbClose %d, line %d\n", CSSM_GetError()->error, __LINE__);
                // we continue trying to close other stuff
            }
        }
    }
}

    ////////////////////////////////////////////////////

```

```

// Now detach the LDAP module
///////////////////////////////////////////////////////////////////
if (datasources_ptr->DLDBHandle[0].DLHandle)
{
    if ((status = CSSM_ModuleDetach(datasources_ptr->DLDBHandle[0].DLHandle)) != 0)
    {
        printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        // we continue trying to close other stuff
    }
    datasources_ptr->DLDBHandle[0].DLHandle = 0; // clear handle
}
#endif // @D1A

// Say goodbye to OCEP
///////////////////////////////////////////////////////////////////
status = CSSM_DL_DbClose(datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1]);
if (status != 0)
{
    printf("Failed CSSM_DL_DbClose %d, line %d\n", CSSM_GetError()->error, __LINE__);
    // we continue trying to close other stuff
}

if (datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle)
{
    if ((status = CSSM_ModuleDetach(datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle)) != 0)
    {
        printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        // we continue trying to close other stuff
    }
    datasources_ptr->DLDBHandle[datasources_ptr->NumHandles - 1].DLHandle = 0;
}

// Farewell PKITP
///////////////////////////////////////////////////////////////////
if (tphandle)
{
    if ((status = CSSM_ModuleDetach(tphandle)) != 0)
    {
        printf("Failed CSSM_ModuleDetach: %d, line %d\n", CSSM_GetError()->error, __LINE__);
        // we continue trying to close other stuff
    }
}

return;
}

/*****
 * name: buildCertGroup - read certificates from files, set up
 *         CSSM_CERTGROUP to reference input certificates
 *
 * input: CSSM_CERTGROUP * -- addresses uninitialized CSSM_CERTGROUP
 *         certFile - array of strings containing names of files that
 *         have DER encoded certificates to be verified by PKITP
 *         certCount - number of elements (strings) in certFile
 *
 * output: returns CSSM_OK if all certificates read
 *         - CSSM_CERTGROUP will have NumCerts set and CertList
 *         will be the address of array of certificates
 *         returns CSSM_FALSE if error reading a file
 *****/

int buildCertGroup(CSSM_CERTGROUP * certGroupPtr,
                  char * certFile[], uint32 certCount)
{
    FILE * inFile;
    CSSM_DATA * certArray = (CSSM_DATA *) calloc(certCount, sizeof(CSSM_DATA));
    uint32 i, certSize;

    if (certArray == NULL) // If calloc failed, exit now
        return(CSSM_FAIL); // @D3A

    certGroupPtr->NumCerts = certCount;
    certGroupPtr->CertList = certArray;

    for (i=0; i < certCount; i++) {
        inFile = fopen(certFile[i], "rb");
        if (!inFile) {
            printf("File %s could not be opened\n", certFile[i]);
            if (i > 1) // if we've read any certs before this
            {
                certGroupPtr->NumCerts = i - 1; // indicate how many read
                freeCertGroup(certGroupPtr); // free alloc'd storage
            }
            return(CSSM_FAIL);
        }
        /* Find size of certificate file */
        fseek(inFile, 0L, SEEK_END);
        certSize = ftell(inFile);
        rewind(inFile);
    }
}

```



```

/* Read in certificate data*/
certArray[i].Length = certSize;
certArray[i].Data = (uint8 *)calloc(certSize, sizeof(char));
if (certArray[i].Data == NULL) { // If calloc failed @D3A
    if (i > 1) // if we've read any certs before this @D3A
    {
        certGroupPtr->NumCerts = i - 1; // indicate how many read @D3A
        freeCertGroup(certGroupPtr); // free alloc'd storage @D3A
    }
    return(CSSM_FAIL); // @D3A
}
fread(certArray[i].Data, 1, certSize, inFile);
fclose(inFile);
}
return(CSSM_OK);
}

/*****
* name: verifyCertGroup - call the Trust Policy (FINALLY)
*
* purpose: call CSSM_TP_PassThrough (PKITP) to verify certificate(s)
*          call reportCertGroupVerify (internal routine to display
*          results to stdout
*          call CSSM_TP_PassThrough (PKITP) to free storage related to
*          results
*
* input:    CSSM_CERTGROUP containing number of and array of certificates
*          CSSM_DL_DB_LIST containing CSSM handles for LDAP and OCEP
*          CSSM_TP_HANDLE CSSM handle for PKITP
*
* output:   none
*
*****/

void verifyCertGroup(CSSM_CERTGROUP certgroup,
                    CSSM_DL_DB_LIST * datasources_ptr,
                    CSSM_TP_HANDLE tphandle)
{
    ///////////////////////////////////////////////////////////////////
    //
    // While there are only 3 parameters on CSSM_TP_PassThrough call to PKITP:
    // - the CSSM_TP_HANDLE,
    // - the function code "TP_VERIFY_PASSTHROUGH" and
    // - a pointer to the TP_VERIFY_EXTRA structure.
    // TP_VERIFY_EXTRA structure contains many parameters, including the address of
    // TP_INITIALPOLICY structure that can be used to override the default
    // policy settings and the address of TP_VERIFY_EXTRA which PKITP can use
    // to pass back more detailed results.
    ///////////////////////////////////////////////////////////////////
    TP_INITIALPOLICY initialPolicyPreferences;
    TP_EVIDENCE pkixEvidence;
    TP_VERIFY_EXTRA extraVerifyInfo;

    memset(&extraVerifyInfo, 0x00, sizeof(TP_VERIFY_EXTRA)); // @D3A
    ///////////////////////////////////////////////////////////////////
    // The field initialPolicyMappingInhibit in TP_INITIALPOLICY is not used
    // by PKITP, therefore we do not set it.
    ///////////////////////////////////////////////////////////////////
    initialPolicyPreferences.NumberofPolicyIdentifiers = NUM_POLICIES;
    #if NUM_POLICIES != 0 //@D1A
        initialPolicyPreferences.PolicyIdentifiers = policydata;
    #else //@D1A
        initialPolicyPreferences.PolicyIdentifiers = NULL; //@D1A
    #endif //@D1A
    initialPolicyPreferences.initialExplicitPolicy = INITIALExplicitPolicy;
    initialPolicyPreferences.initialPolicyMappingInhibit = CSSM_FALSE;
    initialPolicyPreferences.useCRLs = USECRLs;

    ///////////////////////////////////////////////////////////////////
    // Initialize TP_EVIDENCE fields for printEvidence in case call
    // to PKITP, is not successful.
    pkixEvidence.CompleteCertGroup = NULL; /* @D0A */
    pkixEvidence.CRL = NULL; /* @D0A */
    pkixEvidence.Cert = NULL; /* @D0A */
    ///////////////////////////////////////////////////////////////////

    ///////////////////////////////////////////////////////////////////
    // The following fields in TP_VERIFY_EXTRA are not used by PKITP.
    // CLHandle, PolicyIdentifiers and NumberofPolicyIdentifiers
    // (not to be confused with fields of same name in TP_INITIALPOLICY structure),
    // AnchorCerts and NumberofAnchorCerts.
    // Therefore we do not set these fields below.
    ///////////////////////////////////////////////////////////////////
    extraVerifyInfo.DBList = datasources_ptr;
    extraVerifyInfo.VerificationAbortOn = CSSM_TP_STOP_ON_POLICY;
    extraVerifyInfo.CertToBeVerified = &certgroup
    extraVerifyInfo.InitialPolicy = &initialPolicyPreferences
    extraVerifyInfo.Evidence = &pkixEvidence
    extraVerifyInfo.ValidationTime = time(0);

```

## CSSM\_TP\_PassThrough

```

(void*)CSSM_TP_PassThrough(tphandle,
                           0,
                           0,
                           0,
                           0,
                           TP_VERIFY_PASSTHROUGH,
                           (void *)&extraVerifyInfo);

reportCertGroupVerify(extraVerifyInfo);
(void*)CSSM_TP_PassThrough(tphandle,
                           0,
                           0,
                           0,
                           0,
                           TP_FREE_EVIDENCE,
                           (void *)&extraVerifyInfo);
}

//=====
// function: reportCertGroupVerify
//=====

void
reportCertGroupVerify
(TP_VERIFY_EXTRA extraVerifyInfo)
{
    //-----
    // report success or failure
    //-----
    unsigned int reported_err = CSSM_GetError()->error;

    printf("TP_VERIFY_PASSTHROUGH : ");
    if (CSSM_FALSE == extraVerifyInfo.result)
    {
        printf("FAILED. Error code: %d\n",reported_err);
    }
    else
    {
        printf("PASSED\n");
    }

    //-----
    // report evidence
    //-----
    printEvidence(extraVerifyInfo.Evidence);
}

void printEvidence(TP_EVIDENCE_PTR evidence_ptr)
{
    if (evidence_ptr == NULL) return;
    if (evidence_ptr->CompleteCertGroup)
    {
        printf("CompleteCertGroup was returned containing %d certificates at address %x\n",
            evidence_ptr->CompleteCertGroup->NumCerts,
            evidence_ptr->CompleteCertGroup->CertList);
    }
    else printf("CompleteCertGroup was NULL.\n");

    if (evidence_ptr->CRL)
    {
        printf("CRL was returned of %d bytes (decimal) at address %x\n",
            evidence_ptr->CRL->Length,
            evidence_ptr->CRL->Data);
    }
    else printf("CRL was NULL.\n");

    if (evidence_ptr->Cert)
    {
        printf("Cert (failed certificate) was returned of %d bytes (decimal) at address %x\n",
            evidence_ptr->Cert->Length,
            evidence_ptr->Cert->Data);
    }
    else printf("Cert was NULL.\n");
}

/*****
 * name: freeCertGroup - Free certificate data storage
 *****/

void freeCertGroup(CSSM_CERTGROUP * certGroupPtr)
{
    CSSM_DATA * certArray = certGroupPtr->CertList;
    uint32 i;
    uint32 certCount = certGroupPtr->NumCerts;

    for (i=0; i <= certCount-1; i++)
    {
        free(certArray[i].Data);
    }
}

```

```
}  
free(certArray);  
return;  
}
```

End of Programming Interface information



---

## Part 7. Troubleshooting

This part explains using logs and utilities, including the following:

- Chapter 18, “Using information from SYS1.LOGREC,” on page 289 discusses SYS1.LOGREC, which is used to record unusual runtime events, such as an exception.
- Chapter 19, “Using information from the PKI Services logs,” on page 295 discusses using the PKI Services logs, which are ongoing, to debug problems and explains how to change logging options and display log options settings.
- Chapter 20, “Using PKI Services utilities,” on page 301 explains using the PKI Services utilities.

<b>iclview</b>	Displays the entries in the VSAM issued certificate list (ICL) data set.
<b>vosview</b>	Displays the entries contained in the VSAM ObjectStore data set (request database)
<b>pkiprereg</b>	Creates Simple Certificate Enrollment Protocol (SCEP) preregistration records



## Chapter 18. Using information from SYS1.LOGREC

SYS1.LOGREC keeps records of unusual runtime events, such as exceptions or unexpected return codes from calls to system services. It records hardware errors, selected software errors, and selected system conditions in the LOGREC data set. You can use the LOGREC data set as a starting point for diagnosing a problem. It supplies symptom data about the failure and shows the order in which errors occurred. After you have collected this information, you should report the problem to the IBM support center.

The following table describes the contents of the LOGREC data for PKI Services:

Table 68. LOGREC data for PKI Services

CSECT	Description
IKYP0N IKYP81 IKYP8A IKYP8B	<p>Issued when an ABEND occurs in the one of the CSECTs running on the Monitor Thread.</p> <p><b>Primary symptom string:</b></p> <p><b>Component ID (PIDS):</b> 5752XXPKI</p> <p><b>Load module:</b> IKYPKID#L</p> <p><b>CSECT:</b> IKYP0N, IKYP81, IKYP8A, or IKYP8B</p> <p><b>Recovery routine:</b> ESTEXIT</p> <p><b>Error Information:</b> Consists of an abend code and reason code:</p> <p><b>Abend code:</b> The character <i>S</i> followed by 4 hexadecimal digits or the character <i>U</i> followed by 4 decimal digits.</p> <p><b>Reason code:</b> 8 hexadecimal digits.</p>

## Using information from SYS1.LOGREC

Table 68. LOGREC data for PKI Services (continued)

CSECT	Description
IKYP8A	Issued when an exception is caught in the service thread routine IKYP8A01 or in the services thread request routine IKYP8A02.
	<b>Primary symptom string:</b> <b>Component ID (PIDS):</b> 5752XXPKI <b>Load module:</b> IKYPKID#L <b>CSECT:</b> IKYP8A <b>Failing routine:</b> IKYP8A01 or IKYP8A02 <b>Error information:</b> Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number. <b>Abend code:</b> If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits. <b>Reason code:</b> If present, 8 hexadecimal digits. <b>Facility ID:</b> If present, 3 characters. <b>Message number:</b> If present, 8 hexadecimal digits.
	<b>Secondary symptom string:</b> <b>USER</b> The user ID of the requestor. <b>FUNC</b> A function code of 8 hexadecimal digits.
IKYP8B	Issued when an ABEND occurs in the PC routine (or helper routines).
	<b>Primary symptom string:</b> <b>Component ID (PIDS):</b> 5752XXPKI <b>Load module:</b> IKYPKID#L <b>CSECT:</b> IKYP8B <b>Recovery routine:</b> ARREXIT <b>Error information:</b> Consists of an abend code and a reason code. <b>Abend code:</b> The character <i>S</i> followed by 4 hexadecimal digits or the character <i>U</i> followed by 4 decimal digits. <b>Reason code:</b> 8 hexadecimal digits.



Table 68. LOGREC data for PKI Services (continued)

CSECT	Description
IKYSCHDR	Issued from the <b>dispatcher()</b> function when an exception is caught while creating and posting a CRL to LDAP.
	<b>Primary symptom string:</b> <b>Component ID (PIDS):</b> 5752XXPKI <b>Load module:</b> IKYAPI#L <b>CSECT:</b> IKYSCHDR <b>Failing routine:</b> IKYDSPER <b>Error information:</b> Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number. <b>Abend code:</b> If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits. <b>Reason code:</b> If present, 8 hexadecimal digits. <b>Facility ID:</b> If present, 3 characters. <b>Message number:</b> If present, 8 hexadecimal digits.
	<b>Secondary symptom string:</b> <b>THREAD</b> The string DISPATCHR.
IKYSTART	Issued when an exception occurs during <code>daily_timer()</code> processing (general housekeeping for certificate requests and issued certificates).
	<b>Primary symptom string:</b> <b>Component ID (PIDS):</b> 5752XXPKI <b>Load module:</b> IKYAPI#L <b>CSECT:</b> IKYSTART <b>Failing routine:</b> IKYDAYTM <b>Error information:</b> Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number. <b>Abend code:</b> If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits. <b>Reason code:</b> If present, 8 hexadecimal digits. <b>Facility ID:</b> If present, 3 characters. <b>Message number:</b> If present, 8 hexadecimal digits.
	<b>Secondary symptom string:</b> <b>THREAD</b> The string DAY_TIMR.

## Using information from SYS1.LOGREC

Table 68. LOGREC data for PKI Services (continued)

CSECT	Description
IKYTIMER	Issued when an exception is caught while processing a timer event in <b>wakeup_rtn()</b> .
	<b>Primary symptom string:</b> <b>Component ID (PIDS):</b> 5752XXPKI <b>Load module:</b> IKYOSSRV#L <b>CSECT:</b> IKYTIMER <b>Failing routine:</b> IKYWAKUP <b>Error Information:</b> Consists of <i>either</i> an abend code and a reason code <i>or</i> a facility ID and a message number. <b>Abend code:</b> If present, either the character <i>U</i> followed by 4 decimal digits or the character <i>S</i> followed by 3 hexadecimal digits. <b>Reason code:</b> If present, 8 hexadecimal digits. <b>Facility ID:</b> If present, 3 characters. <b>Message number:</b> If present, 8 hexadecimal digits.
	<b>Secondary symptom string:</b> <b>EVENTFUNC</b> The name of the event routine being processed (postEvt, createEvt, or removeEvt).

---

## Sample LOGREC data

The following is a sample of a LOGREC data for PKI Services:

## Using information from SYS1.LOGREC

```

TYPE:  SYMPTOM RECORD      REPORT:  SOFTWARE EDIT REPORT      DAY YEAR
                                REPORT DATE: 221  01
SCP:   VS 2 REL 3          MODEL:  9672                      HH MM SS.TH
                                SERIAL: 048288                TIME: 19:05:16.02

```

```

SEARCH ARGUMENT ABSTRACT:
  PIDS/5752XXPKI RIDS/IKYPKID#L RIDS/IKYP8A RIDS/IKYP8A01 AB/S0C4
  FLDS/RSNCODE VALU/H00000000

```

```

SYSTEM ENVIRONMENT:
  CPU MODEL: 9672          DATE: 221  01
  CPU SERIAL: 048288      TIME: 19:05:16.02
  SYSTEM:    DCEIMGUI      BCP:  MVS
  RELEASE LEVEL OF SERVICE ROUTINE: HBB7703
  SYSTEM DATA AT ARCHITECTURE LEVEL: 10
  COMPONENT DATA AT ARCHITECTURE LEVEL: 10
  SYSTEM DATA: 00000000 00000000 |.....|

```

```

COMPONENT INFORMATION:
  COMPONENT ID:           5752XXPKI
  COMPONENT RELEASE LEVEL: 7706
  SERVICE RELEASE LEVEL:  HKY7706
  DESCRIPTION OF FUNCTION: PKI SERVICES DAEMON

```

```

PRIMARY SYMPTOM STRING:
  PIDS/5752XXPKI RIDS/IKYPKID#L RIDS/IKYP8A RIDS/IKYP8A01 AB/S0C4
  FLDS/RSNCODE VALU/H00000000

```

SYMPTOM	SYMPTOM DATA	EXPLANATION
-----	-----	-----
PIDS/5752XXPKI	5752XXPKI	COMPONENT IDENTIFIER
RIDS/IKYPKID#L	IKYPKID#L	ROUTINE IDENTIFIER
RIDS/IKYP8A	IKYP8A	ROUTINE IDENTIFIER
RIDS/IKYP8A01	IKYP8A01	ROUTINE IDENTIFIER
AB/S0C4	0C4	ABEND CODE - SYSTEM
FLDS/RSNCODE	RSNCODE	DATA FIELD NAME
VALU/H00000000	00000000	ERROR RELATED HEXADECIMAL VALUE

```

SECONDARY SYMPTOM STRING:
  FLDS/USER VALU/CG422253 FLDS/FUNC VALU/H00000000

```

SYMPTOM	SYMPTOM DATA	EXPLANATION
-----	-----	-----
FLDS/USER	USER	DATA FIELD NAME
VALU/CG422253	G422253	ERROR RELATED CHARACTER VALUE
FLDS/FUNC	FUNC	DATA FIELD NAME
VALU/H00000000	00000000	ERROR RELATED HEXADECIMAL VALUE

```

THE SYMPTOM RECORD DOES NOT CONTAIN FREE FORMAT COMPONENT INFORMATION.
HEX DUMP OF RECORD:

```

```

HEADER
+000  4C831800  00000000  0001221F  19051602  |<C.....|
+010  FF048288  96720000                |..BHO...|

```

Figure 37. Sample LOGREC data (Part 1 of 2)

SYMPTOM RECORD					
+000	E2D9F9F6	F7F2F0F4	F8F2F8F8	FFFFCA5B	SR9672048288...\$
+010	B64312D1	0360F103	40404040	40404040	...J.-1.
+020	4040C4C3	C5C9D4C7	E4C9F5F7	F5F2C8C2	DCEIMGUI5752HB
+030	C2F7F7F0	F3400080	00000000	00000000	B7703 .....
+040	F1F00030	00640070	005C0138	003101A0	10.....*.....
+050	LENGTH(0032) ==> ALL BYTES CONTAIN X'00'.				
+070	E2D9F2F1	F1F0F5F7	F5F2E7E7	D7D2C900	SR21105752XXPKI.
+080	F7F7F0F6	C8D2E8F7	F7F0F640	00000000	7706HKY7706 ....
+090	00000000	00000000	00000000	D7D2C940	.....PKI
+0A0	E28599A5	898385A2	40848185	94969540	SERVICES DAEMON
+0B0	40404040	40404040	40404040	00000000	....
+0C0	00000000	00000000	00000000	00000000	.....
+0D0	00000000	0B41465C	0B414668	0B414699	.....*.....R
+0E0	0B4146A8	0B4146A8	0B4146A8	01000000	...Y...Y...Y....
+0F0	0B4144C8	00000000	00000000	F0F1F2F3	...H.....0123
+100	F4F5F6F7	F8F9C1C2	C3C4C5C6	00680040	456789ABCDEF...
+110	0000000F	0B414530	00000000	0B414374	.....
+120	00000000	F0F00000	00000008	00000008	....00.....
+130	00000000	40E70030	D7C9C4E2	61F5F7F5	.... X..PIDS/575
+140	F2E7E7D7	D2C940D9	C9C4E261	C9D2E8D7	2XXPKI RIDS/IKYP
+150	D2C9C47B	D340D9C9	C4E261C9	D2E8D7F8	KID#L RIDS/IKYP8
+160	C140D9C9	C4E261C9	D2E8D7F8	C1F0F140	A RIDS/IKYP8A01
+170	C1C261E2	F0C3F440	C6D3C4E2	61D9E2D5	AB/S0C4 FLDS/RSN
+180	C3D6C4C5	40E5C1D3	E461C8F0	F0F0F0F0	CODE VALU/H00000
+190	F0F0F040	0B414780	00000001	00000000	000 .....
+1A0	C6D3C4E2	61E4E2C5	D940E5C1	D3E461C3	FLDS/USER VALU/C
+1B0	C7F4F2F2	F2F5F340	C6D3C4E2	61C6E4D5	G422253 FLDS/FUN
+1C0	C340E5C1	D3E461C8	F0F0F0F0	F0F0F0F0	C VALU/H00000000
+1D0	40				

Figure 37. Sample LOGREC data (Part 2 of 2)

---

## Chapter 19. Using information from the PKI Services logs

This chapter explains viewing SYSOUT information. It describes the `_PKISERV_MSG_LEVEL` environment variable and lists subcomponents and message levels you can select. It explains how to display and change logging options.

---

### Viewing SYSOUT information

To start PKI Services, you use the PKISERVD sample procedure (see “PKISERVD sample procedure to start PKI Services daemon” on page 384 for a code sample of the JCL). When you start PKI Services, error and informational messages for the PKISERVD job are written to the STDOUT and STDERR file streams. Unless you change the DD statements that specify STDOUT and STDERR in the PKISERVD sample procedure, PKI Services writes these messages to SYSOUT.

To view the SYSOUT information of a job, you use the Spool Display Search Facility (SDSF) or a comparable facility. If you are using SDSF, you can use the question mark line command (by entering a question mark in the prefix area in front of the file name) to separate the job files, including STDOUT and STDERR. Figure 38 on page 296 shows this.

## Using information from the PKI Services logs

The screenshot shows a terminal window titled "D - gdlvmg15.ws - [43 x 80]". The menu bar includes File, Edit, Transfer, Appearance, Communication, Assist, Window, and Help. The main display area shows the command "SDSF STATUS DISPLAY ALL CLASSES" and "COMMAND INPUT ==>". Below this is a table of jobs with columns: NP, JOBNAME, JobID, Owner, Prty, Queue, C, Pos, SAff, ASys, and Status. The table lists various jobs, including PKISERVD, BPXAS, \$MASCOMM, SUIMGUN, SOFV3VS2, RACF, CSNET, TSO, SYSLOG, INIT, DFSCM, ASCHINT, and BPXAS, with their respective JobIDs, Owners, and Statuses. The status column shows "EXECUTION" for some jobs and "PRINT" for others. The bottom of the window shows a status bar with "04/003" and a connection message: "Connected to remote server/host gdlvmg15.endicott.ibm.com using port 23".

NP	JOBNAME	JobID	Owner	Prty	Queue	C	Pos	SAff	ASys	Status
?	PKISERVD	STC00687	PKISRVD	15	EXECUTION			EIMG	EIMG	
	BPXAS	STC00691	STCUSER	15	EXECUTION			EIMG	EIMG	
	BPXAS	STC00692	STCUSER	15	EXECUTION			EIMG	EIMG	
	BPXAS	STC00693	STCUSER	15	EXECUTION			EIMG	EIMG	
	BPXAS	STC00694	STCUSER	15	EXECUTION			EIMG	EIMG	
	\$MASCOMM	STC00596		15	PRINT	A	1			
	SUIMGUN	TSU00581	SUIMGUN	1	PRINT		2			
	BPXAS	STC00592	STCUSER	1	PRINT		3			
	BPXAS	STC00593	STCUSER	1	PRINT		4			
	BPXAS	STC00595	STCUSER	1	PRINT		5			
	BPXAS	STC00591	STCUSER	1	PRINT		6			
	BPXAS	STC00594	STCUSER	1	PRINT		7			
	SOFV3VS2	STC00541	STCUSER	1	PRINT		8			
	RACF	STC00571	STC1	1	PRINT		9			
	CSNET	STC00545	STC1	1	PRINT		10			
	TSO	STC00544	STC1	1	PRINT		11			
	SYSLOG	STC00549	+MASTER+	1	PRINT		12			
	INIT	STC00550	STC1	1	PRINT		13			
	DFSCM	STC00547	DFS	1	PRINT		14			
	INIT	STC00551	STC1	1	PRINT		15			
	INIT	STC00552	STC1	1	PRINT		16			
	INIT	STC00553	STC1	1	PRINT		17			
	INIT	STC00554	STC1	1	PRINT		18			
	INIT	STC00556	STC1	1	PRINT		19			
	INIT	STC00555	STC1	1	PRINT		20			
	INIT	STC00557	STC1	1	PRINT		21			
	INIT	STC00558	STC1	1	PRINT		22			
	INIT	STC00559	STC1	1	PRINT		23			
	INIT	STC00560	STC1	1	PRINT		24			
	INIT	STC00561	STC1	1	PRINT		25			
	INIT	STC00562	STC1	1	PRINT		26			
	INIT	STC00563	STC1	1	PRINT		27			
	INIT	STC00564	STC1	1	PRINT		28			
	ASCHINT	STC00565	STC1	1	PRINT		29			
	ASCHINT	STC00566	STC1	1	PRINT		30			
	ASCHINT	STC00567	STC1	1	PRINT		31			
	ASCHINT	STC00568	STC1	1	PRINT		32			
	BPXAS	STC00570	STCUSER	1	PRINT		33			
	BPXAS	STC00569	STCUSER	1	PRINT		34			
	BPXAS	STC00572	STCUSER	1	PRINT		35			

Figure 38. Separating the job files

After using the question mark line command, you can select the file you want to view by entering an S before this file name. Figure 39 on page 297 shows this:

```

D - gdlvmg15.ws - [43 x 80]
File Edit Transfer Appearance Communication Assist Window Help
SDSF JOB DATA SET DISPLAY - JOB PKISERV (STC00687) LINE 1-5 (5)
COMMAND INPUT ==> SCROLL ==> CSR
NP DDNAME StepName ProcStep DSID Owner C Dest Rec-Cnt PAGE
JESMSG LG JES2 2 PKISRV A 2
JESJCL JES2 3 PKISRV A 27
JESYSMSG JES2 4 PKISRV A 2
s- STDOUT PKISERV 101 PKISRV A 69,681
STDERR PKISERV 102 PKISRV A 0

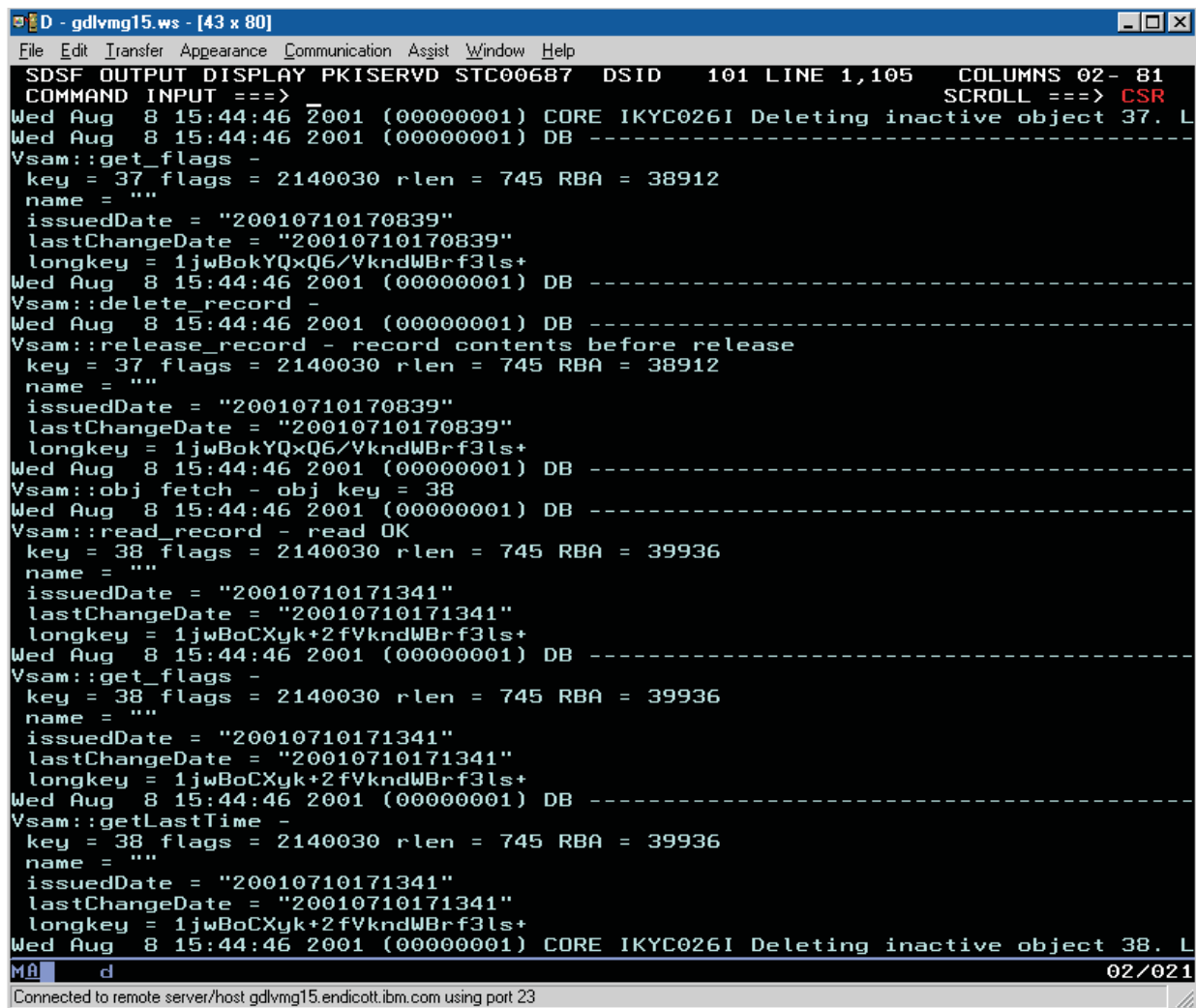
MA d 07/003
Connected to remote server/host gdlvmg15.endicott.ibm.com using port 23

```

Figure 39. Selecting a file to view

Figure 40 on page 298 shows the messages contained in the file:

## Using information from the PKI Services logs



```
D - gdlvmg15.ws - [43 x 80]
File Edit Transfer Appearance Communication Assist Window Help
SDSF OUTPUT DISPLAY PKISERVD STC00687 DSID 101 LINE 1,105 COLUMNS 02- 81
COMMAND INPUT ==>
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 37. L
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::get_flags -
key = 37 flags = 2140030 rlen = 745 RBA = 38912
name = ""
issuedDate = "20010710170839"
lastChangeDate = "20010710170839"
longkey = 1jwBokYQxQ6/VkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::delete_record -
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::release_record - record contents before release
key = 37 flags = 2140030 rlen = 745 RBA = 38912
name = ""
issuedDate = "20010710170839"
lastChangeDate = "20010710170839"
longkey = 1jwBokYQxQ6/VkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::obj fetch - obj key = 38
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::read_record - read OK
key = 38 flags = 2140030 rlen = 745 RBA = 39936
name = ""
issuedDate = "20010710171341"
lastChangeDate = "20010710171341"
longkey = 1jwBoCXyk+2fVkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::get_flags -
key = 38 flags = 2140030 rlen = 745 RBA = 39936
name = ""
issuedDate = "20010710171341"
lastChangeDate = "20010710171341"
longkey = 1jwBoCXyk+2fVkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) DB -----
Vsam::getLastTime -
key = 38 flags = 2140030 rlen = 745 RBA = 39936
name = ""
issuedDate = "20010710171341"
lastChangeDate = "20010710171341"
longkey = 1jwBoCXyk+2fVkndWBrf3ls+
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 38. L
MA d 02/021
Connected to remote server/host gdlvmg15.endicott.ibm.com using port 23
```

Figure 40. Messages contained in the file

### Notes:

1. These messages were produced when Verbose tracing was active.
2. The SYSOUT records have a logical record length of 133, so you may have to scroll to the right to see the entire record.

From left to right, each record contains:

- A time stamp
- The thread identifier, in parenthesis
- The subcomponent name (in the example that follows, this is CORE)
- The message itself, which may span multiple lines

Informational, warning, error, and severe level messages begin with a message number. (See Chapter 21, "Messages," on page 315.) Verbose and diagnostic level messages do not have message numbers and are not documented.

The following is an example of an informational message:

```
Wed Aug 8 15:44:46 2001 (00000001) CORE IKYC026I Deleting inactive object 37.
Last changed at 2001/07/10 17:08:39
```



## **\_PKISERV\_MSG\_LEVEL subcomponents and message levels**

The `_PKISERV_MSG_LEVEL` environment variable specifies the subcomponent and message level for logging messages.

The subcomponents are listed below:

Subcomponent	Meaning
*	The wildcard character (represents all subcomponents)
CORE	The core functions of PKI Services that are not specific to the other subcomponents
DB	Activity related to the request or issued certificate VSAM data stores
LDAP	LDAP posting operations
PKID	The PKI Services daemon address setup and infrastructure
POLICY	Certificate creation and revocation policy processing
SAF	SAF key ring, OCEP, and <code>R_data1ib</code> calls
TPOLICY	Trust policy plug-in processing

The message levels are listed hierarchically below:

Debug level	Meaning
<b>S</b>	This indicates logging only severe messages.
<b>E</b>	This indicates logging severe and error messages.
<b>W</b>	This indicates logging severe, error, and warning messages. This is the <i>default</i> message level for all subcomponents if you do not set the environment variable.
<b>I</b>	This indicates logging severe, error, warning, and informational messages.
<b>D</b>	This indicates logging severe, error, warning, informational, and diagnostic messages.
<b>V</b>	This indicates logging <i>all</i> messages, including verbose diagnostic messages. This is very verbose.

**Guideline:** Do not use **V** level unless IBM support personnel instruct you to do so.

(For information about updating environment variables during configuration, see “Optionally updating PKI Services environment variables” on page 47.)

After PKI Services is up and running, if a problem occurs, the MVS programmer can:

- Change the logging options dynamically—by using the MODIFY (or **F**) console command
- Display the current settings—by using another MODIFY console command.

## **Changing logging options**

To change logging options dynamically, execute the following MODIFY (or **F**) console command:

```
F PKISERVD,LOG sub-component.level[,sub-component.level...]
```

## Using information from the PKI Services logs

*subcomponent.level*

Sets the message level setting(s) for the subcomponent(s). Use one of the subcomponents and message levels listed previously.

---

## Displaying log options settings

To display the current logging options, execute the following MODIFY (or **F**) console command:

```
F PKISERVD,DISPLAY
```

The result of this command is information message IKYP025I. **Sample output:**

```
10.37.39 STC00146: IKYP025I PKI SERVICES SETTINGS:
CA DOMAIN NAME: Customers
SUBCOMPONENT          MESSAGE LEVEL
LDAP                   ERROR MESSAGES AND HIGHER
SAF                    WARNING MESSAGES AND HIGHER
DB                     INFORMATIONAL MESSAGES AND HIGHER
CORE                   WARNING MESSAGES AND HIGHER
PKID                   VERBOSE DIAGNOSTIC MESSAGES AND HIGHER
POLICY                 WARNING MESSAGES AND HIGHER
TPOLICY                WARNING MESSAGES AND HIGHER
MESSAGE LOGGING SETTING: STDOUT_LOGGING
CONFIGURATION FILE IN USE:
/etc/pkiserv/pkiserv.conf
TEMPLATE FILE IN USE:
/etc/pkiserv/pkiserv.tpl
CA CERTIFICATE FINGERPRINTS:
SHA1:   B4 6F EB 2E E0 96 9D 05 32 84 E3 2E D0 3C 10 6F 98 F5 43 3C
MD5:    C2 15 14 AC 12 81 59 46 09 F6 35 ED FB B6 CF 31
```

The PKI administrator can use this command to display the fingerprints of the PKI Services CA certificate in support of Simple Certificate Enrollment Protocol (SCEP) certificate requests. For more information, see “Checking certificate fingerprints” on page 178.

---

## Chapter 20. Using PKI Services utilities

This chapter describes the following UNIX utility programs, which are shipped with PKI Services. These programs are installed in the */install-dir/pkiserv/bin* directory.

<b>iclview</b>	Displays the entries in the VSAM issued certificate list (ICL) data set.
<b>vosview</b>	Displays the entries contained in the VSAM ObjectStore data set (request database)
<b>pkiprereg</b>	Creates Simple Certificate Enrollment Protocol (SCEP) preregistration records

---

### Before you use a PKI Services utility

Update your PATH, LIBPATH, and NLSPATH environment variables with the appropriate pkiserv directory before you run a PKI Services utility. Once you have updated the pkiserv.envars file, the utility programs can be run from the UNIX command line.

Variable name	You must add ...
PATH	<i>/install-dir/pkiserv/bin</i>
LIBPATH	<i>/install-dir/pkiserv/lib</i>
NLSPATH	<i>/install-dir/pkiserv/lib/nls/msg/%L/%N</i>

**Migration action:** Beginning with z/OS V1R8, the command syntax to invoke the **iclview** and **vosview** utility programs is changed. If you used applications and batch programs to invoke these utilities with earlier z/OS releases, these syntax changes might cause errors or unexpected results. If needed, modify your syntax and test your utility applications and batch programs.

## Using the iclview utility

### Purpose

The iclview program displays the entries contained in a VSAM issued certificate list (ICL) data set. Each VSAM ICL record consists of a fixed header followed by a variable-length section containing the BER-encoded certificate. For each entry iclview displays the header information and optionally calls a user-provided program to process the BER-encoded certificate.

### Format

```
iclview {-d vsam-dataset-name [-r] | -c [-p path]}
        [-D CA-domain-name] [-s decode-command-string]
```

### Parameters

You can display usage information about the iclview command format and parameters when you execute the iclview utility command with no parameters.

**-d vsam-dataset-name**

Specifies the MVS data set name of the VSAM issued certificate list (ICL).

**Notes:**

1. Make sure to include the *escape* character, which is a backslash (\), before the quotation marks enclosing the MVS data set name. For example, see '\pkisrvd.vsam.icl\' in “Examples” on page 303.
2. The -d and -c options are mutually exclusive.

**-r** Indicates to open in VSAM record-level sharing (RLS) mode the VSAM data set specified with the -d option.

**-c** Indicates to retrieve the data set name and RLS information from the pkiserv.conf configuration file.

**Notes:**

1. The -c and -d options are mutually exclusive.
2. When you also specify the -D option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

**-p path**

Specifies the directory where the pkiserv.conf configuration file resides. If not specified, the directory defaults to /etc/pkiserv.

**-D CA-domain-name**

Specifies the CA domain name where this utility command is directed.

**Notes:**

1. The -D option is required only if PKI Services is running in multiple-CA mode.
2. The CA-domain-name value can be entered using uppercase or lowercase letters.
3. When you also specify the -c option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

**-s decode-command-string**

Specifies an optional command to call for decoding the ASN.1-encoded data. The command must be able to read and decode binary (BER) data from STDIN.

## Examples

To view the records in VSAM ICL data set 'PKISRVD.VSAM.ICL', passing the certificate to a utility called dumpasn1, use the following:

```
iclview -d \'pkisrvd.vsam.icl\' -s 'dumpasn1 -'
```

**Note:** A dumpasn1 utility is not shipped with PKI Services.

## Output

The fixed header data that is displayed for each record would look like the following:

```
Cert 8: John Q. Public
      ISSUED (Issued certificate)
      Issued at 2001-12-19 17:27:41
      Last changed 2001-12-19 17:42:30
      Subject: CN=John Smith,OU=Class 1 Internet Certificate CA,0=The Firm
      Issuer: OU=PKI Services CA,0=IBM,C=US
      Requester: John Smith
      ApplData: 1YBSSL
      Serial Number: CCD
      Email flag: Off
      Revoked at 2002-12-31 10:31:02
      Revocation Reason: Temporarily suspended
```

An explanation of the lines in the fixed header follows:

The first line specifies certificate's sequential position within the ICL, relative to the other certificates, and requestor's name.

The second line specifies the certificate state, which of one of the following, and a comment (if any):

- ISSUED
- REVOKED, not posted
- REVOKED, awaiting CRL post
- REVOKED, on posted CRL

**Issued at** Indicates when the certificate was issued.

**Last changed** Indicates when the administrator last changed the certificate.

**Subject** Indicates the name of the person who owns the certificate.

**Issuer** Indicates the name of the certificate authority that issued the certificate.

**Requestor** Indicates the requestor's name.

**Appldata** Indicates the 8-character string identifying to the application the short name or nickname of the certificate template. (PKI Services provides sample certificate templates but it is RACF, or an equivalent security product, rather than PKI Services that handles the SAF templates.) The following table shows the nicknames for each certificate template. (These nicknames are supplied in the pkiserv.tmp1 certificate templates file as defaults but your installation may have changed them or added others during customization. See "TEMPLATE sections" on page 103 for more information.)

Table 69. Names, aliases, and nicknames of certificate templates

True name	Alias	Nickname
1-Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1-Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
2-Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
2-Year PKI Authenticode—Code Signing Certificate	PKI Server Certificate	2YIACS
2-Year PKI Windows Logon Certificate	PKI Browser Certificate	2YBWL
5-Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5-Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5-Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
5-Year SCEP Certificate - Preregistration	—	5YSCEPP
<i>n</i> -Year PKI Certificate for Extensions Demonstration	PKI Browser Certificate	SAMPLB
1-Year SAF Browser Certificate	SAF Browser Certificate	—
1-Year SAF Server Certificate	SAF Server Certificate	—

**Serial Number**

Indicates the serial number of the certificate as a hexadecimal number.

**Email flag**

Indicates whether or not to send an expiration warning message. The possible values are 0n or 0ff.

**Revoked at**

Indicates the date and time the certificate was revoked or suspended.

**Revocation Reason**

Indicates one of the following reasons:

- No reason.
- User key was compromised.
- Original use no longer valid.
- CA key was compromised.
- User changed affiliation.
- Certificate was superseded.
- Temporarily suspended.

## Using the vosview utility

### Purpose

The vosview program displays the entries contained in a VSAM ObjectStore data set (the request database). Each VSAM request record consists of a fixed header, followed by a variable-length section. For each entry vosview displays the header information and optionally calls a user-provided program to process the BER-encoded request.

### Format

```
vosview {-d vsam-dataset-name [-r] | -c [-p path]}
        [-D CA-domain-name] [-s decode-command-string]
```

### Parameters

You can display usage information about the vosview command format and parameters when you execute the vosview utility command with no parameters.

**-d** *vsam-dataset-name*

Specifies the MVS data set name of the VSAM ObjectStore.

**Notes:**

1. Make sure to include the *escape* character, which is a backslash (\), before the quotation marks enclosing the MVS data set name. For example, see \'pkisrvd.vsam.ost\' in “Examples” on page 306.
2. The -d and -c options are mutually exclusive.

**-r** Indicates to open in record-level sharing (RLS) mode the VSAM data set specified with the -d option.

**-c** Indicates to retrieve the data set name and RLS information from the pkiserv.conf configuration file.

**Notes:**

1. The -c and -d options are mutually exclusive.
2. When you also specify the -D option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

**-p** *path*

When used with the -c option, specifies the directory where the pkiserv.conf configuration file resides. If not specified, the directory defaults to /etc/pkiserv.

**-D** *CA-domain-name*

Specifies the CA domain name where this utility command is directed.

**Notes:**

1. The -D option is required only if PKI Services is running in multiple-CA mode.
2. The *CA-domain-name* value can be entered using uppercase or lowercase letters.
3. When you also specify the -c option, you must use the -p option to specify the CA domain configuration directory if not /etc/pkiserv.

**-s** *decode-command-string*

Specifies an optional command to call for decoding the ASN.1-encoded data. (The command must be able to read and decode binary (BER) data from STDIN.)

## Examples

To view the records in VSAM ObjectStore data set 'PKISRVD.VSAM.OST', passing the request data to a utility called dumpasn1, use the following:

```
vosview -d \'pkisrvd.vsam.ost\' -s 'dumpasn1 -'
```

**Note:** A dumpasn1 utility is not shipped with PKI Services.

## Output

The fixed header data is displayed for each record. Records 1 and 2 contain system data only. They do not represent certificate requests. Record 2 contains internal system data only.

### Sample record 1

```
Object key = 1
Last used key = 726, CRL serial number = 518, ARL serial number = 518,
High DP = 24, Low DP = 7
name = ""
tid = ??????????????????????
apldata =
comment =
data len = 20
flags = 0 - Type = ??? ObjSt?????
Creation time is: 2002/04/22 17:29:48
Last modified time is: 2002/04/29 18:23:49
```

#### Last used key

The primary index for the last record in the data set.

#### CRL serial number

The number to be used for the next CRL.

#### ARL serial number

The number to be used for the next ARL.

#### High DP

The number of the highest distribution point CRL issued by PKI Services.

#### Low DP

The number of the lowest currently active distribution point CRL.

The remaining fields contain no meaningful data.

### Sample certificate request record

```
Object key = 105
name = "John Q. Public"
tid = 1F45AEF2D3729FA35156BC47
apldata = "1YBSSL"
comment = ""
data len = 570
flags = 1020111 - Type = Cert State = RA CertReqActive [State Flag]
```

**Object key** The index into the VSAM data set name.

**name** The requestor's name.

**tid** The transaction ID data.

**apldata** Indicates the 8-character string identifying to the application the short name or nickname of the certificate template. (PKI Services provides sample certificate templates but it is RACF, or an



equivalent security product, rather than PKI Services that handles the SAF templates.) The following table shows the nicknames for each certificate template. (These nicknames are supplied in the `pkiserv.tmp1` certificate templates file as defaults but your installation may have changed them or added others during customization. See “TEMPLATE sections” on page 103 for more information.)

Table 70. Names, aliases, and nicknames of certificate templates

True name	Alias	Nickname
1-Year PKI SSL Browser Certificate	PKI Browser Certificate	1YBSSL
1-Year PKI S/MIME Browser Certificate	PKI Browser Certificate	1YBSM
2-Year PKI Browser Certificate For Authenticating To z/OS	PKI Browser Certificate	2YBZOS
2-Year PKI Authenticode—Code Signing Certificate	PKI Server Certificate	2YIACS
2-Year PKI Windows Logon Certificate	PKI Browser Certificate	2YBWL
5-Year PKI SSL Server Certificate	PKI Server Certificate	5YSSSL
5-Year PKI IPSEC Server (Firewall) Certificate	PKI Server Certificate	5YSIPS
5-Year PKI Intermediate CA Certificate	PKI Server Certificate	5YSCA
5-Year SCEP Certificate - Preregistration	—	5YSCEPP
<i>n</i> -Year PKI Certificate for Extensions Demonstration	PKI Browser Certificate	SAMPLB
1-Year SAF Browser Certificate	SAF Browser Certificate	—
1-Year SAF Server Certificate	SAF Server Certificate	—

**comment** A comment the administrator supplied the last time the request was updated.

**data len** The length of the variable data portion (that is, the BER-encoded request).

**flags** Represent the current state of the request:

#### Type

**Cert** Certificate request (new or renewal).

**CRL** Certificate revocation list (CRL).

**Rev** Revocation request.

**Post** Certificate waiting to be posted to LDAP.

#### State

The prefix (RA or CA) and one of the following:

**CertPreregistered** Certificate preregistration record.

**CertReqActive** Certificate request in some state of being completed.

**CertSigned** Certificate request

		where the certificate has been created.
	<b>CertReqRejected</b>	Certificate request that has been rejected.
	<b>RevReqActive</b>	Revocation request in some state of being completed.
	<b>CRLWaitingForRA</b>	CRL to be posted to LDAP.
	<b>CertPostPending</b>	Certificate to be posted to LDAP.
	<b>CaInfoPostPending</b>	PKI Services' CA certificate to be posted to LDAP.
<b>State Flag</b>	Optional. If present, is one of the following:	
	<b>Complete</b>	Request is complete. For approved requests, the end user has retrieved the certificate.
	<b>Error</b>	The certificate could not be posted to LDAP.
	<b>NeedsConfirm</b>	Approved or rejected. End user has yet to be notified of the final outcome.

## Using the pkiprereg utility

### Purpose

The pkiprereg program creates Simple Certificate Enrollment Protocol (SCEP) preregistration records in batch.

### Format

```
pkiprereg {-m mode [-t tmpl-file] -s SCEP-file [-o out-file] [-d domain]
           | [-h | -?]}
```

### Parameters

You can display usage information about the pkiprereg command format and parameters when you execute the vosview utility command with the -h or -? option or when you enter an incorrect parameter.

-m *mode*

Indicates one of the following modes of operation for this pkiprereg utility execution.

Mode name	Function
<b>verify</b>	Checks the data file for format errors but does <i>not</i> load the records into PKI Services.
<b>generate</b>	Generates a random 8-character passphrase whenever it finds a passphrase <i>placeholder</i> (the * character) in the file. It does <i>not</i> load the records into PKI Services.
<b>load</b>	Calls PKI Services to load the preregistration records.
<b>remove</b>	Calls PKI Services to remove the preregistration records.

You can specify the mode value as the mode name or any number of initial characters of the mode name. **Example:** You can specify **verify** mode using any of the following options:

```
-m verify
-m ve
-m v
```

-t *tmpl-file*

When used in **verify** or **load** mode, specifies the pathname of the PKI Services certificate templates file (pkiserv.tmpl), which is used as input. When used in other modes, the -t option is ignored.

-s *SCEP-file*

Specifies the pathname of the file containing the preregistration data that is input for all modes.

-o *out-file*

When used in **generate** mode, specifies the pathname of the output file for the preregistration data. When used in other modes, the -o option is ignored.

-d *domain*

When used in **load** or **remove** mode, specifies the PKI Services CA domain name where this utility command is directed.

**Notes:**

- 1. When used in other modes, the -d option is ignored.
- 2. The -d option is required only if PKI Services is running in multiple-CA mode.

3. The *domain* value can be entered using uppercase or lowercase letters.
- h or -?
- Displays the syntax of the pkiprereg command.

## Input

The input for the pkiprereg utility is a data file (*SCEP-file*) containing preregistration records. The rules regarding format of the preregistration records are as follows:

### Rules:

- Each preregistration record consists of multiple consecutive *name=value* pairs, terminated by a blank line or an end-of-file indicator.
- Each line consists of single field name and its value, separated by a = character, forming one *name=value* pair per line.
- Each *name=value* pair (except the Template *name=value* pair) represents data that the client must supply at enrollment time for authentication purposes.
- In pkiprereg **verify** and **load** modes, the Template and ClientName *name=value* pairs are required. All other *name=value* pairs are optional.
- In pkiprereg **remove** mode, only the ClientName *name=value* pair is required.
- Any line beginning with a # character is considered a comment.
- All characters before the = character, except any leading and trailing white space, are considered the *name*.
- All characters after the = character, except any leading and trailing white space, are considered the *value*.
- The field name supplied as a *name* is case-sensitive and must match one the fields names listed in Table 71.
- Line length is limited to 300 characters. After 300 characters, any additional characters are truncated.

The following field names are supported in the preregistration record.

Table 71. List of valid field names for use in the preregistration record as input to the pkiprereg utility

Field name	Maximum length	Description
Template	8 characters	The nickname of the preregistration template to be used.
ClientName	64 characters	The name of the person or device being preregistered. The first 32 characters are case-insensitive and must be unique for each user preregistered in PKI Services.
PassPhrase	32 characters	The password to be communicated to the requestor. The * value can be used as a <i>placeholder</i> when running in <b>generate</b> mode.
SerialNumber	64 characters	Serial number of the subject device.
UnstructAddr	64 characters	Unstructured address of the subject device.
EmailAddr	64 characters	E-mail address for the EMAIL= attribute for subject distinguished name.
Mail	64 characters	E-mail address for the MAIL= attribute for subject distinguished name.
Title	64 characters	Title for subject distinguished name.

Table 71. List of valid field names for use in the preregistration record as input to the pkiprereg utility (continued)

Field name	Maximum length	Description
OrgUnit	64 characters	Organizational unit for subject distinguished name.
Org	64 characters	Organization for subject distinguished name.
Street	64 characters	Street for subject distinguished name.
Locality	64 characters	Locality for subject distinguished name.
StateProv	64 characters	State or province for subject distinguished name.
PostalCode	64 characters	Postal code for subject distinguished name.
Country	2 characters	Country abbreviation for subject distinguished name.
AltIPAddr	15 characters	The IP version 4 address in dotted-decimal format ( <i>nnn.nnn.nnn.nnn</i> ) for subject alternate name extension.
AltURI	255 characters	URI for subject alternate name extension.
AltEmail	100 characters	E-mail address for subject alternate name extension.
AltDomain	100 characters	Domain name for subject alternate name extension.
AltOther	255 characters	Other Name for subject alternate name extension.

For information on the format required for these values, see the R\_PKIServ (IRRSPX00) section of *z/OS Security Server RACF Callable Services*.

## Examples

The following example shows two preregistration records in the `/etc/scep.txt` file.

### Examples:

```
Template=5YSCEPP
ClientName=www.ibm.com
PassPhrase=gumby Org=IBM

Template=5YSCEPP
ClientName=scep.company.com
PassPhrase=*
```

The following command examples produce the actions listed:

### Example:

```
pkiprereg -m v -t /etc/pkiserv/pkiserv.tmpl -s /etc/scep.txt
```

**Action:** Verifies the `/etc/scep.txt` file, indicating that one PassPhrase has an incorrect value, the placeholder (\*).

### Example:

```
pkiprereg -m g -s /etc/scep.txt -o /etc/scepfinal.txt
```

**Action:** Generates a passphrase for the placeholder and saves the records in the `/etc/scepfinal.txt` output file.

## pkiprereg

---

**Example:**

```
pkiprereg -m l -t /etc/pkiserv/pkiserv.tmpl -s /etc/scepfinal.txt
```

**Action:** Loads the preregistration records into PKI Services. If the template identified by the nickname 5YSCEPP contains any subject or alternate name information in the **<CONSTANT>** section, that information overrides any matching information specified in the /etc/scepfinal.txt file.

---

## Part 8. Reference information

This part provides reference information, including listings of code samples for certain important files.

**Note:** The listings in this part might not be identical to the code samples shipped with the product. For the most current sample, see the appropriate source directory.

- Chapter 21, “Messages,” on page 315 explains PKI Services messages.
- Chapter 22, “File directory structure,” on page 341 describes product and file system directories for PKI Services and files contained in them.
- Chapter 24, “Environment variables,” on page 347 explains the pkiserv.envars environment variables file and provides a code sample.
- Chapter 25, “The IKYSETUP REXX exec,” on page 351 explains the contents of the IKYSETUP REXX exec that performs RACF administration and provides a code sample.
- Chapter 26, “Other code samples,” on page 371 provides additional code samples. Table 72 summarizes information about these code samples and those in the preceding chapters, summarizing their use, directory location, and the page where the code sample begins.
- Chapter 27, “SMF recording,” on page 385 describes the content of the System Management Facility (SMF) record that is generated by PKI Services.

Table 72. Summary of information about important files

File	Description	Source location (default)	For code sample...
httpd.conf and httpd2.conf	Contain z/OS HTTP Server directives.	/usr/lpp/pkiserv/samples/	See page 371.
IKYCVSAM	Sample IDCAMS JCL to create VSAM data sets (regardless of whether you are using a sysplex or non-sysplex).	SYS1.SAMPLIB	See page 373.
IKYMVSAM	Sample IDCAMS JCL to create VSAM alternate indexes and PATH data sets if you are migrating from z/OS Version 1 Release 4 or lower.	SYS1.SAMPLIB	See page 377.
IKYRVSAM	Sample IDCAMS JCL to add VSAM record-level sharing (RLS) support. IKYRVSAM reallocates your VSAM data sets in preparation for sharing in a sysplex.	SYS1.SAMPLIB	See page 380.
IKYSETUP	REXX exec to set up RACF profiles.	SYS1.SAMPLIB	See page 351.
pkiserv.conf	PKI Services configuration file.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	See page 343.
PKISERVD	Sample procedure to start PKI Services daemon.	SYS1.PROCLIB	See page 384.
pkiserv.envars	PKI Services environment variables file.	/usr/lpp/pkiserv/samples/ (You might need to copy this file to the runtime directory, /etc/pkiserv.)	See page 349.

Table 72. Summary of information about important files (continued)

File	Description	Source location (default)	For code sample...
pkiserv.tpl	PKI Services certificate template file.	/usr/lpp/pkiserv/samples/ (You copy this file to the runtime directory, /etc/pkiserv.)	Not provided.



## Chapter 21. Messages

PKI Services message numbers begin with the three-character component prefix (IKY), followed by a fourth character that identifies the subcomponent. The following table lists the characters representing various subcomponents and describes where the messages appear.

Table 73. Meaning of fourth character in message number

Character	Meaning	Component producing messages	Where messages appear
C	CORE	Core subcomponent	PKI Services log
D	DB	Database accessing subcomponent	PKI Services log
I	INTERFACE	PKISERV CGIs	In the user's Web browser window
L	LDAP	LDAP bind subcomponent	PKI Services log
O	POLICY	Certificate creation and revocation policy subcomponent	PKI Services log
P	PKID	PKI Services daemon address space controller	<ul style="list-style-type: none"><li>• PKI Services log</li><li>• (For those with destination and routing codes) operators console</li></ul>
S	SAF	SAF interfacing subcomponent	PKI Services log
U	UTILITY	Utility programs	UNIX standard error (stderr)

Characters five through seven are numeric. The eighth character is the message type:

Table 74. Meaning of eighth character in message number

Character	Meaning	Action required
I	Informational (status message)	No action required
E	Eventual action	Possible problem that may require eventual action
A	Action required	Problem that requires immediate attention

For information about setting messages options using environment variables, see page 348.

**IKYC001I**    **Error** *nnnn action-being-performed:*  
*error-code-description*

**Explanation:** PKI Services is processing a request and has encountered an internal error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The request is not processed.

**User response:** Report the error to the IBM support center.

**IKYC002I**    **Error** *nnnn returned from*  
**CP\_NewCertCreate:**  
*error-code-description*

**Explanation:** PKI Services is attempting to create a certificate and has encountered an internal error. The action being performed and the error code encountered

## Messages

are displayed. A description of the error is also displayed, if known.

**System action:** The certificate is not created.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC003I     Error *nnnn* registering the next CRL cutting job: *error-code-description***

**Explanation:** PKI Services has just finished creating the current CRL and is attempting to schedule the next CRL creation thread. An error was encountered. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** Future CRLs are not created until the problem is corrected and PKI Services is restarted.

**System programmer response:** Look for other error messages that may be issued such as IKYC011I. If no other messages were issued, report the error to the IBM support center.

---

### **IKYC004I     Error *nnnn* creating and sending CRLs: *error-code-description***

**Explanation:** PKI Services is attempting to create the current CRL and has encountered an error. The error code encountered is displayed. A description of the error is also displayed, if known. Note: If the error code is an OCSF return code, no error description is displayed. This would indicate a problem posting the CRL to the LDAP directory.

**System action:** If the CRL was created and the post to LDAP was unsuccessful, the post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted. For all other errors, PKI Services tries again to create the CRL during the next CRL interval.

**System programmer response:** If this is a problem with posting to LDAP, you should also see messages IKYC007I or IKYC008I or both. If so, follow the instructions for these messages. Otherwise, report the error to the IBM support center.

---

### **IKYC005I     Error *nnnn* posting {User | CA} Certificate to LDAP for distinguished-name: *error-code-description***

**Explanation:** PKI Services is attempting to post a certificate to the LDAP directory and has encountered an error. The distinguished name for which the post was attempted and the error code encountered are displayed. A description of the error is also displayed, if known. If the error code is an OCSF return code, no error description is displayed.

**System action:** The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted.

**System programmer response:** Determine if the error occurred on the call to LDAP or within PKI Services, based on the presence of an error code description in the message. If no error code description is displayed in the message, the error occurred on the call to LDAP and it is an OCSF error code. Look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. If the error code is LDAP\_NO\_SUCH\_OBJECT (6032), the LDAP entry could not be created because the required suffix does not exist. Check the message to determine the entry that could not be created. If the entry should be posted to LDAP, you need to define the suffix in the LDAP server configuration file, and then stop and restart the LDAP server. For all other LDAP (OCSF) errors, follow the instructions in *z/OS Integrated Security Services LDAP Client Programming*. The true LDAP error code is the rightmost three digits of the OCSF error code. If an error code description is displayed in the message, the error occurred within PKI Services.

If the error code description is Missing LDAP information, then the CreateOUValue directive is missing from the **LDAP** section of the PKI Services configuration file. Add the directive, then stop and restart PKI Services. See Chapter 8, "Tailoring the PKI Services configuration file for LDAP," on page 71 for more information.

Report any other PKI Services error to the IBM support center. If message IKYC009I is also displayed, report that information as well.

---

### **IKYC007I     Error *nnnn* posting {CRL | ARL} to LDAP: *error-code-description***

**Explanation:** PKI Services is attempting to post a CRL or ARL to the LDAP directory and has encountered an error. The error code encountered is displayed. A description of the error is also displayed, if known. Note: If the error code is an OCSF return code, no error description is displayed.

**System action:** The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted.

**System programmer response:** If no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. If the error is LDAPDL\_NO\_SUCH\_OBJECT, the LDAP entry to contain the CRL or ARL does not yet exist. This is expected if you are starting PKI Services for the first time. For all other LDAP errors, follow the instructions in *z/OS Integrated Security Services LDAP*

*Client Programming.* Report non-OCSF errors to the IBM support center. If message IKYC009I is also displayed, report that information as well.

---

**IKYC008I**     **Error *nnnn* creating an {CSSM\_DL\_DB\_PKICA entry for CA Certificate | CSSM\_DL\_DB\_RECORD\_CRL entry for {CRL | ARL} | CSSM\_DL\_DB\_PKIUSER entry for User Cert} to LDAP for distinguished-name:**  
*error-code-description*

**Explanation:** PKI Services is attempting to post a certificate, CRL or ARL to the LDAP directory and has encountered an error. The distinguished name for which the post was attempted and the error code encountered are displayed. A description of the error is also displayed, if known. Note: If the error code is an OCSF return code, no error description is displayed.

**System action:** The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database and deleted.

**System programmer response:** You may also see message IKYC005I or IKYC007I displayed. If so, follow the instructions for the message displayed. Otherwise, if no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. Follow related instructions in *z/OS Integrated Security Services LDAP Client Programming*. Report non-OCSF errors to the IBM support center. If message IKYC009I is also displayed, report that information as well.

---

**IKYC009I**     **LDAP post unsuccessful for object id = *nnnn*, state = *nnnn*, status = *nnnn*:**  
*status-code-description*

**Explanation:** This message appears as supplemental information for messages IKYC005I and IKYC008I.

**System programmer response:** If reporting message IKYC005I or IKYC008I to the IBM support center, report this information as well.

---

**IKYC010I**     **Error *nnnn* returned from action-being-performed:**  
*error-code-description*

**Explanation:** PKI Services is processing a request and has encountered an error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The request is not processed.

**System programmer response:** In some cases the

error code description may be self-explanatory. If not, report the error to the IBM support center.

---

**IKYC011I**     **Bad TimeBetweenCRLs value in pkiserv.conf file: *incorrect-value***

**Explanation:** PKI Services is reading its configuration file to locate the value specified for TimeBetweenCRLs in the **CertPolicy** section. The value specified has an incorrect syntax.

**System action:** CRL processing is suspended until the problem is corrected and PKI Services is restarted.

**System programmer response:** Correct the value and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 51.

---

**IKYC012I**     **Bad CRLDuration value in pkiserv.conf file: *incorrect-value***

**Explanation:** PKI Services is reading its configuration file to locate the value specified for CRLDuration in the **CertPolicy** section. The value specified has an incorrect syntax.

**System action:** CRL processing is suspended until the problem is corrected and PKI Services is restarted.

**System programmer response:** Correct the value and restart PKI Services. For more information, see "(Optional) Steps for updating the configuration file" on page 51.

---

**IKYC013I**     **Bad CreateInterval value in pkiserv.conf file**

**Explanation:** PKI Services is reading its configuration file to locate the value specified for CreateInterval in the **CertPolicy** section. The value specified has an incorrect syntax.

**System action:** PKI Services uses the default value of 3 minutes.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see "(Optional) Steps for updating the configuration file" on page 51.

---

**IKYC014I**     **Bad RemoveCompletedReqs or RemoveInactiveReqs value in pkiserv.conf file**

**Explanation:** PKI Services is reading its configuration file to locate the value specified for either RemoveCompletedReqs or RemoveInactiveReqs in the **ObjectStore** section. The value specified has an incorrect syntax.

**System action:** Completed and inactive requests are not removed until the problem is corrected and PKI Services is restarted.

## Messages

**System programmer response:** Correct the value and restart PKI Services. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

### **IKYC015I      Bad PostInterval value in pkiserv.conf file**

**Explanation:** PKI Services is reading its configuration file to locate the value specified for PostInterval in the LDAP section. The value specified has an incorrect syntax.

**System action:** PKI Services uses the default value of 5 minutes.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “Steps for tailoring the LDAP section of the configuration file” on page 72.

---

### **IKYC016I      action-being-performed returned nnnn in sub-function: error-code-description**

**Explanation:** PKI Services is processing a request and has encountered an internal error. The action being performed, the sub-function that returned the error, and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC017I      JNH\_inquire\_certreq\_startdate (object-id) found neither certificate request nor response (nnnn): error-code-description**

**Explanation:** PKI Services is processing the start date in a request and has encountered an internal error. The request's ID and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC018I      {read | get\_value} of certificate-or-CRL-extension-name returned nnnn: error-code-description**

**Explanation:** PKI Services is processing a CRL or certificate extension field and has encountered an internal error. The field name and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The CRL or certificate is not processed.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC020I      Retrieving CA value failed nnnn: error-code-description**

**Explanation:** PKI Services is processing a certificate extension field in preparation of posting the certificate to the LDAP directory. The processing has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** The certificate is not posted to the LDAP directory.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC021I      CRL claims to have only User and only CA certs**

**Explanation:** PKI Services is processing a CRL extension field in preparation of posting the CRL to the LDAP directory. The processing has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** The CRL is not posted to the LDAP directory.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC022I      Invalid type for object object-id in JNH\_set\_revreq\_invalidDate: error-code-description**

**Explanation:** PKI Services is processing a revocation request and has encountered an internal error. The revocation request's ID and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The revocation request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

### **IKYC023I      Request index (index-number) greater than number of revocations (nnnn) in JNH\_set\_revreq\_invalidDate**

**Explanation:** PKI Services is processing a revocation request and has encountered an internal error.

**System action:** The revocation request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYC024I**     **Failed to schedule event in *nnnn* seconds, status = *nnnn*:  
error-code-description**

**Explanation:** PKI Services is attempting to schedule a timed event and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** The event is not scheduled.

**System programmer response:** Report the error to the IBM support center.

---

**IKYC025I**     **Failed to schedule event status = *nnnn*:  
error-code-description**

**Explanation:** PKI Services is attempting to schedule a timed event and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** The event is not scheduled.

**System programmer response:** Report the error to the IBM support center.

---

**IKYC026I**     **Deleting {inactive | completed} object  
object-id. Last changed at YYYYMMDD  
HH:MM:SS**

**Explanation:** PKI Services is attempting to purge the request database of inactive and completed requests. A request that has met the criteria for deletion has been found. The request's ID is displayed along with information on when it was last changed. This is an informational message only.

**System action:** The request is deleted. PKI Services continues normal processing.

---

**IKYC027I**     **Removing certificate post request after  
*nnnn* unsuccessful attempts**

**Explanation:** PKI Services is attempting to purge the request database of unsuccessful LDAP post requests. A request that has met the criteria for deletion has been found. The number of unsuccessful attempts for this request is displayed. This is an informational message only.

**System action:** The request is deleted. PKI Services continues normal processing.

---

**IKYC028I**     **Export for CertId *certificate-id*  
unsuccessful. Request is still pending  
approval or yet to be issued**

**Explanation:** A client has requested a certificate and is attempting to retrieve it. The retrieval was unsuccessful because the certificate is not yet available. The request either has yet to be approved by a PKI Services administrator or has been approved, but has

not yet been issued by PKI Services. This is an informational message only.

**System action:** The state of the request is unchanged. PKI Services continues normal processing.

**PKI Services administrator response:** Use PKI Services administrative functions to query the request to check its state. If the request is still pending approval, determine whether the request should be approved or rejected and take action accordingly. For more information, see "Processing certificate requests" on page 220.

---

**IKYC029I**     **Error: certificate request type is invalid  
for certificate creation**

**Explanation:** PKI Services is processing a certificate request and has encountered an internal error.

**System action:** The certificate request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYC030I**     **Error *nnnn* retrieving LDAP  
attribute-name attribute data from  
distinguished-name: error-code-  
description**

**Explanation:** PKI Services is trying to retrieve some attribute data from an entry in the LDAP directory and has encountered an error. The attribute name and distinguished name for which the retrieve was attempted and the error code encountered are displayed. If known, a description of the error is also displayed.

**Note:** If the error code is an OCSF return code, no error description is displayed.

**System action:** If the attribute being retrieved is 'MAIL', PKI Services is trying to retrieve the client's e-mail address to send the client a certificate expiration warning message. The warning message is not sent at this time but sending will be tried later.

**System programmer response:** If no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. Follow related instructions in *z/OS Integrated Security Services LDAP Client Programming*. Report non-OCSF errors to the IBM support center.

---

**IKYC031I**     **Error *nnnn* invoking sendmail with  
email address *email-address* retrieved  
from LDAP entry *distinguished-name***

**Explanation:** PKI Services is trying to call the sendmail utility to notify a client that his or her certificate is expiring. The call was unsuccessful. This message displays the e-mail address and distinguished name



## Messages

from which it was retrieved and the error code encountered.

**System action:** The warning message may or may not have been sent. If the e-mail address appears to be genuine, PKI Services retries sending later.

**System programmer response:** Diagnose the problem by consulting the *z/OS Communications Server: IP Diagnosis Guide* and related manuals. Report non-Communications Server errors to the IBM support center.

---

### IKYC032I    Error *nnnn* invoking sendmail with email address *email-address* provided by *distinguished-name*

**Explanation:** PKI Services is trying to notify a client that his or her certificate is either ready or rejected. Notification is accomplished by calling the sendmail utility. The call was unsuccessful. This message displays the e-mail address and the subject's distinguished name from the request. The error code encountered is also displayed.

**System action:** The message may or may not have been sent.

**System programmer response:** Diagnose the problem by consulting the *z/OS Communications Server: IP Diagnosis Guide* and related manuals. Report non-Communications Server errors to the IBM support center.

---

### IKYC033I    Error *nnnn* accessing {ReadyMessageForm | RejectMessageForm | ExpiringMessageForm} *form-value*

**Explanation:** PKI Services is attempting to notify a client that his or her certificate is either ready, rejected, or expiring. The message to be sent is derived by reading the message form from a file or data set specified in the **General** section of the PKI Services configuration file. Either the file name was not specified correctly, or the file read was unsuccessful. The configuration file keyword in error is displayed. The name of the failing file or data set and the error code encountered are also displayed, if known. For ExpiringMessageForm an error code of zero with no file or data set name displayed indicates that the keyword is required but is missing from the PKI Services configuration file.

**System action:** The message is not sent. If this is the expiring warning message, sending will be attempted later.

**System programmer response:** Locate the failing form-typeMessageForm value in the pkiserv.conf file. Make sure that the value specifies the correct file or data set name and that the file or data set exists. If no errors are found, contact your RACF administrator to ensure that the user ID assigned to the PKI Services

daemon has permission to open the file or data set for reading. After making a correction, restart PKI Services. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

### IKYC034I    Error issuing DEQ for resource *resource-name*, return code was *return-code*

**Explanation:** PKI Services background certificate processing has encountered an internal error trying to release control of a resource using the DEQ service. The resource name and return code from the DEQ macro are displayed.

**System action:** PKI Services processing continues. However, further processing of certificate requests may fail until PKI Services is stopped and restarted.

**System programmer response:** Stop and restart PKI Services. If the problem reoccurs, report the error to the IBM support center.

---

### IKYC035I    Bad ExpireWarningTime value in pkiserv.conf file

**Explanation:** PKI Services is reading its configuration file to locate the value specified for ExpireWarningTime in the **CertPolicy** section. The value specified has an incorrect syntax.

**System action:** PKI Services continues, but no expiration warning messages will be issued.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

### IKYC036I    Bad MaxSuspendDuration value in pkiserv.conf file

**Explanation:** PKI Services is reading its configuration file to locate the value specified for MaxSuspendDuration in the **CertPolicy** section. The value specified has an incorrect syntax.

**System action:** CRL processing continues. PKI Services will process as if the suspension grace period is unlimited.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

### IKYC037I    Bad RemoveExpiredCerts value in pkiserv.conf file

**Explanation:** PKI Services is reading its configuration file to locate the value specified for RemoveExpiredCerts in the **ObjectStore** section. The value specified has an incorrect syntax.

**System action:** PKI Services continues, but expired certificates will not be removed from the issued certificate list (ICL).

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC038I      Deleting expired certificate with serial number *certificate-serial-number***

**Explanation:** PKI Services is attempting to purge the issued certificate list (ICL) of expired certificates. A certificate that has met the criteria for deletion has been found. The certificate's serial number is displayed. This is an informational message only.

**System action:** The request is deleted. PKI Services continues normal processing.

---

**IKYC039I      Bad CRLDistName value in pkiserv.conf file**

**Explanation:** PKI Services is initializing and is reading its configuration file to locate the value specified for CRLDistName in the **CertPolicy** section. The value specified does not contain all alphanumeric characters.

**System action:** PKI Services stops.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC040I      Bad CRLDistURI*n* value in pkiserv.conf file: LdapServer*n***

**Explanation:** PKI Services is reading its configuration file to locate the LDAP server specified in the **LDAP** section of the PKI Services configuration file. The value is to be used to create the URI format for the CRLDistributionPoints extension for the LDAP protocol. The Server value specified by LdapServer*n* cannot be found or contains incorrect information.

**System action:** PKI Services continues, but the URI format for that protocol distribution point is not created.

**System programmer response:** Ensure that the CRLDistURI*n* value locates the correct LdapServer*n* or BindProfile*n* value in the **LDAP** section or the default FACILITY class profile, IRR.PROXY.DEFAULTS. For profile values, ensure the profile exists and contains the correct information. Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC041I      Bad CRLDistURI*n* value in pkiserv.conf file, exceeds the number of LDAP servers: LdapServer*n***

**Explanation:** PKI Services is reading its configuration file to locate the LDAP server specified in the **LDAP** section of the PKI Services configuration file. The value is to be used to create the URI format for the CRLDistributionPoints extension for the LDAP protocol. The value *n* indicated in LdapServer*n* is greater than that specified by NumServers.

**System action:** PKI Services continues, but the URI format for that protocol distribution point is not created.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC042I      Bad CRLDistURI*n* format in pkiserv.conf file: CRLDistURI*n***

**Explanation:** PKI Services is reading its configuration file to create the URI format for the CRLDistributionPoints extension. The value specified has incorrect syntax.

**System action:** PKI Services continues, but the URI format for that protocol distribution point is not created.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC043I      Error *nnnn* in creating HFS file *file name* to store distribution point CRL**

**Explanation:** PKI Services is trying to store the distribution point CRL in a file system file. An I/O error occurred during the processing.

**System action:** PKI Services continues, but the distribution point CRL is not created.

**System programmer response:** Fix the I/O error and wait for the next distribution point CRL to be created.

---

**IKYC044I      Bad OCSPTType value in pkiserv.conf file: *value***

**Explanation:** PKI Services responder is reading its configuration file to check if the OCS responder is enabled when receiving an OCS request. The expected value is either 'none' or 'basic'. The value specified is not one of these.

**System action:** The responder is not enabled. The client will get a response back with status 'Try later'.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the

## Messages

configuration file” on page 51.

---

**IKYC045I**      **Unknown section or keyword in pkiserv.conf file: Section: [section], Keyword keyword**

**Explanation:** PKI Services is reading its configuration file during initialization. One of the following conditions occurred:

- An unknown section name was found.
- An unknown keyword was found.
- A valid keyword was placed in the wrong section.
- A keyword was placed before any sections were defined.

**System action:** The keyword is ignored and PKI Services continues.

**System programmer response:** Correct the section name or the keyword and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC046I**      **Incorrect encoding of SCEP {request | PKCS10}**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation contains incorrect ASN.1 encoding.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Correct the SCEP client to produce a correctly encoded ASN.1 request. Report the error to the support center for the provider of your SCEP client.

---

**IKYC047I**      **Incorrect signature on SCEP {request | PKCS10}**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation or its enclosed PKCS #10 certificate request is incorrectly signed or was altered.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Correct the SCEP client to produce a correctly signed request. Report the error to the support center for the provider of your SCEP client.

---

**IKYC048I**      **Unsupported algorithm in SCEP {request | PKCS10}**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation or its enclosed PKCS #10 certificate request contains an

algorithm identifier that is unsupported by PKI Services.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Correct the SCEP client to produce algorithms supported by PKI Services. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC049I**      **SCEP signing certificate is {expired | revoked | not known to PKI Services}**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation contains a signing certificate that is expired, revoked, or unknown to PKI Services. The signing certificate cannot be used to authenticate the SCEP client.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Determine if the SCEP client should request certificates from PKI Services. If so, correct the SCEP client to use a valid signing certificate previously issued by PKI Services. If none exists, reconfigure the SCEP client to remove any existing certificates and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC050I**      **Requested SCEP certificate is {expired | revoked}**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from a SCEP client. The request for a PKI operation requests a certificate that was previously issued by PKI Services but is currently revoked or expired, as indicated by the message displayed. The requested certificate cannot be used by the SCEP client.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.



**IKYC051I Error 0xnnnn decrypting SCEP request**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. PKI Services is attempting to recover the encrypted portion of the request using System SSL services but is unable to do so. The error code returned by System SSL is displayed in the message.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Look up the 0xnnnn error code in *z/OS Cryptographic Service System Secure Sockets Layer Programming*. Most likely, the SCEP client incorrectly encrypted the request, for example by encrypting it for a different host server. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

**IKYC052I Unsupported SCEP message type: nn**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The message type displayed in the message is either unrecognized or unsupported by PKI Services. PKI Services supports only message types PKCSReq (19) and GetCertInitial (20).

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Reconfigure the SCEP client to produce message types supported by PKI Services. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

**IKYC053I SCEP key or name mismatch. SCEP transaction ID: SCEP-transaction-ID**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. While processing the displayed SCEP request transaction ID, PKI Services found one of the following inconsistencies:

- The signing certificate is self-signed but the public key within it does not match the key in the enclosed PKCS #10 request.
- The signing certificate's subject name does not match the subject name in the PKCS #10 request or the subject portion of the IssuerAndSubject field in the SCEP request.
- The signing certificate is not self-signed and not issued by PKI Services.
- The issuer portion of the IssuerAndSubject field in the SCEP request does not identify PKI Services.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Correct the SCEP client to remove the inconsistency. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

**IKYC054I Incorrect reuse of SCEP transaction ID by client name: client-name**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client named in the message. The request contains a SCEP transaction ID that was previously used in a different request. Transaction IDs must uniquely identify a transaction and cannot be reused.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Report the error to the support center for the provider of your SCEP client. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

**IKYC055I Bad PREREGISTER section in certificate template, nickname: template-nickname**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The named certificate template used to preregister the SCEP client contains one of the following errors related to its PREREGISTER section:

- The PREREGISTER section does not exist.
- The PREREGISTER section is incorrectly terminated.
- A required directive is missing.
- A directive has an incorrect value.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Edit the PKI Services certificate templates file (pkiserv.tpl) and correct the error. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

**IKYC056I Template missing from certificate templates file, nickname: template-nickname**

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI

## Messages

operation from a SCEP client. The named certificate template used to preregister the SCEP client no longer exists

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Edit the PKI Services certificate templates file (pkiserv.tpl) and add the template. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC057I**     **SCEP request for client name**  
*client-name* rejected by  
                  {SemiauthenticatedClient  
                  UnauthenticatedClient  
                  SubsequentRequest  
                  RenewalRequest} directive.

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The certificate template used to preregister the named SCEP client contains a directive in the PREREGISTER section indicating that the request should be rejected. The directive is displayed in the message.

**System action:** PKI Services rejects the SCEP request. The rejected certificate request is recorded in the request database.

**System programmer response:** Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new key-pair and a new self-signed certificate. Create or recreate a PKI Services preregistration record as needed. Also ensure that the SCEP client is using a subject name that is consistent with the preregistration record. Make corrections to the SCEP client or delete and recreate the preregistration record as needed. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC058I**     **No preregistration record found for**  
**SCEP request, client name:** *client-name*

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from the named SCEP client. No preregistration record matching the SCEP client was found.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Determine if the SCEP client should request certificates from PKI Services. If so, ensure that a preregistration record exists for the SCEP client and that the client is using a client name that matches the preregistration record. Make corrections to the SCEP client, or delete and recreate the preregistration record as needed. For more

information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC059I**     **No previous SCEP request found for**  
**SCEP GetCertInitial, client name:**  
*client-name*

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request from the named SCEP client. The request is GetCertInitial (a polling operation to pick up a previously requested certificate). PKI Services is unable to locate the previous SCEP request.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Determine if the SCEP client should request certificates from PKI Services. If so, reconfigure the SCEP client to remove any existing certificate request entries and start the certificate request from the beginning using a new a key-pair and new self-signed certificate. Make corrections to the SCEP client, or delete and recreate the preregistration record as needed. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC060I**     **CONSTANT section of certificate**  
**template, nickname:** *template-nickname*  
**contains an incorrect value:**  
*field-name=value*

**Explanation:** PKI Services is processing a Simple Certificate Enrollment Protocol (SCEP) request for a PKI operation from a SCEP client. The certificate template used to preregister the SCEP client is in error. A field found in the <CONSTANT> section of the template contains a missing or incorrect value. The nickname of the certificate template and the incorrect value are displayed in the message.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Edit the certificate templates file and correct the error. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175.

---

**IKYC801I**     *nnnn* bytes of unconsumed data  
**transferring extensions to certificate**  
**template**

**Explanation:** PKI Services is processing a certificate renewal request and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** The certificate renewal request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYC802I**     **Error *nnnn* {getting certificate-section from old certificate | setting certificate-section in certificate template | removing unnecessary extension from certificate template}:**  
*error-code-description*

**Explanation:** PKI Services is processing a certificate renewal request and has encountered an internal error. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** The certificate renewal request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYC901I**     **Error *nnnn* initializing**  
*sub-function-name: error-code-description*

**Explanation:** PKI Services is initializing one of its sub-functions and has encountered an error. The sub-function name and error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** PKI Services stops.

**System programmer response:** This message may accompany a message more specific to the sub-function that failed. Check the log for other error messages issued prior to this one, and diagnose accordingly. Restart PKI Services after making corrections. If you are unable to diagnose the error, report the error to the IBM support center.

---

**IKYC902I**     **Error initializing the configuration file**

**Explanation:** PKI Services is reading its configuration file to locate the object identifiers defined in the **OIDs** section. Either the section is missing, or a value has an incorrect syntax.

**System action:** PKI Services stops.

**System programmer response:** The OID values must be defined in dotted-decimal form, for example:

sha-1WithRSAEncryption=1.2.840.113549.1.1.5

Correct the configuration file, and restart PKI Services. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYC903I**     **Error *nnnn* adding CA certificate to ICL:**  
*error-code-description*

**Explanation:** PKI Services is initializing and is attempting to store its own certificate-authority certificate in the issued certificate list (ICL). The attempt was not successful. The error code encountered is displayed. A description of the error is also displayed, if known.

**System action:** PKI Services stops.

**System programmer response:** This message may accompany a more specific error message. Check the log for other error messages issued prior to this one and diagnose accordingly. Restart PKI Services after making corrections. If you are unable to diagnose the error, report the error to the IBM support center.

---

**IKYD001I**     **Unable to open VSAM data set**  
*data-set-name*

**Explanation:** PKI Services is attempting to open one of the VSAM data sets specified in the **ObjectStore** section of the pkiserv.conf file or its default data set name. The open has failed. The data set name is displayed.

**System action:** PKI Services stops.

**System programmer response:** Locate the failing DSN value in the pkiserv.conf file. Make sure that the value specifies the correct VSAM data set name and that the data set has been created. If the data set name is not specified in the pkiserv.conf file, then PKI Services uses the default name for the data set. Make sure that this data set exists or add the appropriate DSN value to the pkiserv.conf file to specify the correct data set. If migrating from a previous release of PKI Services, make sure that the additional VSAM alternate index data sets have been created properly.

If no errors are found, contact your RACF administrator to ensure that the user ID assigned to the PKI Services daemon has permission to open the data set for update. Once corrected, restart PKI Services. For information about the values specified in the PKI Services configuration file (pkiserv.conf), including their defaults, see “(Optional) Steps for updating the configuration file” on page 51. See also “Steps for creating the VSAM object store and ICL data sets and indexes” on page 79.

---

**IKYI001I**     **Request denied by installation exit.**  
**RC = *nn***

**Explanation:** A user is requesting PKI Services. The PKIServ Web application called an installation-provided exit program. The exit program has determined that the request should be denied. The return code from the exit program is displayed in the message.

**System action:** The request is not performed.

**User response:** Contact your Web administrator.

**Web administrator response:** Determine why the exit program denied the request and correct the program if necessary.

---

**IKYI002I**     **SAF Service IRRSPX00 Returned SAF**  
**RC = *nn* RACF RC = *nn* RACF**  
**RSN = *nn* {diagnostic-information}**

**Explanation:** A user is requesting PKI Services. The PKIServ Web application called the IRRSPX00 SAF

## Messages

callable service as requested. The service was unsuccessful. The diagnostic information that follows the message describes the problem in greater detail.

The text items listed here comprise all of the possible values for *diagnostic-information* in this message and in message IKYU002I.

- 1 Incorrect field name specified in CertPlist:  
    <field-name>.
- 2 <field-name> has an incorrect value.
- 3 Required field <field-name> missing from the request.
- 4 Request denied, not authorized.
- 5 Certificate generation provider is not available [for CA domain <CA-domain-name>].
- 6 Certificate generation provider indicated the following error: <provider-specific-error-msg>.
- 7 Incorrect CertId PassPhrase specified.
- 8 Request has been rejected by the administrator.
- 9 Request is still pending approval or yet to be issued.
- 10 Incorrect certificate specified.
- 11 The certificate could not be {renewed | revoked} because of a state change.
- 12 Incorrect {CertId | Serial Number} specified.
- 13 The status of the {request | certificate} has been changed by another process.
- 14 {CertIds | SerialNums} has an incorrect length.
- 15 CertAnchor area missing.
- 16 CertAnchor area too small.
- 17 CertPlist has an incorrect length.
- 18 CertPlist DiagInfo field missing or has an incorrect length.
- 19 Conflicting field names specified in CertPlist :  
    field-name.
- 20 Incorrect action specified.
- 21 Incorrect status criteria specified.
- 22 Incorrect transaction ID specified.
- 23 Incorrect reason specified.
- 24 Incorrect SerialNum specified.
- 25 SerialNums has an incorrect length.
- 26 Summary list or CertPlist area missing.
- 27 Summary list or CertPlist area too small.
- 28 A parameter list error has been detected.
- 29 An internal error has occurred during RACF processing.
- 30 Unable to establish recovery environment.
- 31 Function code specified is not defined.
- 32 Parameter list version specified is not supported.
- 33 RACF not installed.
- 34 Certificate generation provider internal error.
- 35 Unexpected error.
- 36 Incorrect value specified for CA domain.
- 37 Client already preregistered.

**System action:** The request is not performed.

**User response:** Correct the problem if applicable. If you cannot correct the problem, contact your Web administrator.

**Web administrator response:** Problems 1, 2, and 3 probably indicate an error with the certificate template.

Change the certificate template definition in the pkiserv.tpl file to correct the error.

Problem 4 indicates the user ID assigned to the unit of work calling the IRRSPX00 callable service is not RACF-authorized to perform the request. Determine if the user should have access. If so, use RACF commands to permit the user ID to the required resources.

Problem 5 indicates the PKI Services daemon process has not been started. If PKI Services is configured for multiple-CA mode then the CA domain name is displayed as part of the diagnostic information. Start the correct instance of PKI Services; then retry the request.

For problems 6–13, 22, and 24, or for more information on any of the preceding problems, see earlier chapters in this document and *z/OS Security Server RACF Callable Services*.

For problems 14–21, 23, and 25–35, report the error to the IBM support center.

For problem 36, PKI Services is configured for multiple-CA mode, but the CA domain name as found in the URL contains characters that cannot be used as a CA domain name. Correct the value in the URL; then retry the request.

---

### IKYI003I      PKI Services CGI error in cgi-program-name: diagnostic-error- information

**Explanation:** A user is requesting PKI Services. The PKIServ Web application CGI program processing the request detected a problem. The name of the CGI program and additional diagnostic information is displayed in the message.

**System action:** The request is not performed.

**User response:** Contact your Web administrator.

**Web administrator response:** Locate the CGI program mentioned in the message. (Its default installation location is in a subdirectory under /usr/lpp/pkiserv/PKIServ.) Examine the CGI program's source code to determine the spot where it is failing and why. In most cases, the problem is caused by an error in the PKI Services template file (usually located in /etc/pkiserv/pkiserv.tpl). Correct the problem and retry the request. For more information, see Chapter 11, "Customizing the end-user Web application," on page 91 and Chapter 12, "Customizing the administration Web pages," on page 141.

---

### IKYI004I      Installation exit failed. RC = nn

**Explanation:** A user is requesting PKI Services. The PKIServ Web application called an installation-provided exit program. The exit program either terminated abnormally or returned an unsupported return code value. The return code from the invocation of the exit program is displayed in the message.



**System action:** The request is not performed.

**User response:** Contact your Web administrator.

**Web administrator response:** Determine why the exit program has failed and correct the program as necessary.

---

**IKYL001I      Error *nnnn* {importing | converting}  
LDAP username *user's-distinguished-name*: *error-code-description***

**Explanation:** PKI Services is reading its configuration file to locate one of the values specified for AuthName in the **LDAP** section. The value specified has a syntax error. The incorrect value is displayed. A description of the error is also displayed, if known.

**System action:** PKI Services binds to the LDAP directory anonymously and continues processing. When PKI Services attempts to post certificates and CRLs to this directory, it might fail due to insufficient access. Look for message IKYC007I to determine this is happening. (RC = LDAPDL\_INSUFFICIENT\_ACCESS)

**System programmer response:** Locate the incorrect AuthName value in the pkiserv.conf file and correct it. The value must be specified as an LDAP distinguished name, for example, CN=root,0=IBM. Note: The OID qualifiers must be specified in uppercase and there cannot be any spaces surrounding the equal signs or commas separating the attribute value assertions (AVAs). Make corrections as needed, then stop and restart PKI Services. For more information, see “Steps for tailoring the LDAP section of the configuration file” on page 72.

---

**IKYL002I      LDAP bind to *LDAP-server-domain-name:port* failed, status = *nnnn*:  
*status-code-description***

**Explanation:** PKI Services is attempting to bind to one of the LDAP servers specified in the **LDAP** section of the pkiserv.conf file. The bind has failed. The failing server name is displayed. A description of the error is also displayed, if known. Note: If the error code is an OCSF return code, no error description will be displayed.

**System action:** PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests will remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

**System programmer response:** If no error description is displayed, look up the error code in *z/OS Open Cryptographic Services Facility Application Programming*. Diagnose the problem indicated by the return code. For LDAPDL\_SERVER\_DOWN, ensure that your LDAP server is running. If so, you may have specified

the server name incorrectly in the PKI Services configuration file. Locate the failing Server value in the pkiserv.conf file. Correct the value if it does not specify the correct LDAP server domain name and port, then stop and restart PKI Services. For all other LDAP errors, follow the instructions in *z/OS Integrated Security Services LDAP Client Programming*. Report non-OCSF errors to the IBM support center. If message IKYC009I is also displayed, report that information as well. For more information, see “Steps for tailoring the LDAP section of the configuration file” on page 72.

---

**IKYL003I      Incorrect value specified for LDAPBIND or FACILITY Class profile *profile-name***

**Explanation:** PKI Services LDAP bind processing is trying to retrieve its LDAP bind information in preparation for communicating with the LDAP server. The bind information is contained in either an LDAPBIND class profile or the IRR.PROXY.DEFAULTS profile in the FACILITY class. Either the profile does not exist or some of the information is missing or incorrect. The name of the profile in question is displayed.

**System action:** PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

**System programmer response:** Locate the profile name in the PKI Services configuration file and correct it if needed. If the host name is specified as a URL, you cannot specify it as an SSL URL (for example, ldaps://). PKI Services does not use SSL to communicate with the LDAP server. If you make corrections, stop and restart PKI Services. If the profile name is already correct, contact your RACF administrator. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

**RACF administrator response:** Display the PROXY segment of the profile using the RLIST TSO command. Check the LDAPHOST for accuracy, and correct it if needed. If non-anonymous access is required, do the same for the BINDDN and BINDPW.

**Note:** The BINDPW value is not displayed. Respecify it to ensure that it is accurate.

To alter the fields, use the RALTER TSO command. If the profile does not exist, create it using the RDEFINE TSO command. For more information, see *z/OS Security Server RACF Command Language Reference*.

## Messages

---

**IKYL004I      Bad LDAP Server value *server-value* in pkiserv.conf file**

---

**Explanation:** PKI Services LDAP bind processing is trying to retrieve its LDAP bind information in preparation for communicating with the LDAP server. (The *Server1*, *Server2*, and so forth keywords in the **LDAP** section of the PKI Services configuration file specify the server host name information.) The host name has been specified incorrectly. Its value is displayed.

**System action:** PKI Services attempts to bind to your other LDAP servers, if any. If PKI Services is unable to bind to any LDAP servers, the LDAP posting of certificates and CRLs is temporarily suspended. PKI Services attempts to bind again during the next posting interval. All post requests remain in the request database to be attempted later, subject to being deleted after one week of unsuccessful attempts.

**System programmer response:** Locate the server name in the PKI Services configuration file, and correct it if needed. If the host name is specified as a URL, you cannot specify it as an SSL URL (for example, *ldaps://*). PKI Services does not use SSL to communicate with the LDAP server. If you make corrections, stop and restart PKI Services. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYO001I      Error *nnnn* {*setting* | *getting*} *certificate-field* {*in certificate* | *from template*}: *error-code-description***

---

**Explanation:** PKI Services is processing a certificate request field and has encountered an internal error. The field name and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The certificate request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYO002I      *nnnn* bytes of unconsumed data transferring *certificate-field* to certificate**

---

**Explanation:** PKI Services is processing a certificate request field and has found that the field is larger than it should be. This is an internal error. The field name and the number of extra bytes are displayed.

**System action:** The certificate request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYO003I      The certificate request failed validity checks. Status is *nnnn*: *status-code-description***

---

**Explanation:** PKI Services is processing a certificate request field and has encountered an internal error. The field name and the status (error) code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The certificate request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYO004I      *action-being-performed* returned *nnnn*: *error-code-description***

---

**Explanation:** PKI Services is processing a request and has encountered an internal error. The action being performed and the error code encountered are displayed. A description of the error is also displayed, if known.

**System action:** The request is not processed.

**System programmer response:** Report the error to the IBM support center.

---

**IKYP001E      ICSF UNAVAILABLE. CERTIFICATE PROCESSING SUSPENDED**

---

**Explanation:** PKI Services background certificate processing is attempting to create a digital signature. ICSF manages the private key required for digital signing but it is not available for any of the following possible reasons:

- ICSF is inactive or incorrectly configured.
- The user ID of the PKI Services daemon has insufficient authority to use the ICSF private key.
- A system administrator inadvertently deleted the ICSF signing certificate and its private key.

**Routing code:** 2

**Descriptor code:** 6

**System action:** PKI Services background certificate processing is suspended. No certificates or CRLs are issued until the problem is corrected. However, certificate request management functions are still available through the *R\_PKIServ* callable service.

**System programmer response:** Ensure that ICSF and the PCI cryptographic coprocessor (if applicable) are properly configured and operational. Follow the documentation for any issued message with the **CSF** prefix.

If ICH408I messages are issued for insufficient authority to CSFKEYS or CSFSERV class resources, then the user ID of the PKI Services daemon has insufficient authority to use the key. Give the user ID the required access to the specified resource.

To determine if the key you are using requires the PCI cryptographic coprocessor, see Chapter 16, “RACF administration for PKI Services,” on page 241.

For more information, see “Installing and configuring ICSF (optional)” on page 22, *z/OS Cryptographic Services ICSF System Programmer's Guide*, and *z/OS Cryptographic Services ICSF Administrator's Guide*.

If you make changes to ICSF to correct the problem, stop and restart PKI Services. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

#### **IKYP002I      PKI SERVICES INITIALIZATION COMPLETE**

**Explanation:** PKI Services has just been started and has finished initializing.

**System action:** PKI Services processing continues.

**Routing code:** 2

**Descriptor code:** 6

---

#### **IKYP003I      PKI SERVICES SHUTDOWN REQUESTED**

**Explanation:** An operator command was issued to stop PKI Services.

**Routing code:** 2

**Descriptor code:** 5

**System action:** PKI Services stops.

---

#### **IKYP004I      LOG OPTION PROCESSED: *log-option***

**Explanation:** A MODIFY operator command was issued to alter the current log setting for PKI Services.

**System action:** The log setting for PKI Services is changed as requested.

**Routing code:** 2

**Descriptor code:** 5

---

#### **IKYP005I      INCORRECT LOG OPTION SPECIFIED**

**Explanation:** A MODIFY operator command was issued to alter the current log setting for PKI Services. The log parameter syntax or value is incorrect.

**System action:** The MODIFY command is not processed. The log setting for PKI Services is unchanged.

**System programmer response:** Execute the MODIFY command specifying a correct log parameter. For more information, see “Changing logging options” on page 299.

**Routing code:** 2

**Descriptor code:** 5

---

#### **IKYP006I      UNRECOGNIZED PKI SERVICES COMMAND: SPECIFY LOG, DISPLAY, OR STOP**

**Explanation:** A MODIFY operator command was issued for PKI Services. The command specified is not a supported PKI Services command.

**System action:** The MODIFY command is not processed. PKI Services continues processing unchanged.

**System programmer response:** Execute the MODIFY command specifying a supported PKI Services command. For more information, see “Stopping the PKI Services daemon” on page 86 and “Changing logging options” on page 299.

**Routing code:** 2

**Descriptor code:** 5

---

#### **IKYP007E      INSUFFICIENT STORAGE AVAILABLE**

**Explanation:** PKI Services is attempting to allocate storage for processing a MODIFY operator command, but is unsuccessful because of a storage shortage.

**System action:** The console command is not processed. However, PKI Services might continue processing normally.

**Operator response:** Report the problem to your system programmer. After the problem is corrected, you can execute the command again.

**System programmer response:** Increase the region size for the PKI Services started procedure. Stop and restart PKI Services. For more information, see “Steps for starting the PKI Services daemon” on page 85 and “Stopping the PKI Services daemon” on page 86.

**Routing code:** 2

**Descriptor code:** 5

---

#### **IKYP008E      DIRECTORY POST UNSUCCESSFUL. LDAP DATA LIBRARY MODULE RC = *nnnn***

**Explanation:** PKI Services background certificate processing is attempting to post information (certificate, CRL, and so forth) to a directory. The post was unsuccessful. The OCSF Data Library Module (LDAPDL) return code is displayed in the message.

**System action:** The information is not posted at this time. The post request remains in the PKI Services request database to be reattempted later. If posting continues to be unsuccessful for one week, the information is removed from the request database.

**System programmer response:** Determine the cause of the failure from the return code displayed and take appropriate action. These return codes are documented in *z/OS Open Cryptographic Services Facility*.

## Messages

**Application Programming.** If the error is LDAPDL\_NO\_SUCH\_OBJECT, the LDAP entry could not be created because the required suffix does not exist. Check the PKI Services log to determine the entry that could not be created, as indicated on messages IKYC005I and IKYC008I. If the entry should be posted to LDAP, you need to define the suffix in the LDAP server configuration file and recycle the LDAP server. For more information, see “Steps for installing and configuring LDAP” on page 20 and *z/OS Integrated Security Services LDAP Server Administration and Use*.

If you want PKI Services to bypass LDAP posting for certificates with missing suffixes, set RetryMissingSuffix=F in the PKI Services pkiserv.conf configuration file. Then, stop and restart the PKI Services daemon. For more information, see “Steps for tailoring the LDAP section of the configuration file” on page 72.

**Routing code:** 2

**Descriptor code:** 6

---

### IKYP009I PKI SERVICES IS STARTING, FMID product-fmid

**Explanation:** The START operator command was issued to start PKI Services. The START command could have been entered directly at the operator's console or indirectly through a COMMNDxx PARMLIB member.

**System action:** PKI Services initialization proceeds.

**Routing code:** 2

**Descriptor code:** 6

---

### IKYP010I THE CONFIGURATION FILE NAME EXCEEDS THE MAXIMUM LENGTH OF nnnn CHARACTERS

**Explanation:** The PKI Services daemon process is starting. Initialization processing is reading the \_PKISERV\_CONFIG\_PATH environment variable. The value specified is too long.

**System action:** PKI Services stops.

**System programmer response:** Determine the location of your PKI Services environment variables file, and correct the value specified for \_PKISERV\_CONFIG\_PATH. Then, restart PKI Services.

**Routing code:** 2

**Descriptor code:** 6

---

### IKYP011I PKI SERVICES ADDRESS SPACE COULD NOT BE MADE NON-SWAPPABLE: ERROR nnnn

**Explanation:** The PKI Services daemon process is starting. Initialization processing is attempting to make

the PKI Services address space non-swappable. The attempt was unsuccessful. The SYSEVENT TRANSWAP error code is displayed.

**System action:** PKI Services stops.

**System programmer response:** Look up the error code for SYSEVENT TRANSWAP in *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO* to determine what to do. Then, restart PKI Services.

**Routing code:** 2

**Descriptor code:** 6

---

### IKYP012I SYSTEM FUNCTION *function-name* DETECTED ERROR — *error-string*

**Explanation:** PKI Services processing received an error when calling a system service. The service name and error message are displayed.

**System action:** PKI Services stops.

**System programmer response:** See documentation related to the service that failed. Make any necessary corrections. Then, restart PKI Services.

**Routing code:** 2

**Descriptor code:** 6

---

### IKYP013I PKI SERVICES DETECTED AN ERROR DURING INITIALIZATION: ERROR nnnn, REASON 0xnnnn

**Explanation:** PKI Services is starting. Initialization processing is attempting to set up the Program Call (PC) interface. The attempt was unsuccessful. The error and reason codes are displayed.

**System action:** PKI Services stops.

**System programmer response:** Determine the failing service by examining the error code. The values are as follows:

**Note:** This message is also issued after various C function calls. In these cases, ERROR is the value of errno, and REASON is the value of \_\_errno2().

- 1 The PKI Services daemon (IKYPKID) is not APF-authorized.
- 3 Unable to establish recovery. The reason code displayed is the ESTAEX macro return code.
- 5 Unable to create a PC linkage table index. The reason code displayed is the LXRES macro return code.
- 6 Unable to create a PC entry table. The reason code displayed is the ETCRE macro return code.
- 7 Unable to connect the PC entry table to the



linkage table. The reason code displayed is the ETCON macro return code.

#### 8, 10, or 11

Unable to create a name token entry. The reason code displayed is the IEANTCR callable service return code.

For error code 1, make the IKYPKID load module in SYS1.LINKLIB APF-authorized. For all other error codes, see the documentation associated with the MVS service that failed. Make corrections as necessary. Then, restart PKI Services.

**Routing code:** 2

**Descriptor code:** 6

---

#### IKYP014I PKI Services detected an error during termination: Error *nnnn*, Reason *nnnn*

**Explanation:** PKI Services is stopping. Termination processing is attempting to free resources allocated. The attempt was unsuccessful. The error and reason codes are displayed.

**System action:** PKI Services termination processing continues.

**System programmer response:** PKI Services should end normally. If so, no action is needed. However, you might want to diagnose the problem. Determine the failing service by examining the error code:

**16** Unable to establish recovery. The reason code displayed is the ESTAEX macro return code.

See associated documentation for the MVS service that failed. Make corrections as necessary.

---

#### IKYP015I A PKI Services program call request failed: Error *nnnn*

**Explanation:** PKI Services is processing a PC request. The PC request was cancelled before PKI Services completed processing on it. The error code that was posted at the time of the cancel is displayed.

**System action:** PKI Services processing continues.

**System programmer response:** If the error code is 8, no action is required. This is an informational message only. For all other error codes, contact your IBM support center.

---

#### IKYP016I THE PKI SERVICES RUNTIME ENVIRONMENT COULD NOT BE INITIALIZED

**Explanation:** The PKI Services daemon process is starting. Initialization processing is trying to initialize the PKI Services runtime environment within the daemon address space. The attempt was unsuccessful.

**System action:** PKI Services stops.

**System programmer response:** Look for other PKI Services log messages related to this error. For more information, see Chapter 19, "Using information from the PKI Services logs," on page 295.

**Routing code:** 2

**Descriptor code:** 6

---

#### IKYP017I PKI SERVICES IS ALREADY RUNNING

**Explanation:** An attempt was made to start more than one instance of the PKI Services daemon.

**System action:** The first instance of PKI Services continues processing. The second instance stops.

**Routing code:** 2

**Descriptor code:** 6

---

#### IKYP018I PKI Services initialization failed because the program is not APF authorized

**Explanation:** PKI Services is starting. Initialization processing is attempting to initialize the PKI Services runtime environment within the daemon address space. The attempt was unsuccessful because the PKI Services daemon (IKYPKID) is not APF-authorized.

**System action:** PKI Services stops.

**Routing code:** 2

**Descriptor code:** 6

---

#### IKYP019I PKI Services dump created.

**Explanation:** PKI Services encountered a severe error during processing and has dumped the process (using the CEE3DMP callable service).

**System action:** PKI Services processing ends.

**Operator response:** Contact your system programmer.

**System programmer response:** Examine the dump to determine the error. Contact the IBM support center if needed. After the error has been corrected, restart PKI Services. For more information, see "Steps for starting the PKI Services daemon" on page 85 and "Stopping the PKI Services daemon" on page 86.

---

#### IKYP020I PKI SERVICES RESTART REGISTRATION COMPLETE ON *system-name*

**Explanation:** PKI Services is starting. Initialization processing has successfully registered PKI Services for automatic restart (ARM).

**System action:** PKI Services processing continues.

**Routing code:** 2

## Messages

**Descriptor code:** 6

---

**IKYP021I     PKI SERVICES RESTARTING ON**  
*system-name*

**Explanation:** The PKI Services daemon stopped and is being restarted by the Automatic Restart Manager (ARM). The restart was successful.

**System action:** PKI Services processing continues.

**Routing code:** 2

**Descriptor code:** 6

---

**IKYP022I     UNABLE TO REGISTER PKI SERVICES**  
**FOR RESTART: ERROR *nnnn*, REASON**  
**0x*nnnn***

**Explanation:** PKI Services is starting. Initialization processing is attempting to register PKI Services for automatic restart (ARM), using the IXCARM macro service. The attempt was unsuccessful. The IXCARM return and reason codes are displayed. Note: The reason code is displayed in hexadecimal.

**System action:** PKI Services initialization continues without automatic restart capability.

**System programmer response:** Determine and correct the problem with IXCARM as indicated by the error codes displayed. Then, stop and restart PKI Services if automatic restart capability is desired. For more information, see *z/OS MVS Programming: Sysplex Services Reference*.

**Routing code:** 2

**Descriptor code:** 6

---

**IKYP023I     PKI Services failed to format the**  
**display message**

**Explanation:** A MODIFY operator command was issued to display the current settings for PKI Services. Formatting of the display information failed.

**System action:** The settings are not displayed. PKI Services processing continues.

**System programmer response:** Report the error to the IBM support center.

---

**IKYP024I     PKI SERVICES DUMPING FOR ABEND**  
*abend-code RC nnnn*

**Explanation:** PKI Services has incurred an abend. The abend and reason codes are displayed.

**System action:** PKI Services stops.

**System programmer response:** Use IPCS to examine the dump and diagnose the problem. Contact the IBM support center if necessary. Restart PKI Services after the error has been corrected.

**Routing code:** 2

**Descriptor code:** 6

---

**IKYP025I     PKI SERVICES SETTINGS:**

**Explanation:** A MODIFY operator command was issued to display the current settings for PKI Services.

**System action:** The settings are displayed.

```
IKYP025I PKI SERVICES SETTINGS:
|  CA DOMAIN NAME: {CA-domain-name-for-this-PKI-daemon}
|  SUBCOMPONENT      MESSAGE LEVEL
|  LDAP               {current-message-level}
|  SAF                {current-message-level}
|  DB                 {current-message-level}
|  CORE               {current-message-level}
|  PKID               {current-message-level}
|  POLICY              {current-message-level}
|  TPOLICY            {current-message-level}
|  MESSAGE LOGGING SETTING: {STDERR_LOGGING | STDOUT_LOGGING}
|  CONFIGURATION FILE IN USE:
|  {full-UNIX-pathname-of-configuration-file-being-used}
|  TEMPLATE FILE IN USE:
|  {full-UNIX-pathname-of-template-file-being-used}
|  CA CERTIFICATE FINGERPRINTS:
|  SHA1: {EBCDIC-representation-of-sha1-hash}
|  MD5:  {EBCDIC-representation-of-md5-hash}
```

### Restrictions:

- The CA DOMAIN NAME: line is suppressed if the daemon is running in single CA mode.
- For long CA domain names, the *CA-domain-name-for-this-PKI-daemon* value is truncated to 50 characters.

The possible *current-message-level* values for each subcomponent are:

- SEVERE MESSAGES ONLY
- ERROR MESSAGES AND HIGHER
- WARNING MESSAGES AND HIGHER
- INFORMATIONAL MESSAGES AND HIGHER
- DIAGNOSTIC MESSAGES AND HIGHER
- VERBOSE DIAGNOSTIC MESSAGES AND HIGHER

**Operator response:** You can change the subcomponent message levels with the MODIFY operator command if desired. For more information, see “Changing logging options” on page 299.

**Routing code:** 2

**Descriptor code:** 5

---

**IKYP026E     PKI SERVICES {CA | RA}**  
**CERTIFICATE EXPIRES ON *yyyy/mm/dd***

**Explanation:** The certificate that contains the PKI Services CA or RA public key expires on the date shown.

**System action:** If the certificate has not yet expired, processing continues as normal. After the CA certificate expires, certificates issued by PKI Services might be unusable depending on their usage.

**System programmer response:** You should renew

the certificate before it expires. If your security product is RACF, your certificate is contained in a RACF profile established when you first configured PKI Services. Follow RACF documentation on how to renew a certificate. This is done using either the RACDCERT TSO command or RACF ISPF panels. For more information, see “Renewing your PKI Services CA and RA certificates” on page 248 and *z/OS Security Server RACF Security Administrator's Guide*.

**Routing code:** 2

**Descriptor code:** 6

---

#### **IKYP027E ERROR ACCESSING PKI SERVICES CA CERTIFICATE**

**Explanation:** The PKI Services CA certificate is stored in the security product's database. PKI Services background certificate processing is attempting to access the certificate using the R\_data1ib SAF callable service. The attempt failed. Message IKYS015I should also appear in the PKI Services log.

**System action:** PKI Services background certificate processing is suspended. No certificates are issued until the problem is corrected. However, certificate request management functions are still available through the R\_PKIServ callable service.

**System programmer response:** You need to determine why the access failed. Look up the R\_data1ib return code displayed on message IKYS015I in *z/OS Security Server RACF Callable Services*. If your security product is RACF, your certificate is contained in a RACF profile established when you first configured PKI Services. That certificate must be connected as the default certificate to the key ring identified by the KeyRing keyword in the PKI Services configuration file. (The default location for this file is /etc/pkiserv/pkiserv.conf.) If you have only renewed your certificate and have not recycled PKI Services, stopping and restarting the PKI Services daemon might solve the problem. If not, use the RACF RACDCERT LIST and LISTRING commands to determine if the correct certificate is connected to the key ring. Also, use the RACF RLIST command to check that the PKI Services daemon user ID has proper authority to access the profile. Make any required changes. Then, stop and restart PKI Services. For more information, see Chapter 16, “RACF administration for PKI Services,” on page 241 and *z/OS Security Server RACF Security Administrator's Guide*.

**Routing code:** 2

**Descriptor code:** 6

---

#### **IKYP028E PKI SERVICES DISTINGUISHED NAME OR KEY CHANGE ERROR**

**Explanation:** PKI Services is starting. Initialization processing has retrieved the PKI Services signing certificate from the key ring assigned to PKI Services.

The certificate is incompatible with certificate processing that has previously transpired. The subject's distinguished name or the public key or both differ from the previous values used. The subject's distinguished name may not be changed without reconfiguring PKI Services. The public key may be changed, but only if the key rollover process is performed.

**System action:** PKI Services stops.

**System programmer response:** Determine if PKI Services is processing the correct certificate. If your security product is RACF, your certificate is contained in a RACF profile established when you first configured PKI Services. That certificate must be connected as the default certificate to the key ring identified by the KeyRing keyword in the PKI Services configuration file. (The default location for this file is /etc/pkiserv/pkiserv.conf.) Use the RACF RACDCERT LIST and LISTRING commands to determine if the correct certificate is connected to the key ring. If you are attempting to rekey the PKI Services CA, you must follow the rollover process detailed in Chapter 16, “RACF administration for PKI Services,” on page 241. Make any required changes. Then, restart PKI Services. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

**Routing code:** 2

**Descriptor code:** 6

---

#### **IKYP029I PKI Services can only be started from a started procedure**

**Explanation:** An attempt made at starting the PKI Services daemon was rejected because it was not made from a started procedure.

**System action:** The PKI Services daemon halts its initialization and stops after displaying this message to the standard output (STDOUT) of the process.

**System programmer response:** Use the started procedure that PKI Services supplies in SYS1.PROCLIB(PKISERVD). For more information, see “Steps for starting the PKI Services daemon” on page 85.

---

#### **IKYP030I CRL APPROACHING MAXIMUM SIZE**

**Explanation:** PKI Services is creating CRLs as a part of CRL interval processing. CRLs are stored in the request database before being published to the LDAP directory. At least one CRL has been determined to be rather large in size, approaching the VSAM record size limit of 32K bytes. CRL resizing is highly suggested.

**System action:** PKI Services CRL processing continues. If the CRLs are all less than the VSAM size limit, then CRL processing within PKI Services functions normally. However, CRL processing outside of PKI Services may be adversely affected due to the size of the CRL. If any CRL exceeds the VSAM record size

## Messages

limit, PKI Services CRL processing will be unsuccessful. The CRLs in question will not be published to the LDAP directory. When this happens you will also receive message IKYC010I with the error code description Record too long.

**System programmer response:** It is imperative that you correct the situation immediately to prevent the CRL from exceeding the VSAM record size limit. If you are not yet using distribution point CRLs, then you need to start using them now. Edit the PKI Services configuration file and add the CRLDistSize directive to the **CertPolicy** section. If you are already using distribution point CRLs, then you must decrease the value specified for the CRLDistSize directive. Make the appropriate changes. Then, save the configuration file. Once the configuration file has been saved, stop and restart PKI Services. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

**Note:** The action stated above will not result in an immediate reduction in the size of the CRL. You will continue to see this message until the revoked certificates on the CRL expire and are removed from the CRL.

---

### IKYP031E [RSA | DSA] signing key algorithm error

**Explanation:** PKI Services is reading the **CertPolicy** section of its configuration file (pkiserv.conf) to find the signing algorithm. One of the following conditions occurred:

- The CA certificate key type does not match the signature algorithm you specified with the SigAlg1 value in the **CertPolicy** section of the pkiserv.conf configuration file.
- The OID corresponding to the specified algorithm in the **OIDs** section is incorrect or is not specified at all.

**System action:** PKI Services stops.

**System programmer response:** Make sure the SigAlg1 value in the **CertPolicy** section and its corresponding OID value in the **OIDs** section are correct and compatible with the CA certificate's key type.

If the CA certificate key type is RSA, specify the SigAlg1 algorithm value as one of the following:

- sha-1WithRSAEncryption (OID value 1.2.840.113549.1.1.5)
- md-5WithRSAEncryption (OID value 1.2.840.113549.1.1.4)
- md-2WithRSAEncryption (OID value 1.2.840.113549.1.1.2)

If the CA certificate key type is DSA, specify the SigAlg1 algorithm value as follows:

- id-dsa-with-sha1 (OID value 1.2.840.10040.4.3)

Correct the configuration values, and restart PKI Services. For more information, see “Updating the signature algorithm” on page 149.

---

### IKYP032I PKI SERVICES DOES NOT HAVE RA CAPABILITY. SCEP PROCESSING SUSPENDED

**Explanation:** The PKI Services daemon process is starting. Initialization processing determines that the SCEP interface should be enabled and it reads the contents of the key ring to locate the certificate and key to be used for the SCEP registration authority (RA) function. No RA-capable certificate and key was found.

**System action:** Initialization continues but PKI Services SCEP processing is suspended.

**System programmer response:** The RA function of PKI Services requires a certificate and a private key capable of creating general purpose digital signatures and enciphering session keys, with key usage, digitalSignature, and keyEncipherment (Handshaking). Either the PKI Services CA certificate must have this capability (which is atypical) or an additional, dedicated RA certificate for PKI Services must be established. In either case, the certificate must have the proper key usage and must have an RSA private key. If a dedicated RA certificate is used, specify its label using the RALabel directive in the **SAF** section of the PKI Services configuration file (pkiserv.conf). The RA certificate must be assigned to the user ID of the PKI Services daemon and must be connected to the PKI Services key ring with USAGE PERSONAL and DEFAULT NO. For more information, see “Enabling Simple Certificate Enrollment Protocol (SCEP)” on page 175. If you make changes to the PKI Services key ring to correct the problem, stop and restart PKI Services.

**Routing code:** 2

**Descriptor code:** 6

---

### IKYP033I Incorrect value specified for CA domain name

**Explanation:** The PKI Services daemon process is starting. Initialization processing has determined that the CA domain name value specified in the \_PKISERV\_CA\_DOMAIN environment variable in the pkiserv.envars file contains illegal characters.

**System action:** PKI Services stops.

**System programmer response:** The first eight characters of the CA domain name are limited to the following character set: alphanumeric characters (a–z, A–Z, 0–9) and the hyphen (-). In addition, the first character must not be a number or hyphen. Edit the PKI Services environment variables file pkiserv.envars for the CA instance in error and correct the value specified for \_PKISERV\_CA\_DOMAIN. Restart PKI Services. For more information, see “Adding a new CA domain” on page 161.



---

**IKYP034E ICSF UNAVAILABLE. SCEP PROCESSING SUSPENDED**


---

**Explanation:** PKI Services is attempting to decrypt a Simple Certificate Enrollment Protocol (SCEP) request received from a SCEP client or to sign its response. ICSF manages the private key required for SCEP decryption and signing but ICSF is unavailable for any of the following possible reasons:

- ICSF is inactive or incorrectly configured.
- The user ID of the PKI Services daemon has insufficient authority to use the ICSF private key.
- A system administrator inadvertently deleted the certificate and its ICSF private key.

**System action:** PKI Services rejects the SCEP request.

**System programmer response:** Ensure that ICSF and the PCI cryptographic coprocessor (if applicable) are properly configured and operational. Follow the documentation pertaining to any issued messages having the **CSF** prefix. If you make changes to ICSF to correct the problem, stop and restart PKI Services.

If ICH408I messages are issued for insufficient authority to CSFKEYS or CSFSERV class resources, then the user ID of the PKI Services daemon has insufficient authority to use the private key. Give the user ID the required access to the specified resource.

To determine if your key still exists or requires the PCI cryptographic coprocessor, see Chapter 16, “RACF administration for PKI Services,” on page 241. To determine if your SCEP configuration requires a CA certificate or a CA/RA combination, see “Installing and configuring ICSF (optional)” on page 22.

**Routing code:** 2

**Descriptor code:** 6

---

**IKYS001I Error nnnn {attaching | detaching} OCSF-service-provider-description**


---

**Explanation:** PKI Services is attaching or detaching an OCSF or OCEP service provider module. The attach or detach failed. The service provider in error and the error code encountered are displayed.

**System action:** PKI Services stops.

**System programmer response:** Look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*. Diagnose the problem indicated by the return code. Restart PKI Services after corrections are made.

---

**IKYS002I Error nnnn in OCSF-API-name**


---

**Explanation:** PKI Services is calling an OCSF or OCEP API. The invocation has failed. The API name and error code encountered are displayed.

**System action:** If the error occurs during PKI Services initialization, PKI Services stops. Otherwise, PKI Services continues processing. However, needed cryptographic services may not be available.

**System programmer response:** If you are using ICSF for your CA's private key operations and the failing service is either CSP\_CreateSignatureContext or CSSM\_SignData, check that ICSF is functioning and configured properly for PKA operations. For this problem, you will also see console message IKYP001E. Follow the instructions for message IKYP001E. For all other errors, look up the error code in either *z/OS Open Cryptographic Services Facility Application Programming* or *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*. Diagnose the problem indicated by the return code. Restart PKI Services after corrections are made, if needed.

---

**IKYS003I Error nnnn in getting {subject name | public key} from certificate: error-code-description**


---

**Explanation:** PKI Services is retrieving its CA certificate from the SAF key ring. An error occurred while PKI Services was extracting the subject name or public key from the certificate. The error code encountered is displayed. A description of the error is also displayed, if known. This may indicate a problem with the certificate stored in the SAF key ring or it may be an internal error.

**System action:** PKI Services stops.

**System programmer response:** Ensure that the certificate stored in the SAF key ring is correct. If no problems are found, report the error to the IBM support center. For more information, see Chapter 16, “RACF administration for PKI Services,” on page 241 and *z/OS Security Server RACF Security Administrator's Guide*.

---

**IKYS004I Error 0xnnnn in opening key ring key-ring-name**


---

**Explanation:** PKI Services is initializing and is calling System SSL services to open the SAF key ring containing the CA certificate. The open failed. The key ring name and System SSL services error code encountered is displayed.

**System action:** PKI Services stops.

**System programmer response:** Look up the error code in *z/OS Cryptographic Service System Secure Sockets Layer Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

## Messages

---

**IKYS005I      Error 0xnnnn in closing key ring**

---

**Explanation:** PKI Services is terminating and is invoking System SSL services to close the SAF key ring containing the CA certificate. The close failed. The System SSL services error code encountered is displayed.

**System action:** PKI Services continues termination.

**System programmer response:** Look up the error code in *z/OS Cryptographic Service System Secure Sockets Layer Programming*. Diagnose the problem indicated by the return code. Make corrections as indicated. Restart PKI Services if desired.

---

**IKYS006I      Cannot delete the signing context**

---

**Explanation:** PKI Services is attempting to sign a certificate or CRL and is invoking the OCSF API CSSM\_DeleteContext. The invocation failed.

**System action:** The certificate or CRL is not created.

**System programmer response:** Report the error to the IBM support center.

---

**IKYS007I      No KeyRing value specified under SAF section in pkiserv.conf file**

---

**Explanation:** PKI Services is reading its configuration file to locate the value specified for KeyRing in the **SAF** section. The value is missing or has an incorrect syntax.

**System action:** PKI Services stops.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYS008I      Signing key is from unknown crypto service provider**

---

**Explanation:** PKI Services is retrieving its private key from the SAF key ring. The private key type is not known to PKI Services. This may indicate a problem with the certificate and private key stored in the SAF key ring or it may be an internal error.

**System action:** PKI Services stops.

**System programmer response:** Ensure that the certificate and private key stored in the SAF key ring are correct. If no problems are found, report the error to the IBM support center. For more information, see Chapter 16, “RACF administration for PKI Services,” on page 241 and *z/OS Security Server RACF Security Administrator's Guide*.

---

**IKYS009I      Profile for key ring *key-ring-name* not found**

---

**Explanation:** PKI Services is reading its configuration file to locate the value specified for KeyRing in the **SAF** section. The key ring specified is incorrect. No such key ring exists.

**System action:** PKI Services stops.

**System programmer response:** Correct the value and restart PKI Services if desired. For more information, see “(Optional) Steps for updating the configuration file” on page 51.

---

**IKYS010I      Profile for key ring or default certificate or private key not found**

---

**Explanation:** PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.
- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring has no private key.
- User ID assigned to the PKI Services daemon has insufficient authority to read the key ring or private key.

**System action:** PKI Services stops.

**System programmer response:** Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 and *z/OS Security Server RACF Security Administrator's Guide*.

---

**IKYS011I      Error *error-description* in pthread\_rwlock\_rdlock/wrlock**

---

**Explanation:** PKI Services is retrieving its CA certificate from the SAF key ring. An internal error occurred while PKI Services was calling the pthread\_rwlock\_rdlock or pthread\_rwlock\_wrlock UNIX function. A description of the error is displayed.

**System action:** PKI Services stops.

**System programmer response:** Report the error to the IBM support center.

---

**IKYS012I      Error *error-description* in pthread\_rwlock\_unlock**

---

**Explanation:** PKI Services is retrieving its CA certificate from the SAF key ring. An internal error occurred while PKI Services was invoking the pthread\_rwlock\_unlock UNIX function. A description of the error is displayed.

**System action:** PKI Services stops.

**System programmer response:** Report the error to the IBM support center.

---

**IKYS013I Cannot find the private key associated with the {default | RA} certificate**

---

**Explanation:** PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. The problem is related to either the CA (default) certificate or the RA certificate, as indicated in the message. Possible problems are:

- The certificate is not connected to the ring.
- The certificate is incorrectly connected to the key ring. Both must have USAGE PERSONAL and the CA certificate must be the DEFAULT.
- The certificate has no private key.
- The user ID assigned to the PKI Services daemon has insufficient authority to read the key ring or the private key.

**System action:** If the problem is with the default certificate, PKI Services stops. If the problem is with the RA certificate, PKI Services continues but the Simple Certificate Enrollment Protocol (SCEP) is disabled.

**System programmer response:** Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 4, “Running IKYSETUP to perform RACF administration,” on page 27 and *z/OS Security Server RACF Security Administrator's Guide*.

---

**IKYS014I Cannot find the {default | RA} certificate with private key associated in key ring**

---

**Explanation:** PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. The problem is related to either the CA (default) certificate or the RA certificate, as indicated in the message. Possible problems are:

- The certificate is not connected to the ring.
- The certificate is incorrectly connected to the key ring. Both must have USAGE PERSONAL and the CA certificate must be the DEFAULT.
- The user ID assigned to the PKI Services daemon has insufficient authority to read the key ring or the private key.

**System action:** If the problem is with the default certificate, PKI Services stops. If the problem is with the RA certificate, PKI Services continues but the Simple Certificate Enrollment Protocol (SCEP) is disabled.

**System programmer response:** Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see “Locating your PKI Services certificates and key ring” on page 244 and *z/OS*

*Security Server RACF Security Administrator's Guide*.

---

**IKYS015I RACF callable service, R\_datalib, with function code nnnn returns with SAF return code=nnnn, RACF return code=nnnn, RACF reason code=nnnn**

---

**Explanation:** PKI Services is attempting to retrieve data from the SAF key ring specified by the KeyRing value in the **SAF** section of the pkiserv.conf file. The key ring specified does not appear to be set up properly. Possible problems are:

- Key ring is empty.
- CA certificate in the key ring not connected as PERSONAL DEFAULT.
- CA certificate in key ring has no private key.
- User ID assigned to the PKI Services daemon has insufficient authority to read the key ring or private key.

**System action:** PKI Services stops.

**System programmer response:** Look up the return and reason code displayed in *z/OS Security Server RACF Callable Services*. Make corrections as needed. Ensure that the SAF key ring and the certificate stored in it are correct. For more information, see Chapter 16, “RACF administration for PKI Services,” on page 241 and *z/OS Security Server RACF Security Administrator's Guide*.

---

**IKYS016I Error 0xnnnn getting the {default | RA} key**

---

**Explanation:** PKI Services is initializing and calling System SSL services to get one of its private keys from the key ring. The obtain failed. The problem is related to either the CA (default) certificate private key or the RA certificate private key, as indicated in the message.

**System action:** PKI Services stops.

**System programmer response:** Look up the error code in *z/OS Cryptographic Service System Secure Sockets Layer Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

---

**IKYS017I Error 0xnnnn exporting the {default | RA} key**

---

**Explanation:** PKI Services is initializing and calling System SSL services to export one of its private keys from the key ring. The export failed. The problem is related to either the CA (default) certificate private key or the RA certificate private key, as indicated in the message.

**System action:** PKI Services stops.

**System programmer response:** Look up the error code in *z/OS Cryptographic Service System Secure*

## Messages

*Sockets Layer Programming.* Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

---

### IKYS018I      Error 0xnnnn signing Certificate/CRL

**Explanation:** PKI Services is attempting to sign a certificate or CRL. The signing failed.

**System action:** PKI Services continues processing. However, the needed cryptographic services may not be available.

**System programmer response:** If you are using ICSF for your CA's private key operations, check that ICSF is functioning and configured properly for PKA operations. For all other errors, look up the error code in *z/OS Cryptographic Service System Secure Sockets Layer Programming*. Diagnose the problem indicated by the return code. Restart PKI Services once corrections are made.

---

### IKYU001I      Unable to open file *file-pathname* for {READ | WRITE}

**Explanation:** A user is running a PKI Services utility program. The program is unable to open the input file specified. The name of the file is displayed in the message.

**System action:** The program ends.

**User response:** Check that the specified file pathname is correct and that the file exists. Also check the file's permissions to ensure that the user is allowed to process the file. Make changes as necessary and retry the program.

---

### IKYU002I      SAF Service IRRSPX00 Returned SAF RC = nn RACF RC = nn RACF RSN = nn {*diagnostic-information*}

**Explanation:** A user is running a PKI Services utility program. The program encountered an error while calling the R\_PKIServ (IRRSPX00) callable service. The diagnostic information is displayed at the end of the message. For meanings of the diagnostic information, see message IKYI002I.

**System action:** The program ends.

**RACF administrator response:** Determine if the user should be permitted to perform the task. Make RACF authorization changes if necessary. For authorization information, see "Authorizing users for the PKI Services administration group" on page 241.

**System programmer response:** See message IKYI002I for information. For more information, see "Using the pkiprereg utility" on page 309.

**User response:** Correct the problem if applicable. If you cannot correct the problem, contact your Web administrator.

**Web administrator response:** See the Web administrator responses listed for message IKYI002I.

---

### IKYU003I      Unknown field name found in SCEP data file, *error-field-name*

**Explanation:** A user is running the pkiprereg PKI Services utility program. The program is reading the SCEP data file and has encountered a field name that it does not recognize. The erroneous field name is displayed.

**System action:** The program continues. If the user specified the -I option, the client is not preregistered.

**User response:** Correct the data file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 309.

---

### IKYU004I      Required field name {"ClientName" | "Template"} missing from input file at line *nnn*

**Explanation:** A user is running the pkiprereg PKI Services utility program. The program is reading the SCEP data file and encountered a preregistration entry that is missing a required field name. The missing field name is displayed. The line number where the error was found is also displayed.

**System action:** The program continues. If the user specified the -I or -r option of the pkiprereg utility, the current record is not processed.

**User response:** Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 309.

---

### IKYU005I      Duplicate ClientName=*error-value* entry found in SCEP data file at line *nnn*

**Explanation:** A user is running the pkiprereg PKI Services utility program. The program is reading the SCEP data file and encountered a preregistration entry with a ClientName value that is already in use. The value was either used earlier in the file or is already registered in PKI Services. The ClientName value and the line number where the error was found are displayed.

**System action:** The program continues. If the user specified the -I option of pkiprereg utility, the client is not preregistered.

**User response:** Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see "Using the pkiprereg utility" on page 309.



---

**IKYU006I Preregistration record for ClientName=*error-value* not found in PKI Services**


---

**Explanation:** A user is running the pkiprereg PKI Services utility program with the -r (remove) option. The program is reading the SCEP data file and encountered a preregistration entry with a ClientName value that does not exist in PKI Services. The erroneous field name is displayed.

**System action:** The program continues. The current record is not processed.

**User response:** Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see “Using the pkiprereg utility” on page 309.

---

**IKYU007I Incorrect field-name=*error-value* entry found in SCEP data file**


---

**Explanation:** A user is running the pkiprereg PKI Services utility program. The program is reading the SCEP data file and encountered a field name that has an incorrect value. The field name and error value are displayed.

**System action:** The program continues. If the user specified the -I option of pkiprereg utility, the current entry is not processed.

**User response:** Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see “Using the pkiprereg utility” on page 309.

---

**IKYU008I Template nickname *template* not found in certificate templates file**


---

**Explanation:** A user is running the pkiprereg PKI Services utility program. The program is reading the SCEP data file and encountered a encountered a value pair of *template=nickname* where the nickname is not found in the certificate templates file.

**System action:** The program continues. If the user specified the -I option of pkiprereg utility, the client is not preregistered.

**User response:** Correct the input file. Remove any entry that should not be reprocessed and rerun the program if desired. For more information, see “Using the pkiprereg utility” on page 309.

---

**IKYU009I Out of memory**


---

**Explanation:** A user is running a PKI Services utility program. The program is unable to allocate more memory.

**System action:** The program ends.

**User response:** Increase the amount of memory

available to the program if possible. Otherwise report the error to your system programmer.

**System programmer response:** Increase the amount of REGION available to the user or adjust the Language Environment runtime memory settings. (For instructions, see *z/OS Language Environment Programming Guide*.)

---

**IKYU010I Internal error occurred in function *function-name: diagnostic-information***


---

**Explanation:** A user is running a PKI Services utility program. The program has encountered an internal error. The function in error is displayed. Additional diagnostic information may also be displayed.

**System action:** The program ends.

**User response:** Report the error to the IBM support center.

---

**IKYU011I System function *function-name* detected error -- *error-string***


---

**Explanation:** A user is running a PKI Services utility program. The program received an error calling a system service. The service name and error message are displayed.

**System action:** The program ends.

**User response:** See documentation related to the service that failed. Make any necessary corrections. Then, rerun the program.

---

**IKYU012I Unable to open message catalog *message-catalog-filename* -- *error-string***


---

**Explanation:** A user is running a PKI Services utility program. The program is attempting to open an external message catalog using the default location for the message catalog as set by the NLSPATH environment variable.

**System action:** The program continues using default messages.

**User response:** If you require an external message catalog, correct the indicated error. Then, rerun the program. For system errors, report the problem to your system programmer. For more information, see Chapter 20, “Using PKI Services utilities,” on page 301.

**System programmer response:** If the user requires an external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access.

For information about updating the NLSPATH environment variable, see *z/OS UNIX System Services Programming Tools*. If default messages are acceptable, no action is necessary.

---

## Messages

---

| **IKYU013I**     **Incorrect command syntax. The**  
|                    *error-value* **parameter is**  
|                    **{unknown | missing | incorrectly**  
|                    **specified}**  
|                    **Usage:** *command-usage*

| **Explanation:** A user is running a PKI Services utility  
| program. A command parameter was entered  
| incorrectly. The correct command usage is displayed.

| **System action:** The program ends.

| **User response:** Execute the command with the  
| correct syntax. For more information, see Chapter 20,  
| "Using PKI Services utilities," on page 301.

---

| **IKYU014I**     **pkiprereg complete. *nnn* preregistration**  
|                    **records processed. *mmm* successful.**  
|                    ***000* errors found**

| **Explanation:** A user is running the pkiprereg PKI  
| Services utility program. The program completed and is  
| reporting the results. The value of *nnn* is the total  
| number of preregistration records (entry groups) found  
| in the SCEP data file. The value of *mmm* is the number  
| of preregistration records that were successfully  
| processed. The value of *000* is the number of errors  
| found. Because multiple errors can occur for each  
| preregistration record, *mmm* plus *000* may exceed *nnn*.

| **System action:** The program ends.

| **User response:** If no errors were found, no action is  
| required. If errors were found, correct the errors in the  
| SCEP data file. (You may remove the entries that were  
| successful.) Rerun the program with the corrected  
| SCEP data file. For more information, see Chapter 20,  
| "Using PKI Services utilities," on page 301.

---

| **IKYU015I**     **Template with nickname *template***  
|                    **cannot be used for preregistration**

| **Explanation:** A user is running the pkiprereg PKI  
| Services utility program. The program is reading the  
| certificates template file and found the certificate  
| template with the specified nickname. The template  
| either has no **<PREREGISTER>** section or the  
| **<PREREGISTER>** section is not properly terminated.

| **System action:** The program continues. If the **-I** option  
| was specified, the client is not preregistered.

| **System programmer response:** Correct the  
| certificate template. Edit the SCEP data file to remove  
| any entry that should not be reprocessed and rerun the  
| program if desired. For more information, see "Using the  
| pkiprereg utility" on page 309.

---

| **IKYU016I**     **pkiprereg complete. *nnn* passwords**  
|                    **generated**

| **Explanation:** A user is running the pkiprereg PKI  
| Services utility program in **generate** mode (password  
| generate mode). The program completes and reports

| the results. *nnn* is the total number of passwords  
| generated.

| **System action:** The program ends.

| **User response:** No action is required.

---

| **IKYU017I**     ***program-name* terminated abnormally**

| **Explanation:** A user is running a PKI Services utility  
| program. The program can not complete due to a user  
| or system error. The name of the utility program is  
| displayed in the message.

| **System action:** The program ends.

| **User response:** Check for previously issued  
| messages and their associated actions. Make changes  
| as necessary and retry the program.

# Chapter 22. File directory structure

This chapter discusses the location of files in:

- z/OS product libraries
- File system directory /usr/lpp/pkiserv/ and its subdirectories.

## Product libraries

SMP/E installs PKI Services into the following product libraries:

- SAMPLIB/ASAMPLIB
  - IKYCVSAM
  - IKYMVSAM
  - IKYRVSAM
  - IKYSETUP
  - IKYISMKD
  - IKYMKDIR
  - IKYALLOC
  - IKYDDDEF
- PROCLIB/APROCLIB
  - IKYSPROC with alias PKISERVD
- SIEALNKE/AIEALNKE
  - IKYPKID - The PKI Services daemon
  - IKYPRTM - The Resource Termination Manager for the daemon

## File system directory and subdirectories

Additionally, unless you change the default, SMP/E installs PKI Services into the file system directory /usr/lpp/pkiserv. The following table describes the directory structure and contents:

Table 75. Files contained in subdirectories

Subdirectory	Contains...
bin	Utilities executables: <ul style="list-style-type: none"><li>• iclview— Utility for viewing issued certificate list (certificate database). (For more information, see “Using the iclview utility” on page 302.)</li><li>• pkiprereg—Utility for creating preregistration records in batch for Simple Certificate Enrollment Protocol (SCEP) clients, see “Using the pkiprereg utility” on page 309.)</li><li>• pkitp_install—Program to register the PKI Services Trust Policy plug-in with OCSF. (For more information, see “Configuring and getting started with PKITP” on page 269.)</li><li>• pkitp_ivp—Program to verify that the PKI Services Trust Policy plug-in installed successfully. (For more information, see “Configuring and getting started with PKITP” on page 269.)</li><li>• vosview—Utility for viewing VSAM object store (request database). (For more information, see “Using the vosview utility” on page 305.)</li></ul>
include	C header files: <ul style="list-style-type: none"><li>• pkitp.h—C language header file for writing application programs that use the PKI Trust Policy Plug-in. (For more information, see “Files for PKITP” on page 269.)</li></ul>

## File directory structure

Table 75. Files contained in subdirectories (continued)

lib	<p>Loadable files:</p> <ul style="list-style-type: none"> <li>• <code>pkitp.so</code> — OCSF Trust Policy plug-in for PKI Services. (For more information, see “Files for PKITP” on page 269.)</li> <li>• <code>*.dll</code> — Dynamic link libraries (DLLs) that the PKI Services daemon uses.</li> <li>• <code>nls/msg/En_US.IBM-1047/*.cat</code>—The PKI Services message catalogs. (These message catalogs are also symbolically linked in the <code>/usr/lpp/pkiserv/lib/nls/msg/C</code> directory as well as the <code>/usr/lib/nls/msg/En_US.IBM-1047</code> and <code>/usr/lib/nls/msg/C</code> directories.)</li> </ul>
PKIServ	<p>CGIs that make up the PKIServ Web application. (For information about CGIs, see “Relationship between CGIs and the <code>pkiserv.tmpl</code> file” on page 123 and Table 40 on page 141.)</p> <p>PKIServ contains the following subdirectories:</p> <ul style="list-style-type: none"> <li>• <code>public-cgi</code>—Public (non-SSL) directory</li> <li>• <code>ssi-cgi-bin</code>—SSL-protected <ul style="list-style-type: none"> <li>– <code>auth</code>—SSL with user ID and password protection. Work runs under client’s ID.</li> <li>– <code>surrogateauth</code>—SSL with user ID and password protection. Work runs under surrogate ID (PKISERV).</li> </ul> </li> <li>• <code>clientauth-cgi-bin</code>—SSL with client certificate protection. Work runs under surrogate ID (PKISERV). <ul style="list-style-type: none"> <li>– <code>auth</code>—SSL with client certificate protection. Work runs under administrator’s ID.</li> </ul> </li> </ul>
samples	<p>Various sample files, including:</p> <ul style="list-style-type: none"> <li>• <code>expiringmsg.form</code>—The e-mail message sent to a user as notification about an certificate that will expire</li> <li>• <code>httpd.conf</code>—Contains z/OS HTTP Server directives. (For a code sample, see “z/OS HTTP Server configuration directives” on page 371.)</li> <li>• <code>httpd2.conf</code>—Contains z/OS HTTP Server directives for the second web server. (For a code sample, see “z/OS HTTP Server configuration directives” on page 371.)</li> <li>• <code>httpd.envvars</code>—A sample of the environment variables needed for PKI Services that you should <i>integrate</i> into your existing z/OS HTTP Server environment variables file (<code>httpd.envvars</code>). (For a code sample, see “The <code>pkiserv.envvars</code> environment variables file” on page 349.)</li> <li>• <code>Makefile.pkiexit</code>—The makefile for the PKI Services exit. (For more information, see “Steps for updating the exit code sample” on page 181.)</li> <li>• <code>Makefile.pkitsamp</code>—The makefile for <code>pkitsamp.c</code>, which is a sample application to call the PKI Trust Policy plug-in. (For more information, see “Files for PKITP” on page 269.)</li> <li>• <code>pkiexit.c</code>—The sample PKI Services exit, which PKI Services provides. (For more information, see “Steps for updating the exit code sample” on page 181.)</li> <li>• <code>pkiserv.envvars</code>—The PKI Services environment variables file. (For more information, see “Optionally updating PKI Services environment variables” on page 47 and “The <code>pkiserv.envvars</code> environment variables file” on page 349.)</li> <li>• <code>pkiserv.tmpl</code>—The PKI Services certificate templates file. (For more information, see Chapter 11, “Customizing the end-user Web application,” on page 91.)</li> <li>• <code>pkiserv.conf</code>—The PKI Services configuration file. (For more information, see “(Optional) Steps for updating the configuration file” on page 51 and Chapter 23, “The <code>pkiserv.conf</code> configuration file,” on page 343.)</li> <li>• <code>pkitsamp.c</code>—Sample application to call the PKI Trust Policy plug-in. (For more information, see “Files for PKITP” on page 269 and “Building the sample application to invoke the certificate validation service” on page 274.)</li> <li>• <code>readymsg.form</code>—The e-mail message sent to a user as notification a certificate is ready for retrieval</li> <li>• <code>rejectmsg.form</code>—The e-mail message sent to a user as notification a request for a certificate has been rejected</li> </ul>

## Chapter 23. The pkiserv.conf configuration file

This chapter includes a code sample of the pkiserv.conf configuration file.

The pkiserv.conf file is the configuration file for the PKI Services daemon. By default, you can find this file in the /usr/lpp/pkiserv/samples/ directory. For more information about the sections of the pkiserv.conf configuration file and the parameters, see “(Optional) Steps for updating the configuration file” on page 51 and Table 19 on page 51.

The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the pkiserv.conf file in the source directory /usr/lpp/pkiserv/samples/.

```
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2001, 2006
# Status = HKY7730

[OIDs]
#
# Supported Distinguished Name OIDs
#
C=2.5.4.6
O=2.5.4.10
OU=2.5.4.11
CN=2.5.4.3
L=2.5.4.7
ST=2.5.4.8
TITLE=2.5.4.12
POSTALCODE=2.5.4.17
STREET=2.5.4.9
MAIL=0.9.2342.19200300.100.1.3
EMAIL=1.2.840.113549.1.9.1
SERIALNUMBER=2.5.4.5
UNSTRUCTUREDNAME=1.2.840.113549.1.9.2
UNSTRUCTUREDADDRESS=1.2.840.113549.1.9.8

#
# Signature Algorithm OIDs
#
sha-1WithRSAEncryption=1.2.840.113549.1.1.5
md-5WithRSAEncryption=1.2.840.113549.1.1.4
md-2WithRSAEncryption=1.2.840.113549.1.1.2
id-dsa-with-sha1=1.2.840.10040.4.3

MyPolicy=1.2.3.4

[ObjectStore]
# Data set name of the VSAM request (object store) base CLUSTER
ObjectDSN='pkisrvd.vsam.ost'

# Data set name of the VSAM object store PATH for the transaction ID (TID)
# alternate index.
ObjectTidDSN='pkisrvd.vsam.ost.path'

# Data set name of the VSAM object store PATH for the status alternate index
ObjectStatusDSN='pkisrvd.vsam.ost.status'

# Data set name of the VSAM object store PATH for the requestor alternate index
ObjectRequestorDSN='pkisrvd.vsam.ost.requestor'

# Data set name of the VSAM issued certificate list (ICL) base CLUSTER
ICLDSN='pkisrvd.vsam.icl'

# Data set name of the VSAM ICL PATH for the status alternate index
```

```

ICLStatusDSN='pkisrvd.vsam.icl.status'

# Data set name of the VSAM ICL PATH for the requestor alternate index
ICLRequestorDSN='pkisrvd.vsam.icl.requester'

# How days (d) or weeks (w) should completed requests remain in the object store?
# Specify 0d to indicate completed requests should not be removed
RemoveCompletedReqs=1w

# How days (d) or weeks (w) should inactive requests remain in the object store?
# Specify 0d to indicate inactive requests should not be removed
RemoveInactiveReqs=4w

# How many days (d) or weeks (w) should expired certificates remain in the ICL?
# Specify 0d to indicate expired certificates should not be removed
#RemoveExpiredCerts=26w

# Are the VSAM data sets shared in a sysplex with other instances
# of PKI Services. True (T) or False (F)
SharedVSAM=F

[CertPolicy]
SigAlg1=sha-1WithRSAEncryption
CreateInterval=3m

# When the warning message should be issued. (i.e. the number of days
# or weeks before the certificate expiration date/time). Defaults to never
ExpireWarningTime=4w

TimeBetweenCRLs=1d
CRLDuration=2d

# Maximum number of certificates that may appear on one distribution point CRL.
# The default is 0 which indicates distribution point CRLs should not be created.
CRLDistSize=500

# Constant portion of the CRL distribution point leaf-node relative
# distinguished name. The distribution point number is appended to this value
# to form the common name. The default value is "CRL".
CRLDistName=CRL

# Authority Revocation List(ARL) Distribution Point. 'F' (default) indicates
# no ARL distribution point will be created. 'T' indicates ARL distribution
# point will be created IF CRLDistSize is greater than zero.
ARLDist=F

# Full path of the HFS Directory where PKI Services is to save each
# distribution point CRL specified by the CRL distribution point extension
# "Uniform Resource Identifier" fields(URI) for the http protocols. Defaults
# to /var/pkiserv/. It will be ignored if you do not create the extension
# for the http protocol.
CRLDistDirPath=/var/pkiserv/

# Values for the CRL distribution point extension URI fields for the
# protocols(ldap, http) you choose. This is repeatable. The first one
# always starts with CRLDistURI1, followed by CRLDistURI2, 3, ...n,
# if necessary. Uncomment and update the desired directive to enable
# URI CRL distribution point that you need. If more than one URI field
# is needed, remember to increase the field number sequentially by the
# order of one, e.g. CRLDistURI2, CRLDistURI3...

# For ldap protocol, you may specify the LDAP server indicated in the LDAP
# section below, e.g.,
#CRLDistURI1=LdapServer1

# or specify a skeleton URL which contains the protocol type, the domain
# name and the port, if needed, e.g.,
#CRLDistURI1=ldap://myotherldapserver.mycompany.com:389/

# For http protocol, specify the complete URL minus the file name of the
# distribution point CRL file, e.g.,
#CRLDistURI1=http://www.mycompany.com/PKIServ/cacerts/

```

```

# What type of OSCP request is desired?
# 'none' - No OSCP responder support (This is the default)
# or
# 'basic' - the signature in the request(if there is one) will be ignored
OCSPType=none

#
# Enable the Simple Certificate Enrollment Protocol (SCEP)
# T = True, SCEP is enabled
# F = False, SCEP is disabled (default if not specified)
#
EnableSCEP=F

PolicyRequired=F
PolicyCritical=F

PolicyName1=MyPolicy
Policy1Org=MyOrganization
Policy1Notice1=3
Policy1Notice2=17
UserNoticeText1=This is some very lawyerly statement for the relying party
to read and make decisions based on.
CPS1=http://www.mycompany.com/cps.html

# Length of certificate suspension grace period in day or weeks (d,w).
# Certificates which remained suspended for longer than this period are
# automatically revoked.
# The default value is 0d which indicates the grace period is unlimited.
MaxSuspendDuration=120d

[General]
InitialThreadCount=10

# full pathname or data set name containing the 'your certificate is ready'
# message form. Defaults to no message issued
ReadyMessageForm=/etc/pkiserv/readymsg.form

# full pathname or data set name containing the 'your certificate request
# has been rejected' message form. Defaults to no message issued
RejectMessageForm=/etc/pkiserv/rejectmsg.form

# full pathname or data set name containing the 'your certificate is about
# to expire' message form. Defaults to no message issued
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form

[SAF]
KeyRing=PKISRVD/CARing

# The Label name for the PKI RA certificate connected to the Key ring
# specified in the KeyRing value above
#
RALabel=Local PKI RA

[LDAP]
NumServers=1
PostInterval=5m
Server1=myldapserver.mycompany.com:389
AuthName1=CN=root
AuthPwd1=root
CreateOUValue= Created by PKI Services
RetryMissingSuffix=T
# Name of the LDAPBIND Class profile containing the bind information for LDAP
# server 1. This key is optional. Used in place of keys Server1, AuthName1.
# and AuthPwd1
#BindProfile1=LOCALPKI.BINDINFO.LDAP1

```

**pkiserv.conf**



---

## Chapter 24. Environment variables

This chapter describes the environment variables that PKI Services uses and their possible values. It also includes a code sample of the environment variables file, `pkiserv.envvars`. (See “The `pkiserv.envvars` environment variables file” on page 349.) For information about PKISERVD procedure, which specifies the pathname of the environment variables file, see “PKISERVD sample procedure to start PKI Services daemon” on page 384.

---

### Environment variables in the environment variables file

The environment variables contained in `pkiserv.envvars` and their values are:

#### `_PKISERV_CA_DOMAIN`

Specifies the CA domain. The first eight characters must be unique. The first eight characters of the CA domain name are limited to the following character set: alphanumeric characters (a–z, A–Z, 0–9) and the hyphen (-). In addition, the first character must not be a number or hyphen.

#### **Example:**

```
_PKISERV_CA_DOMAIN=WebAppCA
```

#### `_PKISERV_CONFIG_PATH`

Specifies the pathname for the directory containing the configuration file, `pkiserv.conf`, and the certificate template file, `pkiserv.tmpl` for this CA domain. The default value (if you do not set the environment variable) is `/etc/pkiserv`.

**Guideline:** Copy both of these files from the install directory, `/usr/lpp/pkiserv/samples`, before making any changes.

**Note:** Because the PKISERV CGI scripts run in a z/OS HTTP Server address space, if the `pkiserv.tmpl` is not in its default location of `/etc/pkiserv/pkiserv.tmpl`, you need to add the `_PKISERV_CONFIG_PATH` variable to the z/OS HTTP Server environment variable file. The HTTP servers environment variables file is usually in `/etc/httpd.envvars`. PKI Services uses two instances of the z/OS HTTP Server. Therefore, if the two servers are using different environment variables files, you need to update both files.

#### `_PKISERV_EXIT`

Specifies the full pathname for the installation-provided PKI exit program that the PKI Services Web page interface calls. (This exit is a UNIX-executable program or shell script.) If you do not define this variable or if it contains a null value, the PKI exit processing is disabled.

**Note:** The PKI Services CGI scripts run in a z/OS HTTP Server address space, so you must specify the `_PKISERV_EXIT` environment variable in the z/OS HTTP Server environment variables file. The z/OS HTTP Server environment variables file is usually `/etc/httpd.envvars`. PKI Services uses two instances of the z/OS HTTP Server. Therefore, if the two servers are using different environment variables files, you need to update both files.

## Environment variables

### \_PKISERV\_MSG\_LOGGING

Values include:

#### STDOUT\_LOGGING

Indicates writing *all* messages (verbose, diagnostic, informational, warning, error, and severe) to STDOUT and *additionally* writing the error and severe messages to STDERR. This is the default if the environment variable is not set.

#### STDERR\_LOGGING

Indicates writing verbose, diagnostic, informational, and warning messages to STDOUT and writing error and severe messages to STDERR.

### \_PKISERV\_MSG\_LEVEL

Specifies the subcomponent and message level to log. Messages for a particular subcomponent are logged only if the message level is greater than or equal to the specified level for that subcomponent. You can use an asterisk (\*) to indicate all subcomponents. The subcomponent list consists of a subcomponent name and a message level separated by a period (.).

For example, the following sets the message level for all subcomponents to log warning messages or higher. (This is the default setting.)

#### **Example:**

```
_PKISERV_MSG_LEVEL=*.W
```

You can specify multiple subcomponents by separating entries with a comma (.). For example, the following indicates that all subcomponents are set to message level **W** (warning) and that the PKID subcomponent is set to message level **D** (diagnostic).

#### **Example:**

```
_PKISERV_MSG_LEVEL=*.W,PKID.D
```

The subcomponents are:

Subcomponent	Meaning
*	The wildcard character (represents all subcomponents)
CORE	The core functions of PKI Services that are not specific to the other subcomponents
DB	Activity related to the request or issued certificate VSAM data stores
LDAP	LDAP posting operations
PKID	The PKI Services daemon address setup and infrastructure
POLICY	Certificate creation and revocation policy processing
SAF	SAF key ring, OCEP, and R_data1ib calls
TPOLICY	Trust policy plug-in processing

The message levels are listed hierarchically:

Debug level	Meaning
<b>S</b>	This indicates logging only severe messages.
<b>E</b>	This indicates logging severe and error messages.
<b>W</b>	This indicates logging severe, error, and warning messages. This is the <i>default</i> message level for all subcomponents if you do not set the environment variable.
<b>I</b>	This indicates logging severe, error, warning, and informational messages.
<b>D</b>	This indicates logging severe, error, warning, informational, and diagnostic messages.
<b>V</b>	This indicates logging <i>all</i> messages, including verbose diagnostic messages. This is very verbose.

**Guideline:** Do not use **V** level unless IBM support personnel instruct you to do so.

## The pkiserv.envars environment variables file

The following code sample is for the pkiserv.envars environment variables file. (For information about updating the environment variables file, see “Optionally updating PKI Services environment variables” on page 47.) The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the pkiserv.envars file in the source directory /usr/lpp/pkiserv/samples/.

```
#-----#
#
# PKI Services sample environment variable file
#
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2001, 2006
# Status = HKY7730
#
#-----#
#
# Language and Path configurations
#
LANG=en_US.UTF-8
PATH=/usr/sbin
LIBPATH=/usr/lpp/pkiserv/lib:/usr/lib
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/pkiserv/lib/nls/msg/%L/%N

#
# When running as a CA Domain, set the CA Domain name by assigning
# desired value to the _PKISERV_CA_DOMAIN variable.
# Note: The first eight characters must be unique.
#
# example: _PKISERV_CA_DOMAIN=WebAppCA

#
# Configuration File location and Message configuration Options
#
_PKISERV_CONFIG_PATH=/etc/pkiserv
_PKISERV_MSG_LOGGING=stdout_logging
_PKISERV_MSG_LEVEL=*.w
```

## Environment variables

```
#  
# Location of the OCSF Registry (/var/ocsf is the default location)  
#  
OCSFREGDIR=/var/ocsf
```

---

## Chapter 25. The IKYSETUP REXX exec

IKYSETUP is a REXX exec that issues RACF commands to perform RACF administration. This chapter describes the actions IKYSETUP performs and provides a code sample of IKYSETUP.

---

### Actions IKYSETUP performs by issuing RACF commands

In broad terms, the actions that IKYSETUP performs are as follows:

- Sets up the PKI Services daemon user ID
- Sets up the access control to protect PKI Services
  - Protects end-user functions
  - Protects administrative functions
- Defines one or more CA domains with associated administrative domains
- Creates the CA and RA certificates, their private keys, and key ring
- Creates the z/OS HTTP Server certificate, private key, and key ring
- Enables surrogate operation for the z/OS HTTP Server
- Enables the PKI Services daemon to call OCSF functions

### Setting up the PKI Services daemon user ID

Create the daemon user ID (by default, PKISRV) using the RACF ADDUSER TSO command. Give it an OMVS segment because it needs access to z/OS UNIX. This user ID also needs update access to the VSAM data sets identified in the **ObjectStore** section of the pkiserv.conf file. If necessary, use the RACF ADDSD and PERMIT TSO commands to give this user ID UPDATE access to the VSAM data sets.

**Guideline:** Define the daemon user ID with the NOPASSWORD attribute.

To associate this user ID to the PKI Services started procedure, use the following RACF TSO commands:

```
RDEFINE STARTED PKISRV.* STDATA(USER(PKISRV))
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS RACLIST(STARTED) REFRESH
```

### Setting up access control to protect PKI Services

This task can be divided into two steps:

1. Protecting end-user functions
2. Protecting administrative functions.

#### Protecting end-user functions

You must first determine who your end-users are and how they will be using their certificates. In general there are two categories of end-users:

- Internal clients, such as employees who have SAF user IDs on the host system and who may be using their certificates to access resources on the host
- External clients, who have no access to the host system.

When PKI Services is called, the unit of work has some identity (user ID) associated with it. For external customers, a surrogate user ID is necessary.

**Guideline:** Although under certain circumstances it may be beneficial for internal clients to access PKI Services under their own identities, your implementation will be simpler if you use surrogate user IDs for internal clients as well.

Use the RACF ADDUSER command to create the surrogate user ID (PKISERV). Give it an OMVS segment because it needs access to z/OS UNIX. **Guideline:** Define the surrogate user ID with the PROTECTED and RESTRICTED attributes.

The R\_PKIServ SAF callable service is protected by FACILITY class resources of the form IRR.RPKISERV.*function*[.*ca\_domain*], where *function* is one of the following and *ca\_domain* specifies an optional CA domain name. (Specify *ca\_domain* when your installation has established multiple PKI Services CAs.)

The R\_PKIServ functions are:

<b>EXPORT</b>	Retrieves (exports) a previously requested certificate, or retrieves (exports) the PKI Services registration authority (RA) certificate or the certificate authority (CA) certificate.
<b>GENCERT</b>	Generates an auto-approved certificate.
<b>GENRENEW</b>	Generates an auto-approved renewal certificate. (The request submitted is automatically approved.)
<b>REQCERT</b>	Requests a certificate that an administrator must approve before it is created.
<b>REQRENEW</b>	Requests certificate renewal. The administrator needs to approve the request before the certificate is renewed.
<b>RESPOND</b>	Invokes the PKI OCSP responder.
<b>REVOKE</b>	Revokes a certificate that was previously issued.
<b>SCEPREQ</b>	Generates a certificate request using Simple Certificate Enrollment Protocol (SCEP).
<b>VERIFY</b>	Confirms that a given user certificate was issued by this certificate authority and, if so, returns the certificate fields.

Create these resources and give the PKISERV user ID either READ or CONTROL access to them. CONTROL bypasses subsequent resource checks.

Additional FACILITY class resources of the form IRR.DIGTCERT.*function* protect the actual certificate generation and retrieval functions. If subsequent resource checks are not being bypassed, define these resources and their access.

There are two ways to handle certificate approval:

- An administrator can review certificate requests
- Requests can be auto-approved without administrator action (this should probably be reserved for internal clients only).

If you plan to have an administrator approve certificate requests before issuing certificates, PKISERV needs the following access:

Table 76. Access required if you plan to have an administrator approve certificate requests

Resource	Access
IRR.DIGTCERT.REQCERT	READ
IRR.DIGTCERT.REQRENEW	READ

If your clients can request certificates that are auto-approved without action by an administrator, PKISERV needs the following access:

Table 77. Access required if you plan to use auto-approval

Resource	Access
IRR.DIGTCERT.ADD	UPDATE
IRR.DIGTCERT.GENCERT	CONTROL
IRR.DIGTCERT.GENRENEW	READ

Finally, because the Web server will be switching identities to PKISERV, you must give it surrogate permission. This is done by creating another resource in the SURROGAT class (BPX.SRV.PKISERV) and giving the Web server daemon user ID READ access to it.

### Protecting administrative functions

This is much easier to set up than protecting the end-user functions. Your PKI Services administrators must have SAF user IDs on the host system. When PKI Services is called for administrative functions, the unit of work is always tagged with the identity of the authenticated administrator. Each administrator needs the following FACILITY class resource access to:

Table 78. FACILITY class access needed for protecting administrative functions

Resource	Access	Purpose
IRR.RPKISERV.PKIADMIN[.ca_domain]	READ	For list and query operations
	UPDATE	To act on certificate requests, preregistration requests, and issued certificates

To grant user ID ADMID authority to administer PKI Services, use the following RACF TSO commands:

#### Example:

```
RDEFINE FACILITY (IRR.RPKISERV.PKIADMIN.CUSTOMER) UACC(NONE)
PERMIT IRR.RPKISERV.PKIADMIN.CUSTOMER CLASS(FACILITY) ACCESS(UPDATE) ID(ADMID)
SETROPTS RACLIST (FACILITY) REFRESH
```

## Establishing your CA and RA certificates

To create and sign digital certificates for others, you need to establish a CA certificate, and optional RA certificate, and their associated private keys using the RACDCERT command.

### Steps for establishing your CA and RA certificates

Perform the following steps to create your CA certificate, RA certificate, and their associated keys, back up the keys, connect them to a key ring and authorize PKI Services to use them.

**Before you begin:** Determine the CA or RA's distinguished name and where it will be located (under CERTAUTH for the CA and under the PKI Services daemon user ID for the RA). Typically, CAs and RAs have distinguished names in the following form:

OU=*your-CA-or-RA's-friendly-name*.O=*your-organization*.C=*your-two-letter-country-abbreviation*

1. Create your CA certificate, and optional RA certificate, and their associated private keys using the RACDCERT GENCERT command. **Rule:** If you create an optional RA certificate, it must be signed by the CA certificate.

**Example:**

- a. This example creates a 20-year CERTAUTH certificate with a distinguished name of OU=Human Resources Certificate Authority.O=Your Company, Inc.C=US.

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(
  OU('Human Resources Certificate Authority')
  O('Your Company, Inc') C('US')) WITHLABEL('Local PKI CA') HIGHTRUST
  NOTAFTER(DATE(2026/05/06))
  SIZE(1024) KEYUSAGE(HANDSHAKE)
  SIGNWITH(CERTAUTH LABEL('Local Root CA'))
```

- b. This example creates a 20-year RA certificate signed by the CA certificate created in Example 1a.

```
RACDCERT GENCERT ID(PKISRVD) SUBJECTSDN(
  CN('Registration Authority')
  OU('Human Resources Certificate Authority')
  O('Your Company, Inc') C('US')) WITHLABEL('Local PKI RA')
  NOTAFTER(DATE(2026/05/06))
  SIZE(1024) KEYUSAGE(HANDSHAKE)
  SIGNWITH(CERTAUTH LABEL('Local PKI CA'))
```

2. Back up your CA certificate, RA certificate (if created), and their associated private keys to password-protected data sets using the RACDCERT EXPORT command.

**Example:**

```
RACDCERT CERTAUTH EXPORT(LABEL('Local PKI CA'))
  DSN('PKISRVD.PRIVATE.KEY.P12BIN')
  FORMAT(PKCS12DER) PASSWORD('your-passphrase')

RACDCERT ID(PKISRVD) EXPORT(LABEL('Local PKI RA'))
  DSN('PKISRVD.PRIVATE.RAKEY.P12BIN')
  FORMAT(PKCS12DER) PASSWORD('your-passphrase')
```

3. (Optional) If you want to use ICSF for private key protection and signing, migrate the private keys to ICSF using the RACDCERT ADD command. For this step to be successful, ICSF must be operational and configured for RSA operations. (For additional information about ICSF, see *z/OS Cryptographic Services ICSF Administrator's Guide*.)

**Example:**

```
RACDCERT CERTAUTH ADD('PKISRVD.PRIVATE.KEY.P12BIN') PASSWORD('your-passphrase') ICSF

RACDCERT CERTAUTH ADD('PKISRVD.PRIVATE.RAKEY.P12BIN') PASSWORD('your-passphrase') ICSF
```

4. Create a key ring for the PKI Services daemon and add the CA certificate and RA certificate (if created) to it so that PKI Services can use the certificates. The example creates a key ring called CAring for user ID PKISRVD and connects the CA and RA certificates to it.

**Important:** Make sure your CA certificate is marked with the TRUST or HIGHTRUST attribute in RACF. (Otherwise, PKI Services will not be able to



use the certificate.) Check this by issuing the RACDCERT LIST command and execute the RACDCERT ALTER command to change it if needed.

**Example:**

```
RACDCERT ADDRING(CAring) ID(PKISRVD)
```

```
RACDCERT ID(PKISRVD) CONNECT(CERTAUTH LABEL('Local PKI CA') RING(CAring
  USAGE(PERSONAL) DEFAULT)
```

```
RACDCERT ID(PKISRVD) CONNECT(ID(PKISRVD) LABEL('Local PKI RA') RING(CAring)
  USAGE(PERSONAL))
```

5. Authorize the PKI Services to use RACF certificates and act as the CA. The daemon user ID needs (PKISRVD) needs access to the FACILITY class resources listed in Table 79. Define the FACILITY resources and RACLIST the FACILITY class and if not already done. Refresh the FACILITY class when defined.

**Examples:**

```
SETROPTS RACLIST(FACILITY)
```

```
RDEFINE FACILITY IRR.DIGTCERT.GENCERT
RDEFINE FACILITY IRR.DIGTCERT.LISTRING
RDEFINE FACILITY IRR.DIGTCERT.LIST
```

```
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) ID(PKISRVD) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(PKISRVD) ACCESS(READ)
```

```
SETROPTS RACLIST(FACILITY) REFRESH
```

*Table 79. Access PKISRVD needs to use RACF certificates*

Resource	Access
IRR.DIGTCERT.GENCERT	CONTROL (if the CA certificate was created under CERTAUTH)
	READ (if the certificate was created under the PKI Services daemon user ID)
IRR.DIGTCERT.LISTRING	READ
IRR.DIGTCERT.LIST	READ

## Configuring the z/OS HTTP Server for SSL mode

The PKISERV application requires the z/OS HTTP Server to operate in three modes. That is why PKI Services requires two z/OS HTTP Servers. The modes are:

- Normal
- SSL without client authentication
- SSL with client authentication.

For SSL, your server needs to obtain a digital certificate. You can:

- Purchase one from an external source
- Create one using RACF

**Note:** If your server is already operating in SSL mode, you can skip the following section, “Using RACF to obtain a certificate for the Web server” on page 356.

## Using RACF to obtain a certificate for the Web server

The z/OS HTTP Server supports using either gskkyman key databases (.kdb files) or RACF (SAF) key rings for the server's certificate store. You are expected to use SAF key rings if setting up their Web server for the first time.

**Note:** If you have already set up your Web server using gskkyman, you can continue to use it.

Use RACDCERT to generate the server certificate signed by the new Certificate Authority.

### Example:

```
RACDCERT GENCERT ID(WEBSRV) SIGNWITH(CERTAUTH LABEL('Local PKI CA'))
        WITHLABEL('SSL Cert') SUBJECTSDN(CN('www.YourCompany.com') O('Your Company Inc')
        L('Millbrook') SP('New York') C('US'))
```

The Web server needs a key ring containing its new certificate and any trusted CA certificate. The RACDCERT command with operands ADDRING and CONNECT also sets this up. For example, the RACDCERT commands to create a key ring called SSLring for user ID WEBSRV and to connect the Web server and CA certificates to it are:

### Example:

```
RACDCERT ADDRING(SSLring) ID(websrv)
RACDCERT ID(websrv) CONNECT(CERTAUTH LABEL('Local PKI CA')) RING(SSLring)
        USAGE(PERSONAL) DEFAULT)
RACDCERT ID(websrv) CONNECT(ID(websrv) LABEL('SSL Cert') RING(SSLring)
        USAGE(PERSONAL) DEFAULT)
```

Export the CA certificate to an MVS data set. Then OPUT it to an file system file so that it can be made available to your clients.

### Example:

```
RACDCERT EXPORT(LABEL('Local PKI CA'))
        CERTAUTH DSN('pkisrvc.cacert.derbin') FORMAT(CERTDER)
```

## Enabling the z/OS HTTP Server for surrogate operation

Your server must be able to act as a surrogate for clients requesting certificates. To enable this, create:

- Profile BPX.SERVER in the FACILITY class
- Profile BPX.SRV.PKISERV in the SURROGAT class.

Give the z/OS HTTP Server daemon user ID READ access to both of these profiles.

## Enabling the PKI Services daemon to call OCSF functions

For access to OCSF, the PKI Services daemon needs READ access to BPX.SERVER. If program control is in effect (for example, z/OS UNIX level security), then the daemon needs READ access to BPX.DAEMON as well.

## IKYSETUP sample

IKYSETUP contains the commands to perform the RACF administrator tasks of adding groups and user IDs, setting up access control, creating CA, RA, and SSL certificates, and setting up daemon security. The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYSETUP.

```

/* REXX */
/*****
/*
/* DESCRIPTIVE NAME:  PKI Services RACF setup CLIST
/*
/* Licensed Materials - Property of IBM
/* 5694-A01
/* (C) Copyright IBM Corp. 2001, 2006
/* Status = HKY7730
/*
/*
/*01* EXTERNAL CLASSIFICATION: OTHER
/*01* END OF EXTERNAL CLASSIFICATION:
/*
/* FUNCTION:
/*
/* This CLIST will issue the RACF TSO commands necessary to set up*
/* security for PKI Services. It must be run from TSO by a user ID*
/* that is RACF SPECIAL.
/*
/* USAGE:
/*
/* 1) Read accompanying PKI Services post installation
/* instructions.
/* 2) Perform necessary prerequisite product installation for
/* the webserver (websphere), LDAP, etc.
/* 3) Make note of any predetermined values such as the LDAP
/* suffix, webserver fully qualified domain name, and the
/* settings contained in the pkiserv.conf file.
/* 4) Copy the CLIST to a data set where you can edit it.
/* 5) Examine the entire CLIST, in particular, the configurable
/* section.
/* 6) Modify the values in the configurable section as needed for
/* your installation.
/* 7) Run the CLIST. Syntax:
/*
/* EX 'data-set-name(IKYSETUP)' 'RUN(YES | NO | PROMPT)'
/*
/* where: YES - indicates to run CLIST as is
/* NO - indicates to display the commands only
/* PROMPT - indicates to prompt the user prior
/* to invoking each command
/*
/* DISCLAIMER:
/*
/* This CLIST is not intended to cover every possible customer
/* scenario. Modification of the actual commands to be issued
/* may be required
/*
*****/
/* Change-Activity:
/* $L1=PKIS4 , HKY7708, 020514, JWS: PKI Services
/* $D1=MG01103, HKY7708, 020813, BRW:
/* $D1=MG01405, HKY7708, 021114, BRW:
/* $L2=PKIS6 , HKY7720, 040204, TCG: PKI Services VI
/* $L3=PKIS6 , HKY7720, 040130, MSF: PKI Services VI
/* $L4=PKIS7 , HKY7730, 050301, TCG: PKI Services VII
/* $D2=MG06477, HKY7730, 051018, TCG: Incorrectly formed command
/*
/* Change Descriptions:
/* A - PCICC and CSSM support @L1A*/
/* C - Use restrict_surrog when checking for whether @D1A*/
/* surrogate user should be restricted @D1A*/
/* C - change SEZALINK to SEZALOAD @D1A*/
/* C - LDAP and SSL loadlibs now in SYS1.SIEALNKE, updated @L2A*/
/* program control setup. @L2A*/
/* A - DSA key support @L3A*/
/* A - Added RA certificate and CA Domain support @L4A*/
/* C - Updated RACDCERT command for connecting the RA cert */

```

```

/*      to the PKI daemons' keyring.                                @D2A*/
/*                                                                 */
/*****

trace value('0')

/*-----*/
/* configurable section                                           */
/*-----*/

/*-----*/
/* Part 1 - Things you must change */
/*-----*/

/*****
/* This exec will create the certificate, private key, and      */
/* keyring needed for your certificate authority.                */
/*                                                                 */
/* You must update the distinguished name of your certificate    */
/* authority defined below. The suffix of this DN must match     */
/* the suffix set up for your LDAP directory (suffix value from */
/* your slapd.conf file).                                       */
/*                                                                 */
/* Typically, Certificate Authorities have distinguished names    */
/* in the following form:                                         */
/*      OU=<your-CA's-friendly-name>,O=<your-organization>,       */
/*      C=<your-2-letter-country-abbreviation>                   */
/*                                                                 */
/* e.g., OU=Human Resources Certificate Authority.O=IBM,C=US     */
/*                                                                 */
/* If you already have your CA certificate and private key set   */
/* up in RACF, set ca_dn="" and update the ca_label variable to  */
/* equal your CA certificate's label. Note, it must reside       */
/* under CERTAUTH                                                */
/*                                                                 */
/* If you are running with Multiple-CAs:                         */
/* You could run IKYSETUP once for each separate CA you         */
/* want to operate, changing ca_domain everytime. The           */
/* ca_domain value will help qualify the other variables thus   */
/* reducing the amount of work the RACF administrator needs     */
/* to perform. Otherwise, set to NULL.                           */
/*                                                                 */
/*****
ca_domain = ""                                                    /* @L4A*/

if LENGTH(ca_domain) > 8 then                                     /* @L4A*/
    ca_domain_trunc = LEFT(ca_domain,8)                           /* @L4A*/
else                                                             /* @L4A*/
    ca_domain_trunc = ca_domain                                   /* @L4A*/

OrgUnit = STRIP(ca_domain "Human Resources Certificate Authority") /* @L4A*/
ca_dn= "OU(''||OrgUnit||'|')",
      "O('Your Company')",
      "C('Your Country 2 Letter Abbreviation')"                 /* @L4C*/

ca_label = STRIP(ca_domain "Local PKI CA") /* Label for CA
                                           certificate with the
                                           CA Domain name
                                           prepended @L4A*/

/*****
/* ra_label:                                                    */
/* A "must change" variable - default: "Local PKI RA"          */
/* ra_dn:                                                       */
/* A "must change" variable. If you don't wish to have PKI     */
/* Services operate with a separate RA certificate, set         */
/* ra_dn=""                                                     */
/*****

ra_label = "Local PKI RA" /* Label for RA certificate @L4A*/

if (ra_label = "") then /* If no RA Label ... @L4A*/
    ra_dn=""            /*@L4A*/
else                    /*@L4A*/
    ra_dn=,             /*@L4A*/
    "CN('Registration Authority')",
    ca_dn

```

```

/*****
/* This exec will create the certificate, private key, and */
/* keyring needed for your webserver. (Required for SSL.) */
/* */
/* You must update the distinguished name of your */
/* webserver. The Common Name (CN) must match your webserver's */
/* fully qualified domain name. */
/* */
/* e.g., CN=www.ibm.com,O=IBM,C=US */
/* */
/* If you already have your webserver configured for SSL, set */
/* web_dn="". */
/*****
web_dn=,
"CN('www.YourCompany.com')",
"O('Your Company')",
"L('Your City')",
"SP('Your Full State or Province Name')",
"C('Your Country 2 Letter Abbreviation')"
```

```

/*****
/* The sample web server protection directives supplied by PKI */
/* use SSLring for the web server's SAF key ring. If you change */
/* the value below, you will need to modify the "KeyFile" */
/* directive in the samples/httpd.conf and samples/httpd2.conf */
/* files when configuring the web server. */
/* */
/* If you already have your webserver configured for SSL and */
/* are using a SAF key ring (vs a gskkyman keyfile), then set */
/* web_ring equal to your webserver's SAF key ring name. If you */
/* are using a gskkyman keyfile, then set web_ring="". Note, */
/* you will have to add the CA's certificate to the webserver's */
/* keyfile manually */
/*****
web_ring = "SSLring" /* SAF keyring for web server */
```

```

/*****
/* You must provide UID and GID values for the user IDs and */
/* groups being created below */
/*****
daemon="PKISRV" /* user ID for PKI daemon */
daemon_uid="554" /* uid for PKI daemon */
surrog="PKISERV" /* user ID for the surrogate */
surrog_uid="555" /* uid for the surrogate id */
```

```

/*****
/* pkigroup members are authorized to administer PKI Services */
/* certificates and certificate requests. If you know the user */
/* IDs that should be connected to this group, update the */
/* pkigroup_mem stem variable. If not, you can always connect */
/* users later. */
/* */
/* If you do not wish to have this exec create this group, */
/* set the group name to "" */
/* */
/*****
pkigroup="PKIGRP" /* PKI Services Admin group name */
pki_gid="655" /* PKI Services Admin group id */
pkigroup_mem.0=0 /* Number of pkigroup members to connect */
pkigroup_mem.1=""
```

```

/*-----*/
/* Part 2 - Questions you must answer */
/*-----*/
```

```

/*****
/* Question 1 - Restrict the surrogate user ID? */
/* */
/* The surrogate user ID is the identity assigned to client */
/* processes when requesting certificate services. The */
/* RESTRICTED attribute can be assigned to this ID to limit the */
/* resources available to this user should the user ID be */
/* hijacked by an unfriendly client (hacker). We recommend */
/* that you run the surrogate this way. However, this probably */
/* will cause additional setup work. If you want the RESTRICTED */
/* attribute assigned now, set restrict_surrog=1. Note, you */
/* can always do this at some later time. */
/*****
restrict_surrog=0
```

```

/*****
/* Question 2 - Use ICSF?
/*
/* There are four possible choices for generation and
/* protection of your CA's private key:
/*
/* - Generate the RSA key using software and retain it as a
/* software key. This is the default. (Option 0)
/*
/* - Generate the key using software then store the key in
/* ICSF. (Option 1)
/*
/* - Generate the key through ICSF using the PCI cryptographic
/* coprocessor (PCICC) then store the key in ICSF. (Option 2)
/*
/* - Generate the DSA key using software and retain it as a
/* software key. This key cannot be saved in ICSF. (Option 3)
/*
/* Notes:
/*
/* - For options 1 and 2, ICSF must be configured for PKA
/* support and running. Additionally, for option 2, a PCICC
/* must be present and operational.
/*
/* - Option 2 and Option 3 are the only ways to generate
/* a private key larger than 1024 bits.
/*
/* - If option 2 is selected, the certificate and private key
/* will not be backed up by this exec.
/*
/* - If you select option 0, you can always migrate the key
/* to ICSF later (recommended). However, if you wish to use
/* the PCICC, you must select that option now.
/*
/* Select the option desired by setting key_type=0, 1, 2 or 3.
*****/
key_type=0

/*****
/* If you set key_type=1 or key_type=2 above, you will need to
/* restrict access to the CA's private key. Unless you indicate
/* otherwise, this exec will activate the CSFKEYS class,
/* create a profile in the CSFKEYS class to protect the CA's
/* private key, and permit the PKI Services daemon to use it.
/*
/* If you are already using ICSF, then you may have profiles in
/* the CSFSERV class protecting ICSF services. The PKI Services
/* daemon would need access to the profile that covers the
/* CSFDSV and CSFDSG services. Also, the PKI Services surrogate
/* ID would need access to the profile that covers the
/* CSFENC and CSFDEC services. You may also have a RACF group
/* for authorized ICSF users. Both of these user IDs
/* would need to be added to this group.
/*
/* Set the following variables as needed:
/*
/* csfkeys_profile - Profile to be created in the CSFKEYS class
/* Set the value to '' if you don't want the profile
/* csfserv_profile - Profile to be created in the CSFSERV class
/* e.g., 'CSF*'
/* csfusers_grp - Group name for authorized ICSF users
/* e.g., 'ICSFUGRP'
*****/
csfkeys_profile='IRR.DIGTCERT.CERTIFAUTH.*'
csfserv_profile='CSF*'
csfusers_grp=''

/*****
/* Question 3 - Back up your private key?
/*
/* The exec will prompt you to enter a pass phrase to encrypt a
/* backup copy of your CA's certificate and private key.
/* Caution, the text you enter at the prompt WILL be displayed
/* at the terminal. Backup is highly recommended. If you do not
/* wish to back up your CA's certificate and private key to a
/* pass phrase encrypted data set, set key_backup=0. The back up
/* may be done later if the key is not stored in ICSF.
/*
/* Note, back up is not performed if the CA certificate was not

```

```

/* created by this exec or if you specified key_type=2 above. */
/* *****/
key_backup=1

/*****/
/* Question 4 - Set up z/OS UNIX level security? */
/* */
/* z/OS UNIX may be set up to operate with a higher level of */
/* security than traditional UNIX. While we recommend this, it */
/* difficult to set up. You may want to defer this until later. */
/* */
/* If you don't want to set up UNIX security now, leave */
/* unix_sec=0. */
/* */
/* If you already have UNIX level security established and wish */
/* to continue it, set unix_sec=1. */
/* */
/* If you don't have UNIX level security established and wish */
/* to establish it now, set unix_sec=2. Note additional manual */
/* configuration probably will be required. This can be done */
/* by adding, removing, updating members of the two stem */
/* variables below. The pgmcntl_dsn stem contains the data set */
/* names of load libraries that need program control. The */
/* bpx_userid stem contains the user IDs of your server daemons. */
/* (These need access to BPX.SERVER and BPX.DAEMON in the */
/* FACILITY class.) Again, you can defer this until later by */
/* leaving unix_sec=0 */
/*****/
unix_sec=0
pgmcntl_dsn.0=8 /* Number of program controlled data sets below @L2C*/
pgmcntl_dsn.1="'CEE.SCEERUN'"
pgmcntl_dsn.2="'CBC.SCLBDLL'"
pgmcntl_dsn.3="'SYS1.SIEALNKE'" /* Common LINKLIST PDSE dataset @L2A*/
pgmcntl_dsn.4="'SYS1.CSSLIB'" /* @L2C*/
pgmcntl_dsn.5="'TCP/IP.SEZALOAD'" /* @L2C*/
pgmcntl_dsn.6="'SYS1.LINKLIB'" /* @L2C*/
pgmcntl_dsn.7="'CSF.SCSFMODE0'" /* @L2C*/
pgmcntl_dsn.8="'CSF.SCSFMODE1'" /* @L2C*/
bpx_userid.0=1 /* Number of additional bpx server ids below */
bpx_userid.1="OMVSKERN"

/*-----*/
/* Part 3 - Things you can change */
/*-----*/

/*****/
/* Label of the CA certificate that is the superior (signer) of */
/* the PKI Services CA, if self sign leave blank */
/*****/
signing_ca_label = "" /*@L4A*/

/*****/
/* This exec will record results to a log data set if desired. */
/* the name of the data set is specified below. If you do not */
/* want log data set recording, set log_dsn="" (Not recommended) */
/*****/
if (ca_domain = "") then /* If no CA Domain... @L4A*/
    log_dsn="IKYSETUP.LOG" /* Under your ID */
else /* Else use CA Domain @L4A*/
    log_dsn=ca_domain_trunc||".IKYSETUP.LOG" /* CA Domain qualified @L4A*/

/*****/
/* Note IKYCVSAM, the sample JCL to create VSAM datasets and */
/* pkiserv.conf expect the object store and ICL datasets to */
/* have PKISRVD as their high level qualifier. */
/* Changing either "daemon" or "vsamhlq" will */
/* require making the same change to IKYCVSAM and pkiserv.conf */
/*****/
vsamhlq=daemon /* HLQ for VSAM data sets. Same as daemon ID */

web_label = "SSL Cert" /* Label for web server cert */

ca_expires ="2020/01/01" /* date the CA certificate for
                           certificate authority should
                           expire */

web_expires ="2020/01/01" /* date the certificate for
                           web server SSL should
                           expire */

```

```

if (ca_domain = "") then          /* If no CA Domain...    @L4A*/
  ca_ring="CAring"                /* keyring name for PKI Srvs */
else                              /* Else use CA Domain    @L4A*/
  ca_ring="CAring."||ca_domain    /* CA Domain qualified  @L4A*/

/*****
/* You can select the size (in bits) of your CA's private key. */
/* The range is 512-2048. The default is 1024. Note, if you */
/* wish the key size to be greater than 1024, you must select */
/* key_type=2 or key_type=3 above. */
*****/
ca_keysize="1024"

/*****
/* Data set to contain the backup copy of the CA certificate */
/* and private key. (pass phrase encrypted PKCS#12 format) */
*****/
if (ca_domain = "") then          /* If no CA Domain...    @L4A*/
  backup_dsn = " "||daemon||".PRIVATE.KEY.BACKUP.P12BIN"
else                              /* Else use CA Domain    @L4A*/
  backup_dsn = " "||daemon||". "||ca_domain_trunc||",
    ".PRIVATE.KEY.BACKUP.P12BIN"
    /* CA Domain qualify backup dsn @L4A*/

/*****
/* Data set to contain the exported copy of the CA certificate */
/* (DER encoded). This is to be OPUT to an HFS file later to */
/* enable easy downloading by clients. */
*****/
if (ca_domain = "") then          /* If no CA Domain...    @L4A*/
  export_dsn = " "||daemon||".CACERT.DERBIN"
else                              /* Else use CA Domain    @L4A*/
  export_dsn = " "||daemon||". "||ca_domain_trunc||".CACERT.DERBIN"
    /* CA Domain qualify export dsn @L4A*/

/*****
/* Data set to contain the backup copy of the RA certificate */
/* and private key. (pass phrase encrypted PKCS#12 format) */
*****/
ra_backup_dsn = " "||daemon||".PRIVATE.RAKEY.BACKUP.P12BIN" /* @L4A*/

/*****
/* This EXEC expects the web server to be set up. If this is */
/* not the case, please refer to: */
/* z/OS HTTP Server Planning, Installing and Using. */
/* If the user ID assigned to the IBM HTTP Server Daemon is not */
/* WEBSRV, please update the assignment below. */
*****/
webserver="WEBSRV"

/*-----*/
/* End of configurable section */
/*-----*/

parse upper arg "RUN(" runopt ")"

if runopt = '' then
  runopt="NO"
if runopt ^= "YES" & runopt ^= "PROMPT" & runopt ^= "NO" then do
  say "syntax ex 'data-set-name(IKYSETUP)' 'run(yes | no | prompt)'"
  return 8
end
if runopt ^= "YES" & runopt ^= "PROMPT" then
  runopt="NO"

say 'IKYSETUP EXEC invoked ...'
return_code= '0'
max_return_code= '0'
logdata.0=0

if log_dsn ^= "" then do
  say "Allocating log data set" log_dsn "..."
  x = OUTTRAP(MSGS.)
  "FREE FI(IKYLOGDD)"
  "FREE DA("||log_dsn||")"
  "DELETE" log_dsn
  x = OUTTRAP('OFF')
  "ALLOCATE DA("||log_dsn||") FILE(IKYLOGDD) RECFM(V B)" ,
  " LRECL(256) DSORG(PS) BLKSIZE(2560) SP(1,1) TRACKS "
  al_rc= rc

```



```

IF al_rc ^= 0 THEN
  do
    say 'Allocation of log data set failed.'
    return 8
  end
end

call logsay "RUN("runopt") requested on" DATE() 'at' TIME() '...'
if runopt="NO" then
  call logsay "Running in test mode. Commands are not being invoked"
  /*****
  /* Create the daemon and surrogate user IDs using RACF ADDUSER TSO*/
  /* command. Give them an OMVS segment since they will need access */
  /* to UNIX System Services.
  *****/

call logsay2 "Creating users and groups ..."
call tsoserv "ADDUSER " daemon "name('PKI Srvs Daemon')",
  " nopassword",
  " omvs(uid("daemon_uid"))",
  " assize(256000000)",
  " threads(512))"

if restrict_surrog=1 then /*@D1C*/
  resattr="restricted"
else
  resattr=""
call tsoserv "ADDUSER " surrog "nopassword",
  resattr,
  " omvs(uid("surrog_uid"))",
  " name('PKI Srvs Surrogate')"
```

```

call tsoserv "SETOPTS EGN GENERIC(DATASET)"

call tsoserv "ADDSD 'vsamhlq'."*' UACC(NONE)"

if pkigroup ^= "" then do
  call tsoserv "ADDGROUP " pkigroup "OMVS(GID("pki_gid"))"
  do i = 1 to pkigroup_mem.0
    call tsoserv "CONNECT" pkigroup_mem.i "GROUP("pkigroup)"
  end
end

/*****
* Give the administrators access to the VSAM data sets
* identified in the [ObjectStore] section of
* the pkiserv.conf file.
*****/
call logsay2 "Allowing administrators to access PKI databases ..."
call tsoserv "PERMIT 'vsamhlq'."*' ID("pkigroup") ACCESS(CONTROL)"
call tsoserv "SETOPTS GENERIC(DATASET) REFRESH"

/*****
/* In order to create and sign digital certificates for others */
/* you need to define or import in RACF a Certificate Authority */
/* certificate and associated private key.
/* This is done using the RACF RACDCERT GENCERT command.
*****/
if ca_dn ^= "" then do
  call logsay2 "Creating the CA certificate ..."
  if signing_ca_label = "" then /*@L4A*/
    certcmd = "RACDCERT GENCERT CERTAUTH SUBJECTSDN("ca_dn")",
    " WITHLABEL('ca_label') NOTAFTER(DATE("ca_expires"))",
    " SIZE("ca_keysize)"
  else /*@L4A*/
    certcmd = "RACDCERT GENCERT CERTAUTH SUBJECTSDN("ca_dn")",
    " WITHLABEL('ca_label')",
    " SIGNWITH(CERTAUTH LABEL('signing_ca_label'))",
    " NOTAFTER(DATE("ca_expires"))",
    " SIZE("ca_keysize)" /*@L4A*/
  if key_type=1 & key_backup=0 then
    certcmd= certcmd || " ICSF"
  else if key_type=2 then
    certcmd= certcmd || " PCICC"
  else if key_type=3 then
    certcmd=certcmd || " DSA"

  call tsoserv certcmd

  if key_type^=2 & key_backup=1 then do

```

```

/*****
/* Export certificate and key to PKCS#12 dataset */
/*****
    say ""
    say "Enter a passphrase to protect the key. You will need"
    say "this value later if you need to restore the key."
    say ""
    say "Attention, the value will be displayed in the screen:"
    parse pull pp
    call logsay2 "Backing up the CA certificate ..."
    certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('ca_label'))",
              " DSN('backup_dsn') FORMAT(PKCS12DER)",
              " PASSWORD('pp')"

    call tsoserv certcmd
end

if key_type=1 & key_backup=1 then do
/*****
/* If ICSF was requested and key backup, reload the certificate */
/* to get the key migrated to ICSF */
/*****
    call logsay2 "Migrating the CA's private key to ICSF ..."
    certcmd = "RACDCERT CERTAUTH ADD('backup_dsn'",
              " PASSWORD('pp') ICSF"

    call tsoserv certcmd
end

end /* ca_dn ^= "" */
/*****
/* Mark the CA certificate as HIGHTRUST so HostIdMappings */
/* are honored */
/*****
call logsay2 "Marking CA certificate as HIGHTRUST ..."
certcmd = "RACDCERT CERTAUTH ALTER(LABEL('ca_label')) HIGHTRUST"
call tsoserv certcmd

/*****
/* The CA certificate must be saved to a data set so that it may */
/* be OPUT to an HFS file. */
/*****
call logsay2 "Saving the CA certificate to a data set for OPUT ..."
certcmd = "RACDCERT CERTAUTH EXPORT(LABEL('ca_label'))",
          " DSN('export_dsn') FORMAT(CERTDER)"
call tsoserv certcmd

if (ra_label ^= "") then do /* @L4A*/
/*****
/* Creating RA Certificate */
/*****
call logsay2 "Creating the RA certificate ..." /* @L4A*/
certcmd = "RACDCERT ID('daemon') GENCERT SUBJECTSDN('ra_dn'",
          " KEYUSAGE(HANDSHAKE) SIGNWITH(CERTAUTH LABEL('ca_label'))",
          " NOTAFTER(DATE('ca_expires')) WITHLABEL('ra_label'))" /* @L4A*/
call tsoserv certcmd /* @L4A*/

/*****
/* Backing up RA Certificate */
/*****
call logsay2 "Backing up RA certificate ..." /* @L4A*/
certcmd = "RACDCERT ID('daemon') EXPORT(LABEL('ra_label'))",
          " DSN('ra_backup_dsn') FORMAT(PKCS12DER)",
          " PASSWORD('pp')" /* @L4A*/
call tsoserv certcmd /* @L4A*/
end /* @L4A*/

/*****
/* The CA/RA certificate must be placed in a key ring so that */
/* PKI Services can access it. */
/*****
call logsay2 "Creating the PKI Services keyring ..."
call tsoserv "RACDCERT ADDRING('ca_ring') ID('daemon')"
call tsoserv "RACDCERT ID('daemon') CONNECT(CERTAUTH",
          " LABEL('ca_label'))",
          " RING('ca_ring') USAGE(PERSONAL) DEFAULT) "
if (ra_label ^= "") then /* @L4A*/
    call tsoserv "RACDCERT ID('daemon') CONNECT(LABEL('ra_label'))",
          " RING('ca_ring') USAGE(PERSONAL))" /* @D2C*/

```

```

/*****
/* Create the certificate for the webserver signed by your new CA */
*****/

if web_dn ^= "" then do
  call logsay2 "Creating the Webserver SSL certificate and keyring ..."
  call tsoserv "RACDCERT GENCERT ID("webserver") SIGNWITH(CERTAUTH)",
    " LABEL('ca_label')",
    " WITHLABEL('web_label') SUBJECTSDN("web_dn")",
    " NOTAFTER(DATE("web_expires"))"

/*****
/* Add the certificate to the webserver's RACF (SAF) key ring */
*****/
  call tsoserv "RACDCERT ADDRING("web_ring") ID("webserver")"
  call tsoserv "RACDCERT ID("webserver") CONNECT(ID("webserver")",
    " LABEL('web_label') RING("web_ring") USAGE(PERSONAL) DEFAULT)"
end /* web_dn ^= "" */

/*****
/* Add the CA certificate to the webserver's RACF (SAF) key ring*/
*****/
if web_ring ^= "" then
  call tsoserv "RACDCERT ID("webserver") CONNECT(CERTAUTH",
    " LABEL('ca_label') RING("web_ring"))"

if unix_sec = 0 then do
/*****
/* Not setting up z/OS UNIX higher security. However, the */
/* daemon does need access to one server service. So, if the */
/* daemon user ID is not uid 0, then it must be given read */
/* access to FACILITY class profile BPX.SERVER */
*****/
  if strip(daemon_uid,L,'0') ^= "" then do /* if daemon not uid 0 */
    call logsay2 "Giving" daemon "access to BPX.SERVER ..."
    call tsoserv "RDEFINE FACILITY BPX.SERVER"
    call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY)",
      " ID("daemon") ACCESS(READ)"
  end
end
else do
  call logsay2 "Setting up or modifying z/OS UNIX security ..."
  if unix_sec = 2 then do
/*****
/* Set up z/OS UNIX to operate with a higher level of */
/* security than traditional UNIX, by defining BPX.SERVER and */
/* BPX.DAEMON classes. */
*****/
    call tsoserv "RDEFINE FACILITY BPX.SERVER"
    call tsoserv "RDEFINE FACILITY BPX.DAEMON"
    do i = 1 to bpx_userid.0
      call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY)",
        " ID("bpx_userid.i") ACCESS(READ)"
      call tsoserv "PERMIT BPX.DAEMON CLASS(FACILITY)",
        " ID("bpx_userid.i") ACCESS(READ)"
    end
  end
end

/*****
/* To use the higher level of security, you need to establish */
/* RACF program control and enable the PKI Services daemon */
/* user ID and webserver daemon user ID to access protected */
/* UNIX daemon services. */
*****/
  call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY) ID("daemon")",
    " ACCESS(READ)"
  call tsoserv "PERMIT BPX.DAEMON CLASS(FACILITY) ID("daemon")",
    " ACCESS(READ)"
  call tsoserv "PERMIT BPX.SERVER CLASS(FACILITY) ID("webserver")",
    " ACCESS(UPDATE)"
  call tsoserv "PERMIT BPX.DAEMON CLASS(FACILITY) ID("webserver")",
    " ACCESS(READ)"

  if unix_sec = 2 then do
/*****
/* Set the PKI Services daemon and DLLs up for program control */
*****/
    call tsoserv "RDEFINE PROGRAM * UACC(NONE)"
    do i = 1 to pgmcntl_dsn.0
      call tsoserv "RALTER PROGRAM * ADDMEM("pgmcntl_dsn.i"//NOPADCHK)",

```

```

        " UACC(READ)"
    end
    call tsoserv "SETROPTS WHEN(PROGRAM)"
end
call tsoserv "PERMIT * CLASS(PROGRAM)",
    " ID("surrog") ACCESS(READ)"
call tsoserv "SETROPTS WHEN(PROGRAM) REFRESH"
end /* unix_sec ^= 0 */

/*****
/* Allow the daemon to be a certificate authority */
*****/
call logsay2 "Allowing the PKI Services daemon to act as a CA ..."
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.GENCERT"
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.LISTRING"
call tsoserv "RDEFINE FACILITY IRR.DIGTCERT.LIST"
call tsoserv "PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY)",
    " ID("daemon") ACCESS(CONTROL)"
call tsoserv "PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)",
    " ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
    " ID("daemon") ACCESS(READ)"

/*****
/* Allow the webserver to access its keyring */
*****/
call logsay2 "Allowing the Webserver to access its keyring ..."
call tsoserv "PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)",
    " ID("webserver") ACCESS(READ)"
call tsoserv "PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)",
    " ID("webserver") ACCESS(READ)"

/*****
/* Permit the webserver daemon User ID to switch identity to the */
/* surrogate Id */
*****/

call logsay2 "Allowing the Webserver to switch identity to "surrog" ..."
call tsoserv "SETROPTS CLASSACT(SURROGAT)"
call tsoserv "RDEFINE SURROGAT BPX.SRV."surrog
call tsoserv "PERMIT BPX.SRV."surrog" CLASS(SURROGAT)",
    " ID("webserver") ACCESS(READ)"
call tsoserv "SETROPTS RACLIST(SURROGAT) REFRESH"

if (key_type=1 or key_type=2) then do
/*****
/* Allow the daemon to use ICSF */
*****/
call logsay2 "Allowing the PKI Services daemon to use ICSF ..."
if csfkeys_profile ^= '' | csfserv_profile ^= '' then do
    call tsoserv "SETROPTS GENERIC(CSFKEYS CSFSERV)"
    call tsoserv "SETROPTS GENERIC(CSFKEYS CSFSERV) REFRESH"
end
if csfkeys_profile ^= '' then do
    call tsoserv "RDEFINE CSFKEYS" csfkeys_profile "UACC(NONE)"
    call tsoserv "PERMIT" csfkeys_profile "CLASS(CSFKEYS)",
        " ID("daemon") ACCESS(READ)"
    call tsoserv "SETROPTS CLASSACT(CSFKEYS) RACLIST(CSFKEYS)"
    call tsoserv "SETROPTS RACLIST(CSFKEYS) REFRESH"
end
if csfserv_profile ^= '' then do
    call tsoserv "RDEFINE CSFSERV" csfserv_profile "UACC(NONE)"
    call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)",
        " ID("daemon") ACCESS(READ)"
    call tsoserv "PERMIT" csfserv_profile "CLASS(CSFSERV)",
        " ID("surrog") ACCESS(READ)"
    call tsoserv "SETROPTS CLASSACT(CSFSERV) RACLIST(CSFSERV)"
    call tsoserv "SETROPTS RACLIST(CSFSERV) REFRESH"
end
if csfusers_grp ^= '' then do
    call tsoserv "CONNECT" daemon "GROUP(" csfusers_grp ")"
    call tsoserv "CONNECT" surrog "GROUP(" csfusers_grp ")"
end
end

/*****
/* Allow the daemon to use OCSF */
*****/
call logsay2 "Allowing the PKI Services daemon to use OCSF ..."
call tsoserv "PERMIT CDS.CSSM CLASS(FACILITY)",

```

```

    " ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT CDS.CSSM.CRYPTO CLASS(FACILITY)",
    " ID("daemon") ACCESS(READ)"
call tsoserv "PERMIT CDS.CSSM.DATALIB CLASS(FACILITY)",
    " ID("daemon") ACCESS(READ)"

/*****
 * Tie the daemon user ID to PKI Services started procedure
 *****/
call logsay2 "Creating the STARTED class profile for the daemon ..."
call tsoserv "RDEFINE STARTED PKISERV.* STDATA(USER("daemon"))"
call tsoserv "SETROPTS CLASSACT(STARTED) RACLIST(STARTED)"
call tsoserv "SETROPTS RACLIST(STARTED) REFRESH"

/*****
/* Give the surrogate user ID authority to request certificate */
/* generation functions. */
/*****
call logsay2 "Allowing "surrog" to request certificate functions ..."
call tsoserv "SETR GENERIC(FACILITY)"
/*
 * When a CA Domain is not specified, use the default value for
 * the end user functions generic profile name. When a CA Domain
 * is specified, append a dot/period(".") followed by the CA
 * Domain value to the default value.
 */
if (ca_domain = "") then                                /* @L4A*/
    profname = "IRR.RPKISERV.*"                          /* @L4A*/
else                                                    /* @L4A*/
    profname = "IRR.RPKISERV.*"||ca_domain_trunc         /* @L4A*/

call tsoserv "RDEFINE FACILITY "||profname               /* @L4C*/
call tsoserv "PERMIT "||profname||" CLASS(FACILITY)",
    " ID("surrog") ACCESS(CONTROL)"                      /* @L4C*/

/*****
/* The administrative functions of PKI Services are protected */
/* by the IRR.RPKISERV.PKIADMIN FACILITY class resource. */
/* The following commands give UPDATE access to the PKI */
/* services group to allow them to act on certificate */
/* requests and issued certificates. */
/*****
call logsay2 "Creating the profile to protect PKI Admin functions ..."

/*
 * When a CA Domain is not specified, use the default value for
 * the administrative function profile name. When a CA Domain
 * is specified, append a dot/period(".") followed by the CA
 * Domain value to the default value.
 */
if (ca_domain = "") then                                /* @L4A*/
    profname = "IRR.RPKISERV.PKIADMIN"                   /* @L4A*/
else                                                    /* @L4A*/
    profname = "IRR.RPKISERV.PKIADMIN."||ca_domain_trunc /* @L4A*/

call tsoserv "RDEFINE FACILITY "||profname               /* @L4C*/
call tsoserv "PERMIT "||profname||" CLASS(FACILITY)",
    " ID("pkigroup") ACCESS(UPDATE)"                    /* @L4C*/
call tsoserv "PERMIT "||profname||" CLASS(FACILITY)",
    " ID("surrog") ACCESS(NONE)"                         /* @L4C*/
call tsoserv "SETROPTS RACLIST(FACILITY) REFRESH"

/*****
/* Done. Now write to the log */
/*****
upper daemon vsamhlq export_dsn
call logsay " "
call logsay "-----"
call logsay "Information needed for PKI Services UNIX set up:"
call logsay "-----"
call logsay " "
call logsay "The daemon user ID is:"
call logsay " " daemon
call logsay " "
call logsay "The VSAM high level qualifier is:"
call logsay " " vsamhlq
call logsay,
"This is needed for the [ObjectStore] section in pkiserv.conf"
call logsay " "
call logsay "The PKI Services' DER encoded certificate is in data set:"

```

```

call logsay " " export_dsn
call logsay,
"This must be OPUT to /var/pkiserv/cacert.der with the BINARY option"
call logsay " "
call logsay "The fully qualified PKI Services' SAF keyring is:"
call logsay " " daemon"/"ca_ring
call logsay,
"This is needed for the [SAF] section in pkiserv.conf"
if ra_label ^= "" then do
    call logsay " " /*@L4A*/
    call logsay " " /*2@L4A*/
    call logsay "The label of the PKI Services' RA certificate is:"
    call logsay " " ra_label /*@L4A*/
    call logsay, /*@L4A*/
    "This is needed for the [SAF] section in pkiserv.conf" /*@L4A*/
end /*@L4A*/
call logsay " "
if ca_dn ^= "" then do
    call logsay "The PKI Services CA DN is:"
    call norm_dn ca_dn
    call logsay " " dn
    call logsay "The suffix must match the LDAP suffix in slapd.conf"
end
else
    call logsay "CA certificate not created by this exec"
call logsay " "

if ra_dn ^= "" then do
    call logsay "The PKI Services RA DN is:" /*@L4A*/
    call norm_dn ra_dn /*@L4A*/
    call logsay " " dn /*2@L4A*/
    call logsay "The suffix must match the LDAP suffix in slapd.conf"
end /*@L4A*/
else
    call logsay "RA certificate not created by this exec" /*@L4A*/
call logsay " " /*@L4A*/
call logsay, /*2@L4A*/
"The recommended location for the pkiservd.conf and pkiserv.tmpl is:"
if ca_domain = "" then
    call logsay " /etc/pkiserv" /*@L4A*/
else
    call logsay " /etc/pkiserv/"ca_domain /*@L4A*/
call logsay " " /*@L4A*/

call logsay, /*2@L4A*/
"Set the following environment variables in pkiserv.envs:"
if ca_domain = "" then
    call logsay " _PKISERV_CONFIG_PATH=/etc/pkiserv" /*@L4A*/
else do
    call logsay " _PKISERV_CA_DOMAIN="ca_domain /*@L4A*/
    call logsay " _PKISERV_CONFIG_PATH=/etc/pkiserv/"ca_domain /*@L4A*/
end /*@L4A*/
call logsay " " /*@L4A*/
call logsay, /*2@L4A*/
"Set the following environment variable in your httpd envvars files:"
if ca_domain = "" then
    call logsay " _PKISERV_CONFIG_PATH=/etc/pkiserv" /*@L4A*/
else
    call logsay " _PKISERV_CONFIG_PATH="TRANSLATE(ca_domain), /*@L4A*/
    "=/etc/pkiserv/"ca_domain /*@L4A*/
call logsay " " /*@L4A*/
if web_dn ^= "" then do
    call logsay "The webserver's SAF keyring is:"
    call logsay " " web_ring
    call logsay,
    "This is needed for the KeyFile directive in httpd*.conf files"
    call logsay " "
    call logsay "The Webserver's DN is:"
    call norm_dn web_dn
    call logsay " " dn
    call logsay "The left most RDN must be the webserver's fully",
    "qualified domain name"
end
else
    call logsay,
    "Webserver certificate and keyring not created. You must add the CA",
    "certificate as a 'trusted root' manually"
call logsay " "
if log_dsn ^= "" then do
    x = OUTTRAP(MSGS.)
    'EXECIO' logdata.0 'DISKW IKYLOGDD (FINIS STEM LOGDATA.'

```

```

'FREE FI(IKYLOGDD)'
x=OUTTRAP('OFF')
say "Commands complete. Results written to log data set" log_dsn
end
/*****/
/* Exit */
/*****/
say 'The IKYSETUP EXEC has completed.'
Exit max_return_code

/*****/
/* tsoserv - echo rc and commands and track highest rc */
/*****/
tsoserv:
Parse arg cmd
return_code = 0
skipit= 0
if runopt = "NO" | runopt = "PROMPT" then
call logsay cmd
if runopt = "PROMPT" then do
say "Run command (y/n) ?"
parse pull ans
if substr(ans,1,1) ^= 'Y' & substr(ans,1,1) ^= 'y' then
skipit= 1
end
if skipit = 0 then
if runopt = "YES" | runopt = "PROMPT" then do
msg_status= MSG('ON')
x=OUTTRAP('rac_ret.')
Address TSO cmd
return_code=rc
y=OUTTRAP('OFF')
call logsay 'Return code' return_code 'from->' cmd
If return_code\=0 then do
Do j=1 to rac_ret.0
call logsay rac_ret.j
end
end
end
max_return_code= max(max_return_code,return_code)
return return_code
return 0

/*****/
/* logsay - echo messages to the terminal and logdata stem */
/*****/
logsay:
Parse arg cmd
parse var cmd leftpart " PASSWORD(' pw ') " rightpart
if pw ^= "" then
cmd= leftpart "PASSWORD('*****') " rightpart
say cmd
k= logdata.0 + 1
logdata.k= cmd
logdata.0= k

return 0

/*****/
/* logsay2 - echo a blank line before echoing the command */
/*****/
logsay2:
Parse arg cmd2
call logsay " "
call logsay cmd2

return 0

/*****/
/* norm_dn - transform the RACF dn keywords to an LDAP dn */
/*****/
norm_dn:
parse arg in_dn
parse var in_dn q.1 "(" v.1 ")",
q.2 "(" v.2 ")",
q.3 "(" v.3 ")",
q.4 "(" v.4 ")",
q.5 "(" v.5 ")",
q.6 "(" v.6 ")",
q.7 "(" v.7 ")" rest

```

```

dns.= ""
do i = 1 to 7
  q= strip(q.i)
  upper q
  if q = "" then
    leave
  if q = "CN" then
    dns.1= "CN=" || v.i
  else
    if q = "T" then
      dns.2= "T=" || v.i
    else
      if q = "OU" then
        dns.3= "OU=" || v.i
      else
        if q = "O" then
          dns.4= "O=" || v.i
        else
          if q = "L" then
            dns.5= "L=" || v.i
          else
            if q = "SP" then
              dns.6= "ST=" || v.i
            else
              dns.7= "C=" || v.i
            end
          dn= ""
        do i = 1 to 7
          if dns.i ^= "" then
            if dn = "" then
              dn= dns.i
            else
              dn= dn || "," || dns.i
            end
          end
        return 0

```



## Chapter 26. Other code samples

This chapter provides code samples for the following files:

- `httpd.conf` and `httpd2.conf`, which contain z/OS HTTP Server directives. (See “z/OS HTTP Server configuration directives.”)
- `IKYCVSAM`, which is sample IDCAMS JCL to create VSAM data sets (regardless of whether you are using a sysplex or non-sysplex). (See “IKYCVSAM” on page 373.)
- `IKYMVSAM`, which is sample IDCAMS JCL to use if you are migrating from z/OS Version 1 Release 4 or lower. `IKYMVSAM` creates your VSAM alternate indexes and PATH data sets. (See “IKYMVSAM” on page 377.)
- `IKYRVSAM`, which is sample IDCAMS JCL to add VSAM record-level sharing (RLS) support. `IKYRVSAM` reallocates your VSAM data sets in preparation for sharing in a sysplex. (See “IKYRVSAM” on page 380.)
- `PKISERV.D`, which is a sample procedure to start PKI Services daemon. (See “PKISERV.D sample procedure to start PKI Services daemon” on page 384.)

**Note:** Other important programs are contained in other chapters:

- `IKYSETUP` (a REXX exec to set up RACF profiles). See Chapter 25, “The IKYSETUP REXX exec,” on page 351.
- `pkiserv.envars` (the PKI Services environment variables file). See “The pkiserv.envars environment variables file” on page 349.
- `pkiserv.conf` (the PKI Services configuration file). See Chapter 23, “The pkiserv.conf configuration file,” on page 343.

---

## z/OS HTTP Server configuration directives

The following listing might not be identical to the code sample shipped with the product. For the most current sample, see the `httpd.conf` sample z/OS HTTP Server configuration directives in the source directory `/usr/lpp/pkiserv/samples/`.

```
#-----#
# Licensed Materials - Property of IBM          #
# 5694-A01                                       #
# (C) Copyright IBM Corp. 2001,2002            #
# Status = HKY7708                             #
#                                               #
# Change-Activity:                             #
# $L1=PKIS4 , HKY7708, 020429, JWS: PKI Services #
#                                               #
# Change Descriptions:                         #
# A - Multiple application support              @L1A #
#-----#

# For a secure system, set the default User ID to %%CLIENT%%
UserId      %%CLIENT%%

# SSL support using a SAF keyring
keyfile SSLring SAF
# OR
# May use a gskkyman key database instead of SAF keyring
#keyfile /etc/key.kdb

sslmode on
sslport 443
Normalmode on
Protection PublicUser {
    ServerId      PublicUser
    UserID        PKISERV
}
```

## Other code samples

```

        Mask          Anyone
    }
    Protect /PKIServ/public-cgi/* PublicUser
    Protect /PKIServ/ssl-cgi-bin/* PublicUser
    Protect /PKIServ/* PublicUser
    Protect /Customers/public-cgi/* PublicUser
    Protect /Customers/ssl-cgi-bin/* PublicUser
    Protect /Customers/* PublicUser

    Protection AuthenticatedUser {
        ServerId      AuthenticatedUser
        AuthType      Basic
        PasswdFile     %%SAF%%
        UserID         %%CLIENT%%
        Mask           All
    }
    Protect /PKIServ/ssl-cgi-bin/auth/* AuthenticatedUser
    Protect /Customers/ssl-cgi-bin/auth/* AuthenticatedUser

    Protection SurrogateUser {
        ServerId      SurrogateUser
        AuthType      Basic
        PasswdFile     %%SAF%%
        UserID         PKISERV
        Mask           All
    }

    Protect /PKIServ/ssl-cgi-bin/surrogateauth/* SurrogateUser
    Protect /Customers/ssl-cgi-bin/surrogateauth/* SurrogateUser

    Redirect /PKIServ/ssl-cgi/*
    https://<server-domain-name>/PKIServ/ssl-cgi-bin/*
    Redirect /PKIServ/ssl-cgi/auth/*
    https://<server-domain-name>/PKIServ/ssl-cgi-bin/auth/*
    Redirect /PKIServ/ssl-cgi/surrogateauth/*
    https://<server-domain-name>/PKIServ/ssl-cgi-bin/surrogateauth/*
    Redirect /Customers/ssl-cgi/*
    https://<server-domain-name>/Customers/ssl-cgi-bin/*
    Redirect /Customers/ssl-cgi/auth/*
    https://<server-domain-name>/Customers/ssl-cgi-bin/auth/*
    Redirect /Customers/ssl-cgi/surrogateauth/*
    https://<server-domain-name>/Customers/ssl-cgi-bin/surrogateauth/*

    Redirect /PKIServ/clientauth-cgi/*
    https://<server-domain-name>:1443/PKIServ/clientauth-cgi/*
    Redirect /Customers/clientauth-cgi/*
    https://<server-domain-name>:1443/Customers/clientauth-cgi/*

    Exec      /PKIServ/public-cgi/* <application-root>/PKIServ/public-cgi/*
    Exec      /PKIServ/ssl-cgi-bin/* <application-root>/PKIServ/ssl-cgi-bin/*
    Exec      /Customers/public-cgi/* <application-root>/PKIServ/public-cgi/*
    Exec      /Customers/ssl-cgi-bin/* <application-root>/PKIServ/ssl-cgi-bin/*
    Pass      /PKIServ/cacerts/*      /var/pkiserv/*

    AddType .cer application/x-x509-user-cert      ebcdic 0.5 # Browser Certificate
    AddType .der application/x-x509-ca-cert        binary 1.0 # CA Certificate
```

The source of the following sample z/OS HTTP Server configuration directives for your /etc/httpd1443.conf file is /usr/lpp/pkiserv/samples/httpd2.conf.

```
#-----#
# Licensed Materials - Property of IBM          #
# 5694-A01                                       #
# (C) Copyright IBM Corp. 2001,2002            #
# Status = HKY7708                             #
#                                                #
# Change-Activity:                             #
# $L1=PKIS4 , HKY7708, 020429, JWS: PKI Services #
#                                                #
# Change Descriptions:                         #
#
```

```

# A - Multiple application support @L1A #
#-----#

# For a secure system, set the default User ID to %%CLIENT%%
UserId      %%CLIENT%%

# SSL support using a SAF keyring
keyfile SSLring SAF
# OR
# May use a gskkyman key database instead of SAF keyring
#keyfile /etc/key.kdb

sslmode on
sslport 1443
Normalmode off
SSLClientAuth strong
SSLX500CARoots local_and_x500
SSLX500Host <ldap-server-name>
SSLX500Port <ldap-port-number>
SSLX500UserID <ldap-distinguished-name>
SSLX500Password <ldap-password>

Protection RenewRevokeUser {
    ServerId      RenewRevokeUser
    AuthType      Basic
    UserID        PKISERV
    SSL_CLIENTAUTH Client
    Mask          Anyone
}

Protect /PKIServ/clientauth-cgi/* RenewRevokeUser
Protect /Customers/clientauth-cgi/* RenewRevokeUser

Protection AuthenticatedAdmin {
    ServerId      AuthenticatedAdmin
    AuthType      Basic

    UserID        %%CERTIF%%
    SSL_CLIENTAUTH Client
    Mask          Anyone
}

Protect /PKIServ/clientauth-cgi/auth/* AuthenticatedAdmin
Protect /Customers/clientauth-cgi/auth/* AuthenticatedAdmin

Redirect /PKIServ/public-cgi/*
http://<server-domain-name>/PKIServ/public-cgi/*
Redirect /PKIServ/ssl-cgi/*
https://<server-domain-name>/PKIServ/ssl-cgi-bin/*
Redirect /Customers/public-cgi/*
http://<server-domain-name>/Customers/public-cgi/*
Redirect /Customers/ssl-cgi/*
https://<server-domain-name>/Customers/ssl-cgi-bin/*

Exec /PKIServ/clientauth-cgi/* <application-root>/PKIServ/clientauth-cgi-bin/*
Exec /Customers/clientauth-cgi/* <application-root>/PKIServ/clientauth-cgi-bin/*

```

---

## IKYCVSAM

IKYCVSAM contains sample IDCAMS JCL to create VSAM data sets. IKYCVSAM is installed as a member of SYS1.SAMPLIB.

Use IKYCVSAM if you are creating VSAM data sets for the first time, regardless of whether you intend to use Parallel Sysplex support. However, if you intend to use Parallel Sysplex support, execute the IKYRVSAM job *after* this job to add VSAM record-level sharing (RLS) support. (See “IKYRVSAM” on page 380.)

## Other code samples

**Note:** The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYCVSAM.

```
//IKYCVSAM JOB <job card parameters>
//*****
//* SAMP:      IKYCVSAM
//*
//* Licensed Materials - Property of IBM
//* 5694-A01
//* (C) Copyright IBM Corp. 2001, 2006
//* Status = HKY7730
//*
//*****
//* This sample JCL may be used to create the VSAM data sets
//* PKI Services utilizes to store certificate requests and
//* issued certificates.
//*
//*****
//* Caution: This is neither a JCL procedure nor a complete job.
//* Before using this job step, you will have to make the following
//* modifications:
//*
//* 1) Change the job card to meet your system requirements.
//*
//* 2) If you wish to change the data set qualifiers from the
//* default value change all occurrences of "PKISRV.VSAM"
//* to a preferred value. If you choose to modify this value, be
//* sure to also modify the sample configuration file
//* appropriately(/etc/pkiserv/pkiserv.conf). If you are using
//* multiple CA Domains, IBM recommends using the first eight
//* characters of the CA Domain as one of the data set
//* qualifiers.
//*
//* 3) If you are using VSAM record level sharing (RLS), perform
//* the following steps:
//*
//* a) Replace the VOL(vvvvvv) statements in the DEFKSDS step
//* with STORCLAS(class-name) where class-name is the name of
//* the storage class defined for VSAM RLS.
//*
//* b) Remove the VOL(vvvvvv) statements from the DEFALTDX step.
//*
//* c) Remove all the SPANNED and CISIZE statements.
//*
//* If not using VSAM RLS, change all occurrences of vvvvvv to
//* the VOLSER value appropriate for the system this job is to be
//* run on. Do not remove the SPANNED and CISIZE statements.
//*
//* 4) If you wish to change the default userid to own the VSAM
//* data set, change the OWNER(PKISRVD) operand to the userid you
//* want to own the data sets. If you choose to modify this value
//* ensure you have modified the sample setup REXX exec (IKYSETUP)*
//* to account for this change.
//*
//* 5) If you wish to change either the primary or secondary record
//* allocation sizes for either the OST or ICL datasets from the
//* default value, update the RECORDS(50 50) operands on the
//* DEFINE CLUSTER or DEFINE ALTERNATE INDEX commands.
//*
//* **Note, do not change any of the numeric values other than
//* CYL or TRK
//*-----*
//* Change Activity:
//*
//* $L1=PKIS3 HKY7707 020314 PDJWS1: VSAM RLS @L1A*
//* $P1=MG00719 HKY7707 020416 PDJWS1: VSAM RLS 2 @P1A*
//* $L2=MG01176 HKY7708 020826 PDJWS1: VSAM scaling @L2A*
//* $P2=MG01346 HKY7708 021022 PDJWS1: JCL errors @P2A*
```

```

/** $L3=PKIS7 HKY7730 050228 PDTCG1: Multi-CA Support @L3A*
/** *
/** Change Description: *
/** *
/** C: Added STORCLAS instructions, LOG. Removed VOLUME DDs @L1C*
/** D: Removed FILE(VOLUME) statements @P1A*
/** C: Added more alt indexes and changed allocation parms @L2A*
/** C: Removed VOL keywords from ALTERNATEINDEX statements. @P2A*
/** Removed DD statements from BLDINDEX step. Added @P2A*
/** IEBGENER step to remove hardcoded binary zeros @P2A*
/** C: Updated prolog with CA Domain information @L3A*
/** C: Updated DEFALTDX step to use PKISRV.D.VSAM.OST.AIX.IX *
/** dataset name instead of the former dataset name of *
/** PKISRV.D.VSAM.AIX.IX. @P3A*
/**-----*
/**
/**-----*
/** Delete existing clusters, paths, alt indexes
/**-----*
//DELCLUST EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE -
    PKISRV.D.VSAM.OST -
    CLUSTER -
    PURGE -
    ERASE
DELETE -
    PKISRV.D.VSAM.ICL -
    CLUSTER -
    PURGE -
    ERASE
    IF MAXCC LT 9 THEN SET MAXCC = 0
/*
/**-----*
/** Define KSDS *
/**-----*
//DEFKSDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER -
    (NAME(PKISRV.D.VSAM.OST) -
    VOL(vvvvvv) -
    RECSZ(1024 32756) -
    INDEXED -
    NOREUSE -
    KEYS(4 0) -
    SHR(2) -
    CYL(3,1) -
    LOG(NONE) -
    OWNER(PKISRV.D) ) -
DATA -
    (NAME(PKISRV.D.VSAM.OST.DA) -
    CISZ(1024) -
    SPANNED) -
INDEX -
    (NAME(PKISRV.D.VSAM.OST.IX))

DEFINE CLUSTER -
    (NAME(PKISRV.D.VSAM.ICL) -
    VOL(vvvvvv) -
    RECSZ(1024 32756) -
    INDEXED -
    NOREUSE -
    KEYS(4 0) -
    SHR(2) -
    CYL(3,1) -
    LOG(NONE) -
    OWNER(PKISRV.D) ) -
DATA -
    (NAME(PKISRV.D.VSAM.ICL.DA) -
    CISZ(1024) -

```

## Other code samples

```

        SPANNED) -
        INDEX -
        (NAME(PKISRVD.VSAM.ICL.IX))

/*
/*-----*
/* Repro record of all binary zeros into KSDS *
/*-----*
//MKZEROS EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD *

//SYSUT2 DD DSN=&&GENTMP,UNIT=SYSDA,DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=640),SPACE=(TRK,(1,1))
//SYSIN DD *
GENERATE MAXFLDS=4,MAXLITS=80
RECORD FIELD=(20,X'0000000000000000000000000000000000000000',,1),
        FIELD=(20,X'0000000000000000000000000000000000000000',,21),
        FIELD=(20,X'0000000000000000000000000000000000000000',,41),
        FIELD=(20,X'0000000000000000000000000000000000000000',,61)

/*
//REPROKSD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSDATA DD DSN=*.MKZEROS.SYSUT2,DISP=(OLD,DELETE)
//SYSIN DD *
        REPRO INFILE(SYSDATA) -
                OUTDATASET(PKISRVD.VSAM.OST)
        REPRO INFILE(SYSDATA) -
                OUTDATASET(PKISRVD.VSAM.ICL)

/*
/*-----*
/* Define ALTERNATE INDEX and PATH *
/*-----*
//DEFALTDX EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE ALTERNATEINDEX -
                (NAME(PKISRVD.VSAM.OST.AIX) -
                RELATE(PKISRVD.VSAM.OST)-
                VOL(vvvvvv) -
                TRK(5,1) -
                KEYS(24 44) ) -
        DATA -
                (NAME(PKISRVD.VSAM.OST.AIX.DA)) -
        INDEX -
                (NAME(PKISRVD.VSAM.OST.AIX.IX))
        DEFINE PATH -
                (NAME(PKISRVD.VSAM.OST.PATH) -
                PATHENTRY(PKISRVD.VSAM.OST.AIX))
        DEFINE ALTERNATEINDEX -
                (NAME(PKISRVD.VSAM.OST.STATAIX) -
                RELATE(PKISRVD.VSAM.OST)-
                VOL(vvvvvv) -
                TRK(5,1) -
                KEYS(40 4) ) -
        DATA -
                (NAME(PKISRVD.VSAM.OST.STATAIX.DA)) -
        INDEX -
                (NAME(PKISRVD.VSAM.OST.STATAIX.IX))
        DEFINE PATH -
                (NAME(PKISRVD.VSAM.OST.STATUS) -
                PATHENTRY(PKISRVD.VSAM.OST.STATAIX))
        DEFINE ALTERNATEINDEX -
                (NAME(PKISRVD.VSAM.ICL.STATAIX) -
                RELATE(PKISRVD.VSAM.ICL)-
                VOL(vvvvvv) -
                TRK(5,1) -
                KEYS(40 4) ) -
        DATA -
                (NAME(PKISRVD.VSAM.ICL.STATAIX.DA)) -
        INDEX -
                (NAME(PKISRVD.VSAM.ICL.STATAIX.IX))
        DEFINE PATH -

```

```

        (NAME(PKISRVD.VSAM.ICL.STATUS) -
        PATHENTRY(PKISRVD.VSAM.ICL.STATAIX))
DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.OST.REQAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(32 12) ) -
DATA -
        (NAME(PKISRVD.VSAM.OST.REQAIX.DA)) -
INDEX -
        (NAME(PKISRVD.VSAM.OST.REQAIX.IX))
DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.REQUESTR) -
        PATHENTRY(PKISRVD.VSAM.OST.REQAIX))
DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.ICL.REQAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(32 12) ) -
DATA -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.DA)) -
INDEX -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.IX))
DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.REQUESTR) -
        PATHENTRY(PKISRVD.VSAM.ICL.REQAIX))

/*
/*-----*
/* BUILD ALTERNATE INDEX *
/*-----*
//BLDINDEX EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.AIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.STATAIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.STATAIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.REQAIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.REQAIX)

/*
/*-----*
/* Print out the cluster *
/*-----*
//PRTCLUST EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        PRINT -
            INDATASET(PKISRVD.VSAM.OST) CHAR
        PRINT -
            INDATASET(PKISRVD.VSAM.ICL) CHAR
/*

```

## IKYMVVSAM

IKYMVVSAM contains sample IDCAMS JCL to use if you are migrating from z/OS Version 1 Release 4 or lower. IKYMVVSAM creates additional VSAM alternate indexes and PATH data sets not created in lower releases. IKYMVVSAM is installed as a member of SYS1.SAMPLIB.

**Note:** The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYMVVSAM.

## Other code samples

```
//IKYMSVAM JOB <job card parameters>
//*****
//* SAMP:      IKYMSVAM
//*
//* Licensed Materials - Property of IBM
//* 5694-A01
//* (C) Copyright IBM Corp. 2003, 2006
//* Status = HKY7730
//*
//*****
//*
//* This sample JCL may be used to create the additional VSAM
//* alternate indexes and paths required for PKI Services V1R5.
//*
//*****
//*
//* Caution: This is neither a JCL procedure nor a complete job.
//* Before using this job step, you will have to make the following
//* modifications:
//*
//* 1) Change the job card to meet your system requirements.
//*
//* 2) If you are using VSAM record level sharing (RLS), remove
//* the VOL(vvvvvv) statements in the DEFINE step.
//*
//* If not using VSAM RLS, change all occurrences of vvvvvv to
//* the VOLSER value appropriate for the system this job is to be
//* run on.
//*
//* 3) This job assumes you are using the default VSAM data set
//* names (all have high level qualifiers "PKISRV.D.VSAM"). If
//* you have changed these data set names, you will need to
//* modify all occurrences of "PKISRV.D.VSAM" below. You will also
//* need to modify the ObjectStore DSN statements that were added
//* to the configuration file (/etc/pkiserv/pkiserv.conf).
//* If you are using multiple CA Domains, IBM recommends using
//* the first eight characters of the CA Domain as one of the
//* data set qualifiers.
//*
//* 4) If you wish to change either the primary or secondary space
//* allocation sizes for the alternate index datasets from the
//* default value, update the TRK operands on the DEFINE
//* ALTERNATEINDEX commands.
//*
//* **Note, do not change any of the numeric values other than TRK
//-----*
//* Change Activity:
//*
//* $L0=PKIS4 HKY7708 020826 PDJWS1: VSAM scaling
//* $P1=MG01346 HKY7708 021022 PDJWS1: JES3 JCL error @P1A*
//* $P2=MG01508 HKY7708 021222 PDJWS1: Prolog error @P2A*
//* $L1=PKIS7 HKY7730 050228 PDTG1: Multi-CA Support @L1A*
//*
//* Change Description:
//* C: Removed the DD statements from the BUILDID. Added VOL @P1A*
//* statements to the DEFINE step @P1A*
//* C: Updated prolog instructions for RLS @P2A*
//* C: Updated prolog with CA Domain information @L1A*
//-----*
//*
//-----*
//* DELETE THE ALTERNATE INDEXES
//-----*
//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE -
        PKISRV.D.VSAM.OST.STATAIX -
        ALTERNATEINDEX -
```



```

        PURGE ERASE
DELETE -
    PKISRV.D.VSAM.ICL.STATAIX -
    ALTERNATEINDEX -
    PURGE ERASE
DELETE -
    PKISRV.D.VSAM.OST.REQAIX -
    ALTERNATEINDEX -
    PURGE ERASE
DELETE -
    PKISRV.D.VSAM.ICL.REQAIX -
    ALTERNATEINDEX -
    PURGE ERASE
    IF MAXCC LT 9 THEN SET MAXCC = 0
/*
/*-----
/* DEFINE THE ALTERNATE INDEXES AND PATHS
/*-----
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRV.D.VSAM.OST.STATAIX) -
        RELATE(PKISRV.D.VSAM.OST)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(40 4) ) -
    DATA -
        (NAME(PKISRV.D.VSAM.OST.STATAIX.DA)) -
    INDEX -
        (NAME(PKISRV.D.VSAM.OST.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRV.D.VSAM.OST.STATUS) -
        PATHENTRY(PKISRV.D.VSAM.OST.STATAIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRV.D.VSAM.ICL.STATAIX) -
        RELATE(PKISRV.D.VSAM.ICL)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(40 4) ) -
    DATA -
        (NAME(PKISRV.D.VSAM.ICL.STATAIX.DA)) -
    INDEX -
        (NAME(PKISRV.D.VSAM.ICL.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRV.D.VSAM.ICL.STATUS) -
        PATHENTRY(PKISRV.D.VSAM.ICL.STATAIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRV.D.VSAM.OST.REQAIX) -
        RELATE(PKISRV.D.VSAM.OST)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(32 12) ) -
    DATA -
        (NAME(PKISRV.D.VSAM.OST.REQAIX.DA)) -
    INDEX -
        (NAME(PKISRV.D.VSAM.OST.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRV.D.VSAM.OST.REQUEST) -
        PATHENTRY(PKISRV.D.VSAM.OST.REQAIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRV.D.VSAM.ICL.REQAIX) -
        RELATE(PKISRV.D.VSAM.ICL)-
        VOL(vvvvvv) -
        TRK(5,1) -
        KEYS(32 12) ) -
    DATA -
        (NAME(PKISRV.D.VSAM.ICL.REQAIX.DA)) -
    INDEX -
        (NAME(PKISRV.D.VSAM.ICL.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRV.D.VSAM.ICL.REQUEST) -

```

## Other code samples

```
                PATHENTRY(PKISRVD.VSAM.ICL.REQAIX))
/*
//-----
//* BUILD ALTERNATE INDEXES
//-----
//BUILDID EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
            OUTDATASET(PKISRVD.VSAM.OST.STATAIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
            OUTDATASET(PKISRVD.VSAM.ICL.STATAIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
            OUTDATASET(PKISRVD.VSAM.OST.REQAIX)
        BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
            OUTDATASET(PKISRVD.VSAM.ICL.REQAIX)
/*
```

---

## IKYRVSAM

IKYRVSAM contains sample IDCAMS JCL to migrate the PKI Services VSAM data sets to support VSAM record-level sharing (RLS) when you intend to use Parallel Sysplex support. IKYRVSAM is installed as a member of SYS1.SAMPLIB.

Execute this job *after* executing the IKYCVSAM job. This job renames the VSAM data sets created by IKYCVSAM and copies their contents to newly allocated RLS data sets.

**Note:** The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.SAMPLIB member IKYRVSAM.

```
//IKYRVSAM JOB <job card parameters>
//*****
/* SAMP:      IKYRVSAM
/*
/* Licensed Materials - Property of IBM
/* 5694-A01
/* (C) Copyright IBM Corp. 2002, 2006
/* Status = HKY7730
/*
//*****
/*
/* This sample JCL may be used to reallocate the VSAM data sets
/* in a storage class acceptable to VSAM record level sharing (RLS).
/* This is a prerequisite to using PKI Services SYSPLEX support.
/*
//*****
/*
/* Caution: This is neither a JCL procedure nor a complete job.
/* Before using this job step, you will have to make the following
/* modifications:
/*
/* 1) Change the job card to meet your system requirements.
/*
/* 2) Change the STORCLAS statements to provide the name of the
/* storage class defined for use with VSAM RLS.
/*
/* 3) This job assumes you are using the default VSAM data set
/* names (all have high level qualifiers "PKISRVD.VSAM"). If
/* you have changed these data set names, you will need to
/* modify the source data set names in the ALTER
/* statements of the RENAMEDS step. If you are using
/* multiple CA Domains, IBM recommends using the first eight
/* characters of the CA Domain as one of the data set
/* qualifiers.
/*
//*
```

```

/** 4) This job creates destination data sets with the same default *
/** names as the source data sets. (The source data sets are *
/** renamed.) If you wish to use different destination data set *
/** names, you will need to modify the data set names in all *
/** steps except the RENAMEDS step. If you modify these names, *
/** be sure to also modify your configuration file *
/** appropriately(/etc/pkiserv/pkiserv.conf). If you are using *
/** multiple CA Domains, IBM recommends using the first eight *
/** characters of the CA Domain as one of the data set *
/** qualifiers. *
/** *
/** 5) This job renames the source data sets to begin with high *
/** level qualifiers "PKISRVD.OLDVSAM". If you wish to change *
/** these names, you will need to do so in the RENAMEDS and *
/** and REPROCL steps. If you are using multiple CA Domains, *
/** IBM recommends using the first eight characters of the CA *
/** Domain as one of the data set qualifiers. *
/** *
/** 6) If you wish to change either the primary or secondary space *
/** allocation sizes for either the OST or ICL datasets from the *
/** default value, update the CYL or TRK operands on the *
/** DEFINE CLUSTER or DEFINE ALTERNATE INDEX commands. *
/** *
/** **Note, do not change any of the numeric values other than *
/** CYL or TRK *
/**-----*
/** Change Activity: *
/** *
/** $L0=PKIS3 HKY7707 020314 PDJWS1: VSAM RLS *
/** $P1=MG00719 HKY7707 020416 PDJWS1: VSAM RLS 2 @P1A*
/** $L1=MG01176 HKY7708 020826 PDJWS1: VSAM scaling @L1A*
/** $P2=MG01346 HKY7708 021022 PDJWS1: JES3 JCL error @P2A*
/** $P3=MG01521 HKY7708 030106 PDBRW1: Incorrect source names @P3A*
/** $L2=PKIS7 HKY7730 050228 PDTCG1: Multi-CA Support @L2A*
/** *
/** Change Description: *
/** *
/** C: Removed SPANNED, CISIZE, and FILE(VOLUME) statements @P1A*
/** C: Added more alt indexes and changed allocation parms @L1A*
/** C: Removed DD statements from BLDINDEX step @P2A*
/** C: Correct souce dataset names for ICL alternate indexes @P3A*
/** C: Updated prolog with CA Domain information @L2A*
/** C: - Updated RENAMEDS step to use PKISRVD.VSAM.OST.AIX.IX *
/** dataset name and conditionally handle the former dataset *
/** name (PKISRVD.VSAM.AIX.IX). *
/** - Updated DEFALTDX step to use PKISRVD.VSAM.OST.AIX.IX *
/** dataset name instead of the former dataset name of *
/** PKISRVD.VSAM.AIX.IX. *
/** - Updated DEFKSDS to have a line continuation character *
/** after the CYL parameters. @P4A*
/**-----*
/** *
/**-----*
/** Rename source clusters, alternate indexes and PATH *
/**-----*
//RENAMEDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER -
    PKISRVD.VSAM.OST -
    NEWNAME(PKISRVD.OLDVSAM.OST)
ALTER -
    PKISRVD.VSAM.OST.* -
    NEWNAME(PKISRVD.OLDVSAM.OST.*)
ALTER -
    PKISRVD.VSAM.OST.AIX.* -
    NEWNAME(PKISRVD.OLDVSAM.OST.AIX.*)
ALTER -
    PKISRVD.VSAM.ICL -
    NEWNAME(PKISRVD.OLDVSAM.ICL)
ALTER -
    PKISRVD.VSAM.ICL.* -

```

## Other code samples

```

        NEWNAME(PKISRVD.OLDVSAM.ICL.*)
ALTER -
    PKISRVD.VSAM.OST.AIX.IX -
    NEWNAME(PKISRVD.OLDVSAM.OST.AIX.IX)
IF LASTCC EQ 8 THEN
    DO
        SET MAXCC EQ 0
        ALTER -
            PKISRVD.VSAM.AIX.IX -
            NEWNAME(PKISRVD.OLDVSAM.OST.AIX.IX)
    END
ALTER -
    PKISRVD.VSAM.OST.STATAIX.* -
    NEWNAME(PKISRVD.OLDVSAM.OST.STATAIX.*)
ALTER -
    PKISRVD.VSAM.OST.REQAIX.* -
    NEWNAME(PKISRVD.OLDVSAM.OST.REQAIX.*)
ALTER -
    PKISRVD.VSAM.ICL.STATAIX.* -
    NEWNAME(PKISRVD.OLDVSAM.ICL.STATAIX.*)
ALTER -
    PKISRVD.VSAM.ICL.REQAIX.* -
    NEWNAME(PKISRVD.OLDVSAM.ICL.REQAIX.*)
/*
/*-----*
/* Define destination Clusters                                     *
/*-----*
//DEFKSDS EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE CLUSTER -
        (NAME(PKISRVD.VSAM.OST) -
        STORCLAS(class-name) -
        RECSZ(1024 32756) -
        INDEXED -
        NOREUSE -
        KEYS(4 0) -
        SHR(2) -
        CYL(3,1) -
        LOG(NONE) -
        OWNER(PKISRVD) ) -
    DATA -
        (NAME(PKISRVD.VSAM.OST.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.OST.IX))

    DEFINE CLUSTER -
        (NAME(PKISRVD.VSAM.ICL) -
        STORCLAS(class-name) -
        RECSZ(1024 32756) -
        INDEXED -
        NOREUSE -
        KEYS(4 0) -
        SHR(2) -
        LOG(NONE) -
        CYL(3,1) -
        OWNER(PKISRVD) ) -
    DATA -
        (NAME(PKISRVD.VSAM.ICL.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.ICL.IX))
/*
/*-----*
/* Repro source cluster to destination cluster                     *
/*-----*
//REPROCL EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    REPRO INDATASET(PKISRVD.OLDVSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST)
    REPRO INDATASET(PKISRVD.OLDVSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL)

```

```

/*
/*-----*
/* Define ALTERNATE INDEX AND PATH                                     *
/*-----*
//DEFLTDX EXEC PGM=IDCAMS,COND=(8,LE)
//SYSPRINT DD  SYSOUT=*
//SYSIN      DD  *
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.OST.AIX) -
        RELATE(PKISRVD.VSAM.OST)-
        TRK(5,1) -
        KEYS(24 44) ) -
    DATA -
        (NAME(PKISRVD.VSAM.OST.AIX.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.OST.AIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.PATH) -
        PATHENTRY(PKISRVD.VSAM.OST.AIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.OST.STATAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        TRK(5,1) -
        KEYS(40 4) ) -
    DATA -
        (NAME(PKISRVD.VSAM.OST.STATAIX.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.OST.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.STATUS) -
        PATHENTRY(PKISRVD.VSAM.OST.STATAIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.ICL.STATAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        TRK(5,1) -
        KEYS(40 4) ) -
    DATA -
        (NAME(PKISRVD.VSAM.ICL.STATAIX.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.ICL.STATAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.STATUS) -
        PATHENTRY(PKISRVD.VSAM.ICL.STATAIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.OST.REQAIX) -
        RELATE(PKISRVD.VSAM.OST)-
        TRK(5,1) -
        KEYS(32 12) ) -
    DATA -
        (NAME(PKISRVD.VSAM.OST.REQAIX.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.OST.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.OST.REQUESTR) -
        PATHENTRY(PKISRVD.VSAM.OST.REQAIX))
    DEFINE ALTERNATEINDEX -
        (NAME(PKISRVD.VSAM.ICL.REQAIX) -
        RELATE(PKISRVD.VSAM.ICL)-
        TRK(5,1) -
        KEYS(32 12) ) -
    DATA -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.DA)) -
    INDEX -
        (NAME(PKISRVD.VSAM.ICL.REQAIX.IX))
    DEFINE PATH -
        (NAME(PKISRVD.VSAM.ICL.REQUESTR) -
        PATHENTRY(PKISRVD.VSAM.ICL.REQAIX))
/*
/*-----*
/* BUILD ALTERNATE INDEX                                             *
/*-----*
//BLDINDEX EXEC PGM=IDCAMS

```

## Other code samples

```
//SYSPRINT DD  SYSOUT=*
//SYSIN DD  *
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.AIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.STATAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.STATAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.OST) -
        OUTDATASET(PKISRVD.VSAM.OST.REQAIX)
    BLDINDEX INDATASET(PKISRVD.VSAM.ICL) -
        OUTDATASET(PKISRVD.VSAM.ICL.REQAIX)
/*
```

---

## PKISRVD sample procedure to start PKI Services daemon

PKISRVD is the sample procedure to start PKI Services daemon. The PKI Services daemon runs as a started task. The procedure for this can be found in 'SYS1.PROCLIB' member PKISRVD. (PKISRVD is an alias for IKYSPROC.)

PKISRVD contains the **TZ** (time zone) environment variable, which is the environment variable most likely to change. You need to specify any other environment variables that PKI Services needs in an environment variables file, by default pkiserv.envars. PKISRVD contains FN (file name) and DIR (directory) parameters, to specify the pathname of the environment variables file. You can make any needed changes in PKISRVD, such as updating this pathname.

**Guideline:** By default, the pathname for the pkiserv.envars environment variables file is /usr/lpp/pkiserv/samples/pkiserv.envars. If you need to make changes in the environment variables file, you need to copy it from the samples directory to another directory. Specify your environment variables using an environment variables file under the /etc directory, for example /etc/pkiserv/pkiserv.envars.

The following listing might not be identical to the code sample shipped with the product. For the most current sample, see SYS1.PROCLIB member PKISRVD.

```
//*****
//*
/*          Licensed Materials - Property of IBM          *
/*          5694-A01                                       *
/*          (C) Copyright IBM Corp. 2001                 *
/*          Status=HKY7706                                 *
/*
//*****
//*****
//*
/* Procedure for starting the PKI Services Daemon        *
/*
//*****
//PKISRVD PROC REGSIZE=256M,                               X
//          OUTCLASS='A',                                 X
//          TZ='EST5EDT',                                  X
//          FN='pkiserv.envars',                           X
//          DIR='/usr/lpp/pkiserv/samples',                 X
//          STD0='1>DD:STDOUT',                             X
//          STDE='2>DD:STDERR'                             X
//-----
//GO      EXEC  PGM=IKYPKID,REGION=&REGSIZE,TIME=1440,
//  PARM=('ENVAR("_CEE_ENVFILE=&DIR/&FN","TZ=&TZ") / &STD0 &STDE')
//STDOUT DD  SYSOUT=&OUTCLASS
//STDERR DD  SYSOUT=&OUTCLASS
//SYSOUT DD  SYSOUT=&OUTCLASS
//CEEDUMP DD  SYSOUT=&OUTCLASS
```

## Chapter 27. SMF recording

PKI Services produces one SMF record type—type 80. The first 18 bytes of type 80 records represent the standard SMF header without subtypes.

### For more information:

1. See *z/OS MVS System Management Facilities (SMF)* for information about how to use SMF.
2. See *z/OS Security Server RACF Macros and Interfaces* and *z/OS Security Server RACF Auditor's Guide* for information about using the RACF SMF data unload utility (IRRADU00) to prepare reports with the RACF report writer.

## PKI Services event code

Table 80 describes the SMF80EVT (event code) and SMF80EVQ (event code qualifier) fields for the PKI Services event code. It also lists the SMF80DTP and SMF80DA2 values for the relocate type sections.

Table 80. SMF event code and event code qualifier for PKI Services

Event Dec(Hex)	Command	Code qualifier Dec(Hex)	Description	Relocate type sections
79(4F)	CRL publication	0(0)	Successful publication of revocation information	318, 319, 366, 379, 380, 381, 382, 383, 384, 385, 387

## Relocate section variable data

Table 81 describes the variable data elements of the extended-length relocate section.

Table 81. SMF data elements of the extended-length relocate section for PKI Services

Data type (SMF80TP2) Dec(Hex)	Data length (SMF80DL2)	Format	Audited by event code	Description (SMF80DA2)
318(13E)	1–255	EBCDIC	66, 67, 69, 72, 74, 79	Certificate or CRL serial number
319(13F)	1–255	EBCDIC	66, 67, 69, 72, 74, 79	Certificate or CRL issuer's distinguished name
366(16E)	4	Binary	74	Certificate revocation reason
379(17B)	1–255	EBCDIC	79	CRL issuing distribution point DN
380(17C)	10	EBCDIC	79	CRL's date of issue
381(17D)	8	EBCDIC	79	CRL's time of issue
382(17E)	10	EBCDIC	79	CRL's expiration date
383(17F)	8	EBCDIC	79	CRL's expiration time
384(180)	10	EBCDIC	79	CRL's date of publish
385(181)	8	EBCDIC	79	CRL's time of publish
387(183)	1–1024	EBCDIC	79	CRL's issuing distribution point URI





## Appendix A. LDAP directory server requirements

PKI Services typically requires access to an LDAP directory server to store issued certificates and certificate revocation lists. The z/OS LDAP server is preferred but not required. You can use a non-z/OS LDAP server if it can support the objectclasses and attributes PKI Services uses. These are listed in the following table:

Table 82. LDAP objectclasses and attributes that PKI Services sets

End-entity or branch node?	Visible RDN attribute	Objectclasses used	Additional attributes set (other than visible RDN attribute)
Creating a branch node	C=	country	—
Creating a branch node	L=	locality	—
Creating a branch node	O=	organization	—
Creating a branch node	OU=	organizationalUnit	—
Creating a branch node	Any supported value other than the preceding	organizationalUnit, and extensibleObject	ou (the ou value from CreateOUValue in the <b>LDAP</b> section of pkiserv.conf file)
Creating a user end-entity   	unstructuredName or unstructAddress	account, pkiUser, cEPDevice, and extensibleObject	userCertificate, and uid (hardcoded to NoUid)
Creating a user end-entity   	serialNumber	account, pkiUser, pKCS10Device, and extensibleObject	userCertificate, and uid (hardcoded to NoUid)
Creating a user end-entity       	Any supported value other than unstructuredName, unstructAddress, and serialNumber	account, pkiUser, and extensibleObject	userCertificate, and uid (hardcoded to NoUid)
Creating a CA end-entity	O=	organization, and pkiCA	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a CA end-entity	OU=	organizationalUnit, and pkiCA	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a CA end-entity	Any supported value other than O or OU	account, pkiCA, and extensibleObject	cACertificate, certificaterevocationlist, authorityrevocationlist and uid (hardcoded to NoUid)
User end-entity that   already exists	unstructuredName or unstructAddress	pkiUser, cEPDevice	userCertificate
User end-entity that   already exists	serialNumber	pkiUser, pKCS10Device	userCertificate
User end-entity that   already exists     	Any supported value other than unstructuredName, unstructAddress, and serialNumber	pkiUser	userCertificate

## LDAP directory server requirements

Table 82. LDAP objectclasses and attributes that PKI Services sets (continued)

End-entity or branch node?	Visible RDN attribute	Objectclasses used	Additional attributes set (other than visible RDN attribute)
CA end-entity that already exists	Any supported value	pkiCA	cACertificate, certificaterevocationlist, and authorityrevocationlist
Creating a distribution point CRL end-entity	CN=	commonName and cRLDistributionPoint	certificateRevocationList
Distribution point CRL end-entity that already exists	Any supported value	cRLDistributionPoint	certificateRevocationList

The R\_PKIServ SAF callable service supports specifying the subject's DN through named fields in the CertPlist. The CGIs invoke the R\_PKIServ SAF callable service. For more information, see *z/OS Security Server RACF Callable Services*. PKI Services supports the subject's DN fields, plus some additional ones: postal code, street, and mail. They are mapped to LDAP attributes as Table 83 indicates.

Table 83. Relationship of named fields to LDAP attributes and object identifiers

Named field	Visible RDN attribute	OID
CommonName	CN	2.5.4.3
Title	TITLE	2.5.4.12
OrgUnit	OU	2.5.4.11
Org	O	2.5.4.10
Locality	L	2.5.4.7
StateProv	ST	2.5.4.8
Country	C	2.5.4.6
PostalCode	POSTALCODE	2.5.4.17
Street	STREET	2.5.4.9
Email <sup>1</sup>	MAIL	0.9.2342.19200300.100.1.3
Mail	MAIL <sup>2</sup>	0.9.2342.19200300.100.1.3
EmailAddr	EMAIL	1.2.840.113549.1.9.1
UnstructName	UNSTRUCTUREDNAME	1.2.840.113549.1.9.2
UnstructAddr	UNSTRUCTUREDADDRESS	1.2.840.113549.1.9.8
SerialNumber	SERIALNUMBER	2.5.4.5

<sup>1</sup> The use of the field name Email is deprecated; use Mail instead.

<sup>2</sup> When a certificate is created and posted to LDAP, the NotifyEmail value, if specified, is posted as the MAIL attribute. (This replaces any MAIL attribute for the directory entry and for certificate renewals replaces the original NotifyEmail value).

---

## Appendix B. Using a gskkyman key database for your certificate store

This appendix lists the steps the RACF programmer performs to use a gskkyman key database.

---

### Steps for using a gskkyman key database for your certificate store

Perform the following steps to use a gskkyman key database for your server's certificate store:

**Note:** If the z/OS HTTP Server is installed and configured for SSL using gskkyman, you need to perform only steps 9, 10, 11, and 15.

1. From the UNIX shell, **cd** to /etc and enter /usr/lpp/gskssl/bin/gskkyman.

- 
2. Choose option **1** to create a key database. Type in a name or let it default to key .kdb and enter a password you want to use. When asked "Work with the database now?", enter **1** for yes.

- 
3. Choose option **3** to create new key pair and certificate request. Answer the prompts for file name, label, key size (1024 is suggested), and subject name fields.

**Note:** Common Name should be your server's symbolic IP address (for example, *www.YourCompany.com*).

- 
4. Exit gskkyman when you are done.

- 
5. From TSO, use the OGET command to put the certificate request in an MVS data set.

**Example:**

```
OGET '/etc/certreq.arm' certreq.arm
```

- 
6. Use the RACDCERT command to read the request and generate the server certificate.

**Example:**

```
RACDCERT GENCERT(certreq.arm) ID(WEBSRV) SIGNWITH(CERTAUTH  
LABEL('Local PKI CA')) WITHLABEL('SSL Cert')
```

- 
7. Export both the new server certificate and the CA certificate to MVS data sets, and OPUT these to file system files.

**Example:**

```
RACDCERT EXPORT(LABEL('SSL Cert')) ID(WEBSRV) DSN(cert.arm)  
FORMAT(CERTB64)  
OPUT cacert.der '/var/pkiserv/cacert.der' BINARY
```

---

## Using gskkyman

8. You can optionally delete both certificate TSO data sets (but not the file system files).

---
9. In the UNIX shell, **cd** to `/etc` and invoke `/usr/lpp/gskssl/bin/gskkyman`.

---
10. Choose option **2** to open the key database (created earlier). Reply to the name and password prompts.

---
11. Choose option **6** to store a CA certificate and specify the `'/var/pkiserv/cacert.der'` file.

---
12. When asked to “Exit gskkyman?”, enter **0** for No.

---
13. Choose option **4** to receive a certificate issued for your request and specify the `'/etc/cert.arm'` file. Again enter 0 when asked to “Exit gskkyman?”.

---
14. Choose option **11** to store encrypted database password.

---
15. Exit gskkyman.

---
16. You can optionally remove the `/etc/cert.arm` file.

---

---

## Appendix C. Configuring PKI Services as an IdenTrust™ certificate authority

This appendix describes the configuration required to allow your z/OS installations to participate in the IdenTrust™ infrastructure. By following the task described here, you may configure z/OS Cryptographic Services PKI Services to operate as an IdenTrust compliant certificate authority (CA). This allows you to use z/OS to manage the life cycle of digital certificates on behalf of your organization and in accordance with your security policy. You can then issue and revoke digital certificates as needed.

This appendix contains an overview, a task roadmap, a set of procedures you need to perform, and a set of code samples that you can customize as you complete the procedures.

z/OS Cryptographic Services PKI Services is the component of the IBM @server zSeries operating system z/OS that provides support for the Public Key Infrastructure (PKI) infrastructure. Beginning with z/OS Version 1 Release 5, PKI Services was certified at the IdenTrust 3.1 specification level as an IdenTrust compliant CA software program.

For more overview information about z/OS support for IdenTrust, see <http://www.ibm.com/servers/eserver/zseries/security/identrus/>.

---

### Who should use this appendix

This appendix should be used by personnel who support member institutions of the IdenTrust network. It instructs the UNIX programmer or Web server programmer to configure PKI Services to operate as an IdenTrust compliant CA.

This appendix assumes that you have experience installing and configuring software in a network environment. You should be knowledgeable about PKI technology. Previous experience with PKI Services is helpful.

---

### Related information from IdenTrust

You should be familiar with the certificate profiles defined by IdenTrust in following IdenTrust document. As an IdenTrust compliant CA using z/OS, you will implement these policies using PKI Services.

- *Identrus Public Key Infrastructure and Certificate Profiles [IT-PKI]*, Operating Rules and System Documentation.

---

### Overview of configuring z/OS PKI Services as a CA

This appendix describes the configuration required operate z/OS Cryptographic Services PKI Services as an IdenTrust compliant certificate authority (CA). It instructs member institutions of the IdenTrust network to configure their z/OS installations to participate in the IdenTrust infrastructure.

By following the task outlined in this appendix, you may use your z/OS systems to establish CAs within the IdenTrust infrastructure. This allows you to use z/OS to manage the life cycle of digital certificates on behalf of your organization in

accordance with your security policy, while issuing and revoking digital certificates as needed. See <http://www.ibm.com/servers/eserver/zseries/security/identrus/> for additional overview information.

## System prerequisites

To use your z/OS system to establish a CA within the IdenTrust infrastructure, the following requirements apply:

1. You must operate PKI Services with z/OS Version 1 Release 5, or higher.
2. Your PKI Services installation must be completed, with PKI Services established as a certificate authority (CA) and configured as a self-signed CA. (See “Configuring z/OS PKI Services as a CA” on page 393.)
3. You must supply your own (or OEM) online certificate status protocol (OCSP) responder.

## Task overview

The topic “Configuring z/OS PKI Services as a CA” on page 393 leads you through subtasks and associated procedures needed to configure your z/OS PKI Services system as a CA. The procedures configure PKI Services to accomplish the following objectives:

- Establish PKI Services as an intermediate CA under the IdenTrust root
- Adjust your PKI Services general settings.
- Define PKI Services certificate templates for IdenTrust certificate types

### Establish PKI Services as an intermediate CA under the IdenTrust root

In the IdenTrust infrastructure, IdenTrust operates the *root* or top certificate authority (CA). All financial institutions are subordinate CAs that are certified by the root. Therefore, you must acquire a certificate for PKI Services that is signed by IdenTrust. You may need additional certificates issued by IdenTrust, such as for your OCSP responder. Request these certificates directly from IdenTrust following the operational requirements of IdenTrust.

### Adjust your PKI Services general settings

Some of your current PKI Services configuration settings may require adjusting to operate PKI Services within the parameters identified by IdenTrust. In particular, your settings related to the following certificate policy items:

- CRL processing time
- Distribution point CRLs

**CRL processing time:** Processing time for an authenticated revocation request must not exceed 60 minutes. (Your OCSP responder usually provides revocation information by reading the CRLs issued by PKI Services to determine revocation status.) Therefore, the CRLs in LDAP must be refreshed with sufficient time to meet the 60-minute window.

**Applicable PKI Services settings:** TimeBetweenCRLs and CRLDuration.

**Distribution point CRLs:** Depending on how your OCSP responder operates, you may have to disable creation of distribution point CRLs.

**Applicable PKI Services setting:** CRLDistSize.

## Define PKI Services certificate templates for IdenTrust certificate types

The IdenTrust document *Identrus Public Key Infrastructure and Certificate Profiles [IT-PKI]* identifies and details the various certificate types used within the IdenTrust infrastructure. Using PKI Services, you create your IdenTrust compliant certificates. You decide which IdenTrust certificate types your CA will create and then define certificate templates for them. You may use the sample certificate templates provided in this appendix to create your IdenTrust compliant certificates.

IdenTrust compliant certificates must include the following:

- IdenTrust certificate policy information (needed to build the CertificatePolicies extension)
- The location of your OCSP responder (needed to build the AuthorityInformationAccess extension)
- Other IdenTrust required fields and extensions.

The following sample PKI Services certificate templates are provided to help you create IdenTrust compliant certificates:

- “Sample browser certificate template for IdenTrust compliance” on page 397
- “Sample server certificate template for IdenTrust compliance” on page 400.

---

## Configuring z/OS PKI Services as a CA

The following table contains a task roadmap to lead you through the subtasks and associated procedures to configure your z/OS PKI Services system as a CA. Where needed, some notes are included to provide reminders about additional activities that are not described in this document. (For background information on the subtasks and reasons they are required, see “Task overview” on page 392.)

The following procedures are provided in this appendix:

- “Steps to modify pkiserv.conf for different certificate types” on page 394
- “Steps to modify pkiserv.conf general settings” on page 395
- “Steps to create IdenTrust specific certificate templates” on page 395.

Subtask	Associated instructions (see ...)	Notes
If you have not already done so, install and configure PKI Services as a self-signed certificate authority (CA).	Follow the instructions in Part 1, "Planning," Part 2, "Configuring your system for PKI Services," and Part 3, "Customizing PKI Services."	Remember to store your PKI Services CA signing key in Integrated Cryptographic Service Facility (ICSF).
Establish PKI Services as an intermediate certificate authority under the IdenTrust root.	Follow the instructions in "Establishing PKI Services as an intermediate CA" on page 246 in Part 5, "Administering security for PKI Services."	Send your certificate request to IdenTrust for signing. To do this, follow the IdenTrust instructions in <i>IT-PKI</i> to request a certificate for a "Participant CA Key Signing and CRL Signing Certificate Profile".
Modify the PKI Services configuration file (pkiserv.conf).	"Steps to modify pkiserv.conf for different certificate types"	Make sure your certificate policies are in accordance with <i>IT-PKI</i> .
	"Steps to modify pkiserv.conf general settings" on page 395	Make sure your changes are in accordance with <i>IT-PKI</i> .
Create IdenTrust specific certificate templates in the PKI Services certificate templates file (pkiserv.tpl).	"Steps to create IdenTrust specific certificate templates" on page 395	Make sure your certificate templates are in accordance with <i>IT-PKI</i> .
Stop and restart PKI Services to activate your changes.	Follow the instructions in Chapter 10, "Starting and stopping PKI Services," on page 85.	

## Steps to modify pkiserv.conf for different certificate types

**Before you begin:** Refer to the sample configuration file directives in "Sample PKI Services configuration file directives for IdenTrust compliance" on page 397.

Perform the following steps to modify the PKI Services configuration file (pkiserv.conf) to add a certificate policy for each type of IdenTrust certificate you intend to issue:

1. Copy the sample **OIDs** directives to the **OIDs** section.
2. Copy the sample **CertPolicy** directives to the **CertPolicy** section.
3. For each IdenTrust certificate policy you add, replicate one of the **OIDs** directives copied in Step 1.
4. Change the name and value of the directive as needed for the particular certificate profile. The name you choose is arbitrary, but must be unique.
5. Replicate one pair of PolicyName $nn$  and UserNoticeText $nn$  (the **CertPolicy** directives) copied in Step 2.



- 
6. Change the value of the `PolicyName $nn$`  directive to match the name defined in Step 4 on page 394.

---

  7. Change the value of the `UserNoticeText $nn$`  directive as needed for this policy.

---

  8. Change the policy number in the directives' name (the  $nn$  in `PolicyName $nn$`  and `UserNoticeText $nn$` ) as needed for the particular policy being defined. The number you choose is arbitrary, but must be unique. Use the same number for both directives.

---

  9. For each `IdenTrust` certificate policy you add, repeat Step 3 on page 394 through Step 8.
- 

**When you are done:** You have defined a certificate policy for each type of `IdenTrust` certificate you intend to issue.

## Steps to modify `pkiserv.conf` general settings

### Before you begin:

- Refer to the sample configuration file directives in “Sample PKI Services configuration file directives for `IdenTrust` compliance” on page 397.
- For more information about creating directives for certificate policies, see “Using certificate policies” on page 146 in Part 3, “Customizing PKI Services.”

Perform the following steps to modify the PKI Services configuration file (`pkiserv.conf`) to configure certain general settings for `IdenTrust` compliance:

1. Change your current setting to `TimeBetweenCRLs=30m`.

---

  2. Change your current setting to `CRLDuration=60m`.

---

  3. Optionally, change your current setting to `CRLDistSize=0`. This change is required only if your OCSP responder does not support distribution point CRLs.
- 

**When you are done:** You have configured your general PKI Services settings for `IdenTrust` compliance.

## Steps to create `IdenTrust` specific certificate templates

### Before you begin:

- Refer to the sample browser certificate template in “Sample browser certificate template for `IdenTrust` compliance” on page 397 and the sample server certificate template in “Sample server certificate template for `IdenTrust` compliance” on page 400.
- For details about creating certificate templates, see Chapter 11, “Customizing the end-user Web application,” on page 91.

For each IdenTrust certificate profile you need, perform the following steps to create an IdenTrust specific certificate template in the PKI Services certificate templates file (`pkiserv.tmpl`):

1. Determine if you need to define a certificate profile for a browser certificate or a server certificate.

---
2. Copy the appropriate sample browser or server certificate template to the PKI Services certificate templates file.

---
3. Change the name of the template as desired.

---
4. Change the nickname of the template as desired.

---
5. Change the <CONTENT> section to add or remove name fields and matching JavaScript as required for the desired IdenTrust profile. For example, if the subject's alternate name e-mail address is not required, remove it or make it optional.

---
6. Change the <CONSTANT> section as follows:
  - a. Change the `AuthInfoAcc` values to provide the URLs required by your OCSP responder.
  - b. Change the `CertPolicies` value to provide the policy numbers needed for the desired IdenTrust profile. (See Step 3.)

---

**When you are done:** You have created an IdenTrust specific certificate template for each IdenTrust certificate profile you need. Stop and restart PKI Services to activate all of your changes, according to the task roadmap on page 394.

---

## Code samples

This topic contains code samples to help you complete the task of configuring PKI Services as a CA.

- “Sample PKI Services configuration file directives for IdenTrust compliance” on page 397

This sample from the PKI Services configuration file (`pkiserv.conf`) contains sample file directives that are IdenTrust compliant. The sample defines two IdenTrust policies (`IdentrusPolicy4` and `IdentrusPolicy16`). It also shows sample general settings that meet the requirements for IdenTrust compliance (`TimeBetweenCRLs` and `CRLDuration`), and an optional setting (`CRLDistSize`).

- “Sample browser certificate template for IdenTrust compliance” on page 397

This sample from the PKI Services template file (`pkiserv.tmpl`) defines a certificate template for an IdenTrust compliant browser certificate.

- “Sample server certificate template for IdenTrust compliance” on page 400

This sample from the PKI Services template file (`pkiserv.tmpl`) defines a certificate template for an IdenTrust compliant server certificate.

## Sample PKI Services configuration file directives for IdenTrust compliance

```
[OIDs]
IdentrusPolicy4=1.2.840.114021.1.4.1
IdentrusPolicy16=1.2.840.114021.1.16.2

[CertPolicy]
PolicyName4=IdentrusPolicy4
UserNoticeText4=This certificate may be relied upon only by either: (1) a
Relying Customer of an IdenTrust Participant, or (2) a party bound to the
alternative policy regime specified elsewhere in this Certificate

PolicyName16=IdentrusPolicy16
UserNoticeText16=This certificate may be relied upon only by either: (1) a
Relying Customer of an IdenTrust Participant, or (2) a party bound to the
alternative policy regime specified elsewhere in this Certificate

TimeBetweenCRLs=30m
CRLDuration=60m
CRLDistSize=0
```

## Sample browser certificate template for IdenTrust compliance

```
#
# =====
#
# Template Name - 4-Year IdenTrust EE Identity Software Consumer Type 2 Certificate
#
# Function - Creates a 4-year browser certificate for use within
#            the IdenTrust infrastructure. This certificate is used
#            to sign communications between the Subscribing Customer (SC)
#            and Relying Customer (RC) and for S/MIME Digital Signature.
#
# =====
#
<TEMPLATE NAME=4-Year IdenTrust EE Identity Software Consumer Type 2
Certificate>
<TEMPLATE NAME=PKI Browser Certificate>
<NICKNAME=4YIEEIC2>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
%%-AdditionalHead[browsertype]%%
</HEAD>

<BODY>
<H1>4-Year IdenTrust EE Identity Software Consumer Type 2 Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#      "[application]/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#      "[application]/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
<FORM NAME="CertReq" METHOD=POST ACTION=
```

```

        "[application]/ssl-cgi-bin/careq.rexx" onSubmit=
"return ValidateEntry(this)">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript -----
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
    if (ValidRequestor(frm) &&
        ValidCommonName(frm) &&
        ValidOrgUnit(frm) &&
        ValidOrg(frm) &&
        ValidCountry(frm) &&
        ValidAltEmail(frm) &&
        ValidNotifyEmail(frm) &&
        ValidPassPhrase(frm) &&
        ValidPublicKey(frm)){
# Add your validation Javascript here if needed ---
        return true;
    }
    else
        return false;
    }
//-->
</SCRIPT>
    %%Requestor (optional)%%
    %%CommonName%%
    %%OrgUnit (optional)%%
    %%Org (optional)%%
    %%Country (optional)%%
    %%AltEmail%%
    %%NotifyEmail (optional)%%
    %%PassPhrase%%
    %%PublicKey[browsertype]%%
#-- End user input fields and validation Javascript -----
<p>
<INPUT TYPE="Submit" VALUE="Submit certificate request">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>
<FORM METHOD=GET ACTION="[application]/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</u>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
    %%NotBefore=0%%
    %%NotAfter=1461%%
    %%KeyUsage=digitalsig%%
    %%KeyUsage=docsign%%
    %%CertPolicies=16%%
    %%AuthInfoAcc=OCSP,URL=https://ocsp.bank1.com
    %%AuthInfoAcc=IdentrusOCSP,URL=https://tc.bank1.com
    %%SignWith=PKI:%%
</CONSTANT>
<SUCCESSCONTENT>
    %%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
    %%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
%%-copyright%%
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>

```

```

</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "[application]/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "[application]/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
#    "[application]/ssl-cgi-bin/cagetcert.rexx" onSubmit=
#    "return ValidateEntry(this)">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
#-- User input fields and validation Javascript -----
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
if (ValidTransactionId(frm) &&
    ValidChallengePassPhrase(frm)) {
# Add your own Javascript here if needed
return true;
}
else
return false;
}
//-->
</SCRIPT>
%%-TransactionId%%
%%ChallengePassPhrase (optional)%%
#-- End user input fields and validation Javascript -----
<p>
<INPUT TYPE="submit" VALUE="Retrieve and Install Certificate">
</FORM>
<p>
<H2>To check that your certificate installed properly, follow the
procedure below:</h2>
<p><B>Netscape V6</B> - Click Edit->Preferences, then Privacy and Security->
Certificates. Click the Manage Certificates button to start the Certificate
Manager.
Your new certificate should appear in the Your Certificates list.
Select it then click View to see more information.
<p><B>Netscape V4</B> - Click the Security button, then Certificates->
Yours. Your certificate should appear in the list. Select it then
click Verify.
<p><B>Internet Explorer V5</B> - Click Tools->Internet Options, then
Content, Certificates.
Your certificate should appear in the Personal list. Click Advanced to
see additional information.
<p>
<FORM METHOD=GET ACTION="[application]/public-cgi/camain.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Home page">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>

```

```

<RETURNCERT>
%%returnbrowsercert[browsertype]%%
</RETURNCERT>
</TEMPLATE>

```

## Sample server certificate template for IdenTrust compliance

```

#
# =====
#
# Template Name - 4-Year IdenTrust EE Server Signing Certificate
#
# Function - Creates a 4-year server certificate for use within the IdenTrust
#            infrastructure. This certificate is used to sign communications
#            1) between the Subscribing Customer (SC) and Relying Customer (RC),
#            2) between the RC and the Relying Participant (RP), and
#            3) for S/MIME Digital Signature.
#
# =====
#
<TEMPLATE NAME=4-Year IdenTrust EE Server Signing Certificate>
<TEMPLATE NAME=PKI Server Certificate>
<NICKNAME=4YIEESS>
<CONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 2</TITLE>
%%-copyright%%
</HEAD>
<BODY>
<H1>4-Year IdenTrust EE Server Signing Certificate</H1>
<p>
<H2>Choose one of the following:</H2>
<p>
<ul>
<h3><li>Request a New Certificate</h3>
# This ACTION forces userid/pw authentication and runs the task under
# the client's ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "[application]/ssl-cgi-bin/auth/careq.rexx" onSubmit=

# This ACTION forces userid/pw authentication but runs the task under
# the surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "[application]/ssl-cgi-bin/surrogateauth/careq.rexx" onSubmit=

# This ACTION is for non z/OS clients. The task runs under the
# surrogate ID
#<FORM NAME="CertReq" METHOD=POST ACTION=
#           "[application]/ssl-cgi-bin/careq.rexx" onSubmit=
#           "return ValidateEntry(this)">

<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript -----
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
    if (ValidRequestor(frm) &&
        ValidCommonName(frm) &&
        ValidOrgUnit(frm) &&
        ValidOrg(frm) &&
        ValidCountry(frm) &&
        ValidAltEmail(frm) &&
        ValidNotifyEmail(frm) &&
        ValidPassPhrase(frm) &&
        ValidPublicKey(frm)){
# Add your validation Javascript here if needed ---
        return true;
    }
    else
        return false;
}

```

```

}
//-->
</SCRIPT>
%%Requestor (Optional)%%
%%CommonName%%
%%OrgUnit%%
%%Org%%
%%Country%%
%%AltEmail (Optional)%%
%%NotifyEmail (Optional)%%
%%PassPhrase%%
%%PublicKey%%
#-- End user input fields and validation Javascript -----
<p>
<INPUT TYPE="submit" VALUE="Submit certificate request">
<INPUT TYPE="reset" VALUE="Clear">
</FORM>
<p>
<H3><li>Pick Up a Previously Issued Certificate</H3>

<FORM METHOD=GET ACTION="/[application]/ssl-cgi/caretrieve.rexx">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<INPUT TYPE="submit" VALUE="Retrieve your certificate">
</FORM>
</u>
<p>%%-pagefooter%%
</BODY>
</HTML>
</CONTENT>
<CONSTANT>
%%NotBefore=0%%
%%NotAfter=1461%%
%%KeyUsage=digitalsig%%
%%KeyUsage=docsign%%
%%CertPolicies=4%%
%%AuthInfoAcc=OCSP,URL=https://ocsp.bank1.com
%%AuthInfoAcc=IdentrusOCSP,URL=https://tc.bank1.com
%%SignWith=PKI:%%
</CONSTANT>
<SUCCESSCONTENT>
%%-requestok%%
</SUCCESSCONTENT>
<FAILURECONTENT>
%%-requestbad%%
</FAILURECONTENT>

<RETRIEVECONTENT>
<HTML><HEAD>
<TITLE> Web Based PKIX Certificate Generation Application Pg 3</TITLE>
%%-copyright%%
</HEAD>

<BODY>
<H1> Retrieve Your [tmplname]</H1>
<H3>Please bookmark this page</h3>
<p>Since your certificate may not have been issued yet, we recommend
that you create a bookmark to this location so that when you return to
this bookmark, the browser will display your transaction ID.
This is the easiest way to check your status.

# This ACTION forces userid/pw authentication and runs the task
# under the client's ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
# "[application]/ssl-cgi-bin/auth/cagetcert.rexx" onSubmit=
#
# This ACTION forces userid/pw authentication but runs the task
# under the surrogate ID
#<FORM NAME=retrieveform METHOD=POST ACTION=
# "[application]/ssl-cgi-bin/surrogateauth/cagetcert.rexx" onSubmit=
#
# This ACTION is for non z/OS clients. The task runs under surrogate ID
<FORM NAME=retrieveform METHOD=POST ACTION=

```

```

        "[application]/ssl-cgi-bin/cagetcert.rexx" onSubmit=
        "return ValidateEntry(this)">
<INPUT NAME="Template" TYPE="hidden" VALUE="[tmplname]">
<p> Enter values for the following field(s)
#-- User input fields and validation Javascript -----
<SCRIPT LANGUAGE="JavaScript">
<!--
function ValidateEntry(frm){
if (ValidTransactionId(frm) &&
    ValidChallengePassPhrase(frm)) {
# Add your own Javascript here if needed
    return true;
}
else
    return false;
}
//-->
</SCRIPT>
%%-TransactionId%%
%%ChallengePassPhrase (optional)%%
#-- End user input fields and validation Javascript -----
<p>
<INPUT TYPE="submit" VALUE="Continue">
</FORM>
<p>%%-pagefooter%%
</BODY>
</HTML>
</RETRIEVECONTENT>
<RETURNCERT>
%%-returnpkcs10cert%%
</RETURNCERT>
</TEMPLATE>

```



---

## Appendix D. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

---

### Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

---

### Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

---

### z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

[www.ibm.com/servers/eserver/zseries/zos/bkserv/](http://www.ibm.com/servers/eserver/zseries/zos/bkserv/)



---

## Notices

This information was developed for products and services offered in the USA.

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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, 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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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  
Mail Station P300  
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 information 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

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.

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 illustrates 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. 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 IBM's application programming interfaces.

---

## Programming interface information

This document primarily documents information that is NOT intended to be used as Programming Interfaces of PKI Services.

This document also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of PKI Services. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

**Programming Interface information**

**End of Programming Interface information**

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

AIX  
BookManager  
CICS  
DB2  
DFSMS  
DFSMSdfp  
eServer  
IBM  
IBMLink  
Language Environment  
Lotus  
Lotus Notes  
MVS  
NetView  
Notes  
OS/390  
Parallel Sysplex  
RACF  
Redbooks  
Resource Link  
S/390  
SecureWay  
Softcopy Reader  
Tivoli  
VM/ESA  
z/OS  
z/VM  
zSeries

IdenTrust and IdenTrust are trademarks or registered trademarks of IdenTrust Inc. in the United States, other countries, or both.

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Java, JavaScript, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Active Directory, ActiveX, Authenticode, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

## Bibliography

The following lists the titles and numbers of documents referenced in this publication.

- *IBM Health Checker for z/OS: User's Guide*, SA22-7994
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521
- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SA22-7522
- *z/OS Cryptographic Services ICSF System Programmer's Guide*, SA22-7520
- *z/OS Introduction and Release Guide*, GA22-7502
- *z/OS Cryptographic Service System Secure Sockets Layer Programming*, SC24-5901
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394
- *z/OS DFSMS Introduction*, SC26-7397
- *z/OS DFSMS Storage Administration Reference*, SC26-7402
- *z/OS HTTP Server Planning, Installing, and Using*, SC34-4826
- *z/OS Information Roadmap*, SA22-7500
- *z/OS Integrated Security Services LDAP Client Programming*, SC24-5924
- *z/OS Integrated Security Services LDAP Server Administration and Use*, SC24-5923
- *z/OS Migration*, GA22-7499
- *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*, SA22-7612
- *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617
- *z/OS Open Cryptographic Services Facility Application Programming*, SC24-5899
- *z/OS SDSF Operation and Customization*, SA22-7670
- *z/OS SecureWay Security Server Open Cryptographic Enhanced Plug-ins Application Programming*, SC24-5925
- *z/OS Security Server RACF Callable Services*, SA22-7691
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683
- *z/OS Summary of Message and Interface Changes*, SA22-7505
- *z/OS TSO/E REXX Reference*, SA22-7790
- *z/OS UNIX System Services Command Reference*, SA22-7802
- *z/OS UNIX System Services Planning*, GA22-7800





---

# Index

## Special characters

- \_CEE\_RUNOPTS, updating 270
- \_PKISERV\_CA\_DOMAIN environment variable 347
- \_PKISERV\_EXIT
  - adding to environment variables file 181
  - description 347
- \_PKISERV\_MSG\_LEVEL
  - description 348
  - message levels 299
  - subcomponents 299
- \_PKISERV\_MSG\_LOGGING
  - STDERR\_LOGGING 348
  - STDOUT\_LOGGING 348
- AdditionalHeadIE 94
- preregok 94
- renewrevokebad 94
- renewrevokeok 94
- requestbad 94
- requestok 94
- return10cert 94
- /etc/pkiserv 9
- /usr/lpp/pkiserv 9
  - description 9
  - subdirectories 341
- /var/pkiserv 9
  - description 9
- [ ] for substitution variables 91
- % in named fields 93
- %%-renewrevokebad%% 101
- %%-renewrevokeok%% 101
- %%-requestok%% 108
- %%AltOther\_1\_2\_3\_4\_5%% 138
- %%AltOther\_1\_2\_3\_4\_6%% 138
- %%dn%% 136
- %%notafter%% 136
- %%requestor%% 136
- %%SelectCADomain%% 101
- %%transactionid%% 136
- \_PKISERV\_CONFIG\_PATH environment variable 347
- n-year PKI extensions demonstration certificate fields 110

## Numerics

- 1YBSM 103
- 1YBSSL 103
- 2YBZOS 103
- 2YIACS 103
- 5YSCA 103
- 5YSCEPP 103
- 5YSIPS 103
- 5YSSSL 103

## A

- abends, recording 289

- access
  - READ, authorizing 242
  - required for administrator 352, 353
  - to administration pages 217
    - changing 144
  - to end-user Web pages 198
  - to OCSF data sets 42
  - to RACF group 27
  - to VSAM data sets 27
- access control, setting up 27, 351
- accessibility 403
- accessing
  - administration home page 217
  - end-user Web pages 198
- actions
  - on certificate requests 221
  - on certificates 232
- activating KEYSMSTR class profile 261
- active (status of certificate) 231
- adding
  - alternate indexes 80
  - application domain 158
  - CA domains 161
  - certificate template 132
  - members to group 241
  - VSAM alternate indexes 80
- addtype directive 69
- admactcert.rexx 142
- admactid.rexx 142
- admactid2.rexx 142
- admicl.rexx 141
- admiclall.rexx 142
- admiclcert.rexx 141
- ADMINAPPROVE subsection (in TEMPLATE section of pkiserv.tmpl) 106
- adminDN keyword 65, 68, 73
- ADMINFOOTER subsection (in APPLICATION section of pkiserv.tmpl) 100, 143
- ADMINHEADER subsection (in APPLICATION section of pkiserv.tmpl) 100, 143
- administering
  - HostIdMappings extension 242, 243
  - PKI Services 217
  - VSAM record level sharing 79
- administration
  - changing log options 299
  - deleting certificate requests 226
  - deleting certificates 233
  - deleting preregistered requests 226
  - displaying log options settings 300
  - log options
    - changing 299
    - displaying 300
  - modifying certificate requests 224
  - processing
    - certificate requests using searches 227
    - certificates using searches 234
    - multiple certificate requests 228

- administration (*continued*)
  - processing (*continued*)
    - multiple certificates 234
    - selected certificate requests 229
    - selected certificates 236
    - single certificate 232
    - single certificate request 222
    - single preregistered request 222
  - RACF
    - ongoing administration 241
    - running IKYSETUP 27
  - rejecting certificate requests 226
  - resuming certificates 233
  - revoking certificates 233
  - searching
    - certificate requests 226
    - certificates 234
  - selected certificate requests 229
  - selected certificates 236
  - starting PKI Services 85
  - stopping PKI Services daemon 86
  - suspending certificates 233
- administration home page
  - accessing 217
  - using 221
- administration tasks
  - PKI Services 217
    - processing certificate requests 220
    - processing certificates 231
  - RACF
    - ongoing administration 241
    - running IKYSETUP 27
- administration Web application
  - PKI Services component description 4
- administration Web pages
  - changing access to 144
  - customizing 141, 143
  - fields 220
  - steps for customizing 143
  - using 217
- administration, RACF
  - classroom courses xviii
- administrative functions
  - protecting 27, 353
- administrator
  - access required 352
- adminPW
  - LDAP server configuration file 65, 68
- ADMINSCOPE subsection (in APPLICATION section of pkiserv.tmpl) 100, 143
- admmain.rexx 141
- admmodtid.rexx 141
- admpend.rexx 141
- admpendall.rexx 142
- admpendtid.rexx 141
- advanced customization 145
- AIEALNKE 341
- alias
  - file (for sendmail) 23
  - for certificate template
    - changing 126
- alias (*continued*)
  - for certificate template (*continued*)
    - description 103
    - list 103
  - for database entry 192
  - for IKYSPROC 384
  - PKISERVD 384
- AltDomain (named field in pkiserv.tmpl) 95
- AltEmail (named field in pkiserv.tmpl) 95
- alternate indexes
  - adding 80
- alternate name
  - domain name 201
  - e-mail address 201
  - IP address 202
  - OtherName, description 203
  - uniform resource identifier (URI) 203
- AltIPAddr (named field in pkiserv.tmpl) 95
- AltOther (named field in pkiserv.tmpl) 95
- AltURI (named field in pkiserv.tmpl) 95
- APARs
  - documented in z/OS Version 1 Release 5 xxv, xxvi
  - documented in z/OS Version 1 Release 7 xxiv, xxv
- APPL subsection (in TEMPLATE section of pkiserv.tmpl) 105
- application domains
  - adding 158
  - steps for adding 160
- APPLICATION section of pkiserv.tmpl
  - ADMINFOOTER subsection 143
  - ADMINHEADER subsection 143
  - ADMINSCOPE subsection 100, 143
- APPLICATION sections of pkiserv.tmpl
  - ADMINFOOTER subsection 100
  - ADMINHEADER subsection 100
  - CONTENT subsection 100
  - examining 114
  - RECONTENT subsection 100, 101
  - REFAILURECONTENT subsection 100, 101
  - RESUCCESSCONTENT subsection 100, 101
  - subsections 99
- approve (action on certificate request) 221
- approve with modifications (action on certificate request) 221
- approved (status of certificate request) 220
- approving certificate requests
  - multiple 229
  - selected 229
- APROCLIB 341
- ARL (authority revocation list) distribution point 156
- ARLDist (parameter in pkiserv.conf) 54, 155
- ASAMPLIB 341
- associating
  - user ID with PKI Services started procedure 27, 351
  - Web server and CA certificate to key ring 27
- attributes
  - HIGHTRUST 243
  - LDAP, that PKI Services requires 387
  - NOPASSWORD 351
  - OU 74, 76

- attributes (*continued*)
  - PROTECTED 352
  - RDN 387
  - RESTRICTED 33, 352
- Authenticode—code signing PKI server certificate
  - description 102
  - fields 111
- AuthName1 (parameter in pkiserv.conf) 73
- authority revocation list (ARL) 156
- AuthorityInformationAccess
  - certificate extension 268
- AuthorityKeyIdentifier
  - certificate extension 268
  - CRL extension 269
- authorization checking, using PKI exit 180
- authorizing
  - groups 242
  - PKI Services daemon user ID for CA functions 27
  - users for inquiry access 241, 242
- AuthPwd1 (parameter in pkiserv.conf) 74
- auto-approval
  - access required 353
  - of certificates 101, 102, 103
- automatic deletion
  - from ICL 53
  - from ObjectStore 53

## B

- backing up
  - CA certificate and private key 27, 354
  - RA certificate and private key 354
- backup\_dsn (variable in IKYSETUP) 36
- base64-encoded
  - #10 certificate request 98
  - certificate 92, 122, 124
  - response 6
- base64-encoded PKCS (field in end-user Web pages) 201
- base64cert substitution variable 92, 94, 122
- BasicConstraints (certificate extension) 268
- bibliography 409
- bin subdirectory 341
- bind passwords for LDAP
  - encrypted 260
  - in the clear 260
- binding
  - distinguished name for LDAP 73
  - passwords for LDAP servers 74
- BindProfile1 (parameter in pkiserv.conf) 75
- bpx\_userid. (variable in IKYSETUP) 34
- brackets (in substitution variables) 91
- browser certificates
  - n*-year PKI certificate for extensions demonstration 103
  - aliases 103
  - IdenTrust compliant template 397
  - installing 209, 210
  - n*-year PKI certificate for extensions demonstration 103
  - one-year PKI S/MIME browser certificate 102

- browser certificates (*continued*)
  - one-year PKI SSL browser certificate 102
  - one-year SAF browser certificate 102
  - requesting 204
  - retrieving 209, 210
  - supported types 7
  - two-year PKI browser certificate for authenticating to z/OS 102
- browsertype substitution variable 92

## C

- CA certificate
  - backing up 27
  - creating 353
  - using IKYSETUP 27
  - exporting 27
  - installing 199, 218
  - key rollover 254
  - locating 244
  - rekeying 254
  - renewing 249
- CA certificate profile, recovering 251
- CA domains
  - adding 161
- CA functions
  - authorizing PKI Services daemon user ID for 27
- CA revocation list (ARL) 156
- ca\_dn (variable in IKYSETUP) 29
- ca\_domain (variable in IKYSETUP) 37
- ca\_expires (variable in IKYSETUP) 37
- ca\_keysize (variable in IKYSETUP) 34
- ca\_label (variable in IKYSETUP) 30
- ca\_ring (variable in IKYSETUP) 37
- cadisplay.rexx 124
- cadomain substitution variable 92
- cagetcrt.rexx 124
- camain.rexx 123
- camodify.rexx 124
- capturing certificates 180
- careq.rexx 123
- caretrieve.rexx 124
- CARing (default name of SAF key ring) 37, 59
- catmpl.rexx 123
- CBC.SCLBDLL 35
- CDSA 3, 4
- CEE.SCEERUN 35
- CertGroupVerify 265
- certificate authority (CA)
  - certificate
    - backing up 27
    - creating 27
    - exporting 27
    - installing 199, 218
    - renewing 249
  - overview 3
- certificate data set, editing 248
- certificate extensions
  - customizing 8
  - host identity mapping 8
  - in PKI Services 8

- certificate extensions (*continued*)
  - standard 8
  - supported by PKITP 268
- certificate generation application Web page 100
- certificate policies
  - PKITP supports 267
  - using 146
- certificate preregistration
  - enabling in pkiserv.conf 55
- certificate profile, recovering 251
- certificate requests
  - actions on 221
  - changing 224
  - deleting 226
  - modifying 224
  - processing 217
    - multiple 228
    - selected 229
    - single 222
  - using searches 226
  - rejecting 226
  - relationship with certificates 238
  - searching 226
  - states 220
  - statuses 220
  - updating 224
- certificate revocation list (CRL)
  - constant portion, distribution point URI 55
  - constant portion, file system full path for the distribution point 54
  - constant portion, relative distinguished name 55
  - maximum number of certificates 55
  - revoked certificates on 231
  - suspended certificates on 231
  - time interval between issuances 58
  - validity period 55
- certificate serial number incrementer, restoring 252
- certificate store, using gskkyman for 389
- certificate suspension grace period 56
- certificate templates
  - adding 132
  - alias 103
  - file
    - customization, additional first-time 126
    - customization, minimal 124, 126
    - customizing for IdenTrust compliance 391
    - customizing the OtherName field 138
    - retrofitting release changes 129
  - IdenTrust compliant
    - browser certificate 397
    - server certificate 400
  - name 103
  - nickname 103
  - pkiserv.tmpl 101
  - subsections, summary 109
  - true name 103
- certificate validation service 265
- CertificateIssuer (CRL entry extension) 269
- CertificatePolicies extension
  - creating 146
  - in certificate 57
- CertificatePolicies extension (*continued*)
  - organization name for 57
  - PolicyCritical (parameter in pkiserv.conf) 56
  - supported by PKITP 268
  - using certificate policies 146
- certificates
  - actions on 232
  - auto-approval 101, 102, 103
  - capturing 180
  - deleting 233
  - extensions 8
  - locating 244
  - preregistering a SCEP client 214
  - preregistration 103
  - processing 217, 231
  - relationship with certificate requests 238
  - renewing 210
  - requesting 204
  - resuming 233
  - retrieving
    - from bookmarked page 209
    - from home page 210
  - revoking
    - by administrator 233
    - by user 213
  - searching 234
  - single 232
  - standard extensions 8
  - states 231
  - statuses 231
  - supported types 7
  - suspending
    - by administrator 233
    - by user 213
  - uses 7
  - X.509v3 support 8
- Certification Practice Statement
  - Uniform Resource Identifier 54
- CertPolicy section (of pkiserv.conf)
  - default values 54
  - description 50
  - excerpt 50
  - information needed 54
- CF lock structure
  - defining in SMS base configuration 79
  - defining to MVS 79
- CGI debugging, flag in pkiserv.tmpl 91
- CGIs
  - admactcert.rexx 142
  - admacttid.rexx 142
  - admacttid2.rexx 142
  - admicl.rexx 141
  - admiclall.rexx 142
  - admiclcert.rexx 141
  - admmain.rexx 141
  - admmodtid.rexx 141
  - admpend.rexx 141
  - admpendall.rexx 142
  - admpendtid.rexx 141
  - cadisplay.rexx 124
  - cagetcert.rexx 124

## CGIs (*continued*)

- camain.rexx 123
- camodify.rexx 124
- careq.rexx 123
- caretrieve.rexx 124
- catmpl.rexx 123
- summary 141
- chains 265
- challenge passphrase (field in end-user Web pages) 201
- ChallengePassPhrase (named field in pkiserv.tmpl) 95
- changes
  - summary xxiii
- changing
  - access to administration pages 144
  - certificate request 224
  - configuration file
    - overview 49
    - steps 51
  - e-mail notifications 135, 137
  - environment variables
    - overview 48
    - steps 49
  - exit 181
  - expiringmsg.form 137
  - forms for e-mail notifications 135
  - LDAP section of pkiserv.conf 75
  - log options 299
  - notification forms 135
  - parameters 180
  - pkixit.c 181
  - pkiserv.conf
    - overview 49
    - steps 51
  - pkitpsamp.c 274
  - readymsg.form 137
  - rejectmsg.form 137
  - runtime user ID 132
    - for requesting certificates 133
    - for retrieving certificates 134
  - signature algorithm 149
  - z/OS HTTP Server configuration files 67
- check boxes 229, 236
- CISIZE statements 80
- classroom courses, RACF xviii
- clear, LDAP bind passwords in 260
- client user ID 132
- ClientName (named field in pkiserv.tmpl) 95
- code samples
  - added in z/OS Version 1 Release 5 xxvi
  - configuration directives 371, 372
  - configuration file 343
  - environment variables file 349
  - expiringmsg.form 136
  - httpd.conf 371
  - httpd2.conf 372
  - IKYCVSAM 373
  - IKYMVSAM 377
  - IKYRVSAM 380
  - IKYSETUP 357

## code samples (*continued*)

- JCL to create VSAM data sets
  - alternate indexes 377
  - not using RLS 373
  - PATH data sets 377
  - using RLS 380
- pkiserv.conf 343
- pkiserv.envars 349
- pkiserv.tmpl
  - APPLICATION section 114
  - INSERT section 120
  - TEMPLATE section 117
- PKISERVD 384
- pkitpsamp.c 275
- procedure to start PKI Services daemon 384
- readymsg.form 135
- rejectmsg.form 136
- code signing server certificate
  - fields 111
- Common Data Security Architecture (CDSA) 3, 4
- common name (field in end-user Web pages) 201
- CommonName (named field in pkiserv.tmpl) 95
- completed (status of certificate request) 220
- compliance with IdenTrust 391
- components
  - diagram 5
  - in message numbers 315
- configurable section of IKYSETUP 27
- configuration directives
  - example 371, 372
- configuration file
  - example 343
  - for SSL traffic 67
  - pathname 347
  - updating 60
    - for IdenTrust compliance 397
    - overview 49
    - steps 51
- configuring
  - ICSF 22
  - IdenTrust 391
  - LDAP 20, 65
  - OCSF 19
  - PKITP 269
  - prerequisite products 17
  - sendmail 23
  - system for PKI Services 25
  - UNIX runtime environment 45
  - z/OS HTTP Server 17
- connecting
  - members to group 241
  - members to new group 242
- CONSTANT subsection (in TEMPLATE section of pkiserv.tmpl) 105
- CONTENT subsection
  - in APPLICATION section of pkiserv.tmpl 100
  - in TEMPLATE section of pkiserv.tmpl 104
- CONTROL access for IRR.DIGTCERT.GENCERT 353, 355
- copying
  - e-mail notifications files 47

- copying (*continued*)
  - expiringmsg.form 47
  - IKYCVSAM 79
  - IKYMVSAM 80
  - IKYRVSAM 81
  - pkiserv.conf 47
  - pkiserv.tmpl certificate templates file 47
  - readymsg.form 47
  - rejectmsg.form 47
- core function 348
- CORE subcomponent 348
- country (field in end-user Web pages) 201
- Country (named field in pkiserv.tmpl) 95
- courses about RACF xviii
- CPS in URI 54
- CPS1 (parameter in pkiserv.conf) 54, 148
- CreateInterval (parameter in pkiserv.conf) 54
- CreateOUValue (parameter in pkiserv.conf) 74
- creating
  - CA certificate 353
    - using IKYSETUP 27
  - CertificatePolicies extension 146
  - daemon user ID 27, 351
  - ICL data sets 79
  - implementation plan 14
  - IRR.PROXY.DEFAULTS profile 260
  - key ring 27
  - LDAPBIND class profile 261
  - PKI Services daemon user ID 27, 351
  - private key 27, 353
  - RA certificate 353
    - using IKYSETUP 27
  - SAF key ring 27, 353
  - SCEP RA certificate
    - using IKYSETUP 27
  - SSL certificate 27, 356
  - surrogate user ID 27
  - user ID
    - PKI Services daemon 27, 351
    - surrogate 27
  - VSAM data sets 77
    - alternate indexes 377
    - not using RLS 77, 79, 373
    - PATH data sets 377
    - space considerations 77
    - using RLS 81, 380
  - VSAM object store 79
- critical (marking of extension) 267
- critical flag 56
- CRL
  - entry extensions 269
  - extensions 269
  - revoked certificates on 231
  - suspended certificates on 231
- CRLDistDirPath (parameter in pkiserv.conf) 54, 154
- CRLDistName (parameter in pkiserv.conf) 55, 154
- CRLDistributionPoints (certificate extension) 156, 268
- CRLDistSize (parameter in pkiserv.conf) 55, 153
- CRLDistURI (parameter in pkiserv.conf) 55, 154
- CRLDuration (parameter in pkiserv.conf) 55
- CRLNumber (CRL extension) 269
- CRLReason (CRL entry extension) 269
- cryptographic service provider (field in end-user Web pages) 201
- cryptography
  - standards supported 6
- CSECTs
  - IKY8B 290
  - IKYP0N 289
  - IKYP81 289
  - IKYP8A 289, 290
  - IKYP8B 289
  - IKYSCHDR 291
  - IKYSTART 291
  - IKYTIMER 292
- CSF.SCSFMODE0 35
- CSF.SCSFMODE1 35
- csfkeys\_profile (variable in IKYSETUP) 34
- csfserv\_profile (variable in IKYSETUP) 34
- csfusers\_grp (variable in IKYSETUP) 35
- CSSM\_TP\_PassThrough
  - DBList 272
  - evidence 272
  - format 271
  - functions
    - CertGroupVerify 265
    - FreeEvidence 265
  - initial policy 272
  - parameters 271
  - performing certificate validation 274
  - purpose 271
  - return codes 273
- CUSTOMERS
  - application name 116
- customizing
  - administration Web pages 141, 143
    - steps 143
  - advanced 145
  - certificate extensions 8
  - certificate templates file
    - additional first-time 126
    - minimal 124
    - OtherName field 138
    - retrofitting release changes 129
  - e-mail notifications 135, 137
  - end-user Web pages 91
    - additional first-time 126
    - minimal 124
    - OtherName field 137, 138
    - retrofitting release changes 129
  - expiringmsg.form 137
  - forms for e-mail notifications 135
  - introduction 91
  - notification forms 135
  - pkiserv.tmpl
    - additional first-time 126
    - minimal 124
    - OtherName field 138
    - retrofitting release changes 129
  - readymsg.form 137
  - rejectmsg.form 137



## D

### daemon

- description of PKI Services component 5
- enabling to call OCSF functions 356
- PKI Services component
  - description 5
- sample procedure for starting 384
- starting 85
- stopping 86
- user ID
  - creating 27, 351
  - PKISRVD 37
  - WEBSRV 39

variable (user ID for PKI Services) 37

daemon (variable in IKYSETUP) 37

daemon\_uid (variable in IKYSETUP) 30

### data set name

- certificate expiring message form 59
- certificate ready message form 59
- certificate reject message form 59

data sharing environment, for VSAM RLS 79

DB subcomponent 348

DB2 20

debug flag 91

### decision tables

- key\_backup in IKYSETUP 32
- key\_type in IKYSETUP 32
- restrict\_surrog in IKYSETUP 33
- unix\_sec in IKYSETUP 33

### default

- /etc/pkiserv 9
- /usr/lpp/pkiserv 9
- /var/pkiserv 9
- binding information 260
- CA private key size 34
- CAring (SAF key ring) 37
- certificate, data set for backup copy of 36
- configuration file for z/OS HTTP Server 68
- daemon user ID
  - PKI Services 37
  - Web server 39
- data set
  - for copy of PKI Services certificate and private key 36
  - for copy of PKI Services certificate for copying to file system 37
  - for copy of PKI Services RA certificate and private key 38
- data set for backup copy of PKI Services certificate, private key 36
- data set for backup copy of PKI Services RA certificate, private key 38
- environment variables file 47
- file locations 63, 313
- high-level qualifier 51
- ICSF
  - profile to protect ICSF services 34
  - profile to protect PKI Services key 34
- installation directory 9
- key, data set for backup copy of 36
- key, data set for backup copy of RA certificate 38

### default (continued)

- message level 299, 348
- OMVSKERN (z/OS UNIX user ID) 34
- PKI Services
  - administration group 38
  - configuration file 85
  - daemon user ID 37, 59
  - surrogate user ID 39
- PKIGRP 38
- PKISERV 39
- pkiserv.conf file values 51
- PKISRVD (PKI Services daemon user ID) 37
- primary and secondary extent allocations (in JCL) 77
- RA certificate, data set for backup copy of 38
- registry directory for OCSF 19
- runtime directory 9
- SAF key ring 59
- sendmail location 48
- sha-1WithRSAEncryption 58
- STDOUT\_LOGGING 348
- surrogate user ID for PKI Services 39
- time zone 86
- UNIX user ID 34
- variables directory 9
- VSAM data set name
  - ICL base cluster 52
  - ICL requestor alternate index 53
  - ICL status alternate index 53
  - ObjectStore alternate index 52
  - ObjectStore base cluster 51
  - status 52
- Web server's daemon user ID 39
- WEBSRV 39

### defining

- IRR.PROXY.DEFAULTS profile 260
- KEYSMSTR class profile 261
- LDAPBIND class profile 261

delete (action for certificate) 232

delete (action on certificate request) 221

### deleting

#### certificate requests

- multiple 229
- selected 229
- single 226

#### certificates

- multiple 236
- selected 236
- single 233

#### groups 242

members 241, 242

#### preregistered requests

- single 226

diagnosing problems 289

diagnostic messages, logging 349

diagram, PKI Services system 5

### digital certificates

exporting certificates using R\_PKIServ callable service 257

generating certificates using R\_PKIServ callable service 257

- digital certificates *(continued)*
  - retrieving certificates using R\_PKIServ callable service 257
- DIR parameter 48
- directives
  - addtype 69
  - example 371, 372
  - exec 69
  - FastCGI 69
  - keyfile 69
  - log 70
  - normalmode 68, 69
  - pass 69
  - protect 69
  - protection 69
  - redirect 69
  - sslclientauth 69
  - sslmode 68, 69
  - sslport 68, 69
  - SSLX500CARoots 69
  - SSLX500Host 69
  - SSLX500Password 69
  - SSLX500Port 69
  - SSLX500UserID 69
  - userid 68, 69
- directories
  - /etc/pkiserv 9
  - /usr/lpp/pkiserv 9, 341
  - /var/pkiserv
    - description 9
    - for installation 9
    - for runtime 9
    - for variables 9
    - runtime 85
    - structure 341
    - var
      - setting up 63
- directory server, LDAP
  - planning for 10
  - requirements 387
- disability 403
- displaying
  - information about LDAPBIND class 261, 262
  - log options 300
- distinguished name
  - e-mail address 201
  - for LDAP binding 73
  - LDAP administrator's 65, 68
  - qualifiers
    - Email (deprecated) 96
    - EmailAddr 96
    - Mail 97
    - PostalCode 98
    - Street 99
- distribution point ARL 156
- DN fields
  - mapping to LDAP attributes 388
- documents
  - bibliography 409
  - configuring UNIX runtime environment 47

- documents *(continued)*
  - installing prerequisite products
    - ICSF 22
    - LDAP 20
    - OCSF 19
    - sendmail 23
    - z/OS HTTP Server 17
  - RACF administration 28
  - UNIX programmer 47
- domain name
  - field in end-user Web pages 201
  - fully qualified, for LDAP 65, 68, 73
- DSA
  - signature algorithm
    - SigAlg1 parameter 58
  - updating 149
  - standard supported 7

## E

- e-mail
  - applications 3
  - secure 3
- e-mail address
  - for alternate name (field in end-user Web pages) 201
  - for distinguished name (field in end-user Web pages) 201
  - for notifications 203
- e-mail notifications
  - adding PATH statement 49
  - copying files for 47
  - customizing
    - overview 135
    - steps for 137
  - editing 137
  - environment variables, updating for 48
  - ExpireWarningTime
    - description 56
    - updating 61
  - ExpiringMessageForm
    - description 59
    - updating 62
  - expiringmsg.form
    - copying 47
    - description 45
  - forms for
    - expiringmsg.form 136
    - readymsg.form 135
    - rejectmsg.form 136
  - NotifyEmail 97
  - PATH statement, adding 49
  - pkiserv.conf
    - updating, overview 49
    - updating, steps 60
  - ReadyMessageForm
    - description 59
    - updating 62
  - readymsg.form
    - copying 47
    - description 46



- e-mail notifications *(continued)*
  - RejectMessageForm
    - description 59
    - updating 62
  - rejectmsg.form
    - copying 47
    - description 47
  - retrieving your certificate 209
  - updating
    - environment variables 48
    - ExpireWarningTime 61
    - ExpiringMessageForm 62
    - expiringmsg.form 137
    - ReadyMessageForm 62
    - readymsg.form 137
    - RejectMessageForm 62
    - rejectmsg.form 137
  - variables 136
    - %%dn%% 136
    - %%notafter%% 136
    - %%requestor%% 136
    - %%transactionid%% 136
- editing
  - administration Web pages 144
  - certificate data set 248
  - certificate templates file
    - administration Web pages 144
    - end-user Web pages 124
  - configuration file
    - for configuring PKI Services 51
    - to add application sections 159
    - to change signature algorithm 149
    - to create CertificatePolicies extension 147, 148
    - to test configuration 85
  - e-mail notifications 137
  - end-user Web pages 124
  - expiringmsg.form 137
  - httpd.conf
    - to add application domains 161
  - httpd.envvars 181
  - IKYSETUP 40
  - log directives in httpd1443.conf 70
  - pkiserv.conf
    - for configuring PKI Services 51
    - to add application sections 159
    - to change signature algorithm 149
    - to create CertificatePolicies extension 147, 148
    - to customize distribution point CRLs 153
    - to test configuration 85
  - pkiserv.tmpl
    - administration Web pages 144
    - end-user Web pages 124
  - pkitsamp.c 274
  - readymsg.form 137
  - rejectmsg.form 137
  - schema.user.ldif 66
- Email (deprecated field in pkiserv.tmpl) 96
- EmailAddr (named field in pkiserv.tmpl) 96
- EnableSCEP (parameter in pkiserv.conf) 55
- encrypted passwords for LDAP servers 75
- encrypted passwords for LDAP servers *(continued)*
  - BindProfile1
    - description 75
    - updating 75
  - RACF administration 260
  - storing information for 71
  - updating LDAP section of pkiserv.conf 72
- encryption 149
- end-user function 257
- end-user functions
  - protecting 27, 351
- end-user Web application
  - PKI Services component
    - description 5
- end-user Web pages
  - accessing 198
  - code locations 130
  - customizing 91
    - additional first-time 126
    - minimal 124
    - OtherName field 138
  - fields 200
  - using 197
- environment variables
  - \_PKISERV\_CA\_DOMAIN 347
  - \_PKISERV\_EXIT 181, 347
  - \_PKISERV\_MSG\_LEVEL 348
  - \_PKISERV\_MSG\_LOGGING 348
  - \_PKISERV\_CONFIG\_PATH 347
- changing
  - overview 48
  - steps 49
- code sample file 349
- description 347
- file name
  - DIR parameter 48
  - FN parameter 48
- in PKISERVD 384
- OCSFREGDIR 49
- TZ 48
- updating
  - overview 48
  - steps 49
- error messages
  - list 315
  - new in z/OS Version 1 Release 5 xxvi
  - new in z/OS Version 1 Release 7 xxiv
  - new in z/OS Version 1 Release 8 xxiii
- error messages, logging 349
- errorinfo substitution variable 92
- EST5EDT 86
- establishing
  - PKI Services as an intermediate CA 247
  - RLS
    - enabling VSAM data sets for 81
    - preliminary steps 78
- event code, SMF 385
- examples
  - \_PKISERV\_CA\_DOMAIN 347
  - \_PKISERV\_MSG\_LEVEL 348
  - configuration directives 371, 372

## examples (continued)

- configuration file 343
- environment variables file 349
- expiringmsg.form 136
- httpd.conf 371
- httpd2.conf 372
- IKYCVSAM 373
- IKYMVSAM 377
- IKYRVSAM 380
- IKYSETUP 357
- JCL
  - certificate serial number incrementer, restoring 252
  - IKYCVSAM 374
  - IKYMVSAM 378
  - IKYRVSAM 380
  - PKISERVD 384
- ldapmodify 66
- log options settings 300
- LOGREC data 292
- named field 93
- output from displaying log options settings 300
- pkiserv.conf 343
- pkiserv.envvars 349
- pkiserv.tmpl
  - APPLICATION section 114
  - INSERT section 120
  - TEMPLATE section 117
- PKISERVD 384
- pkitpsamp.c 275
- procedure to start PKI Services daemon 384
- readymsg.form 135
- rejectmsg.form 136
- substitution variable 92

## excerpt

- pkiserv.conf
  - CertPolicy section 50
  - General section 50
  - LDAP section 71
  - ObjectStore section 50
  - OIDs section 50
  - SAF section 50
- pkiserv.tmpl 114

## exec directive 69

## exit

- \_PKISERV\_EXIT environment variable 347
- arguments 182
- description of PKI Services component 5
- environment variable 347
- pathname 347
- PKI Services component
  - description 5
- post-processing 184, 186
  - EXPORT 188
  - GENRENEW 184
  - REQRENEW 186
  - REVOKE 190
- preprocessing 185
  - EXPORT 187
  - GENCERT 183
  - GENRENEW 183

## exit (continued)

- preprocessing (continued)
  - REQRENEW 185
  - REVOKE 189
- scenarios 191
- updating sample code 181
- using 180
- expired (status of certificate) 231
- ExpireWarningTime (parameter in pkiserv.conf) 56
- expiring message form for certificate 59
- ExpiringMessageForm (parameter in pkiserv.conf) 59
- expiringmsg.form
  - code sample 136
  - copying 47
  - customizing 137
  - in samples directory 342
  - purpose 45
- EXPORT
  - accesses required 258
  - description 192
  - parameters
    - post-processing 188
    - preprocessing 187
  - R\_PKIServ function 352
  - return codes
    - post-processing 188
    - preprocessing 187
- export\_dsn (variable in IKYSETUP) 37
- exporting CA certificate 27
- extended key usage (field in end-user Web pages) 202
- extensions
  - CertificatePolicies 57
  - supported by PKITP 268
  - X.509 version 3 standard 8
- extensions demonstration
  - certificate
    - fields 110
- extent allocations in IKYCVSAM 77
- ExtKeyUsage (named field in pkiserv.tmpl) 96

## F

- FACILITY class profile
  - IRR.PROXY.DEFAULTS 260, 261
  - IRR.RPKISERV.PKIADMIN 259
- FAILURECONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 108
- FastCGI directive 69
- fields
  - administration Web pages 220
  - end-user Web pages 200
  - in IKYSETUP REXX exec
    - change based on setup 31
    - change optionally 36
    - change required 29
  - modifiable by administrator 226
  - X.509 version 3 standard 8
- file directory structure 341
- file system
  - installation directory 9

- file system (*continued*)
  - runtime directory 9
  - subdirectories 341
- files
  - CGIs
    - administrator Web pages 141
    - end-user Web pages 123
  - copying
    - for configuring PKI Services 47
  - exit 181
  - expiringmsg.form
    - copying 47
    - customizing 137
  - httpd.conf
    - code sample 371
    - updating 68
  - httpd1443.conf
    - source for 372
    - updating 69
  - httpd2.conf
    - code sample 372
    - copying from 69
  - IKYCVSAM
    - code sample 374
    - copying 79
    - extent allocations 77
    - updating 79
  - IKYMVSAM
    - code sample 378
    - copying 80
    - updating 80
  - IKYRVSAM
    - code sample 380
    - copying 81
    - updating 81
  - IKYSETUP
    - code sample 357
    - running 27
  - Makefile.pkiexit 181
  - pkiexit.c 181
  - pkiserv.conf
    - code sample 343
    - copying 47
    - updating 49, 72, 147, 148
  - pkiserv.envvars
    - code sample 349
    - copying 47
    - updating 49
  - pkiserv.tmpl
    - additional customization 126
    - contents 91
    - copying 47
    - minimal customization 124
    - retrofitting changes 129
  - PKISERVD
    - code sample 384
    - updating 49
  - PKITP 269
  - pkitp.h 341
  - pkitp.so 342

- files (*continued*)
  - readymsg.form
    - code sample 135
    - copying 47
    - customizing 137
  - rejectmsg.form
    - code sample 136
    - copying 47
    - customizing 137
- firewall certificate
  - description 102
  - fields 111
- five-year PKI intermediate CA certificate
  - description 103
  - fields 111
- five-year PKI IPSEC server (firewall) certificate
  - description 102
  - fields 111
- five-year PKI SSL server certificate
  - description 102
  - fields 111
- five-year SCEP certificate
  - description 103
- five-year SCEP preregistration
  - fields 113
- FN parameter 48
- forms for e-mail notifications
  - copying 47
  - customizing 135, 137
  - expiringmsg.form 136
  - readymsg.form 135
  - rejectmsg.form 136
  - variables 136
    - %%dn%% 136
    - %%notafter%% 136
    - %%requestor%% 136
    - %%transactionid%% 136
- FreeEvidence 265
- fully qualified domain name
  - LDAP 65, 68

## G

- GENCERT
  - accesses required 258
  - exit scenario use 191, 192
  - parameters
    - post-processing 184
    - preprocessing 183
  - R\_PKIServ function 352
  - return codes
    - post-processing 184
    - preprocessing 183
- General section (of pkiserv.conf)
  - default values 58
  - description 50
  - excerpt 50
  - information needed 58
- GENRENEW
  - accesses required 258
  - exit scenario use 192

## GENRENEW (continued)

- parameters
  - post-processing 184
  - preprocessing 183
- R\_PKIServ function 352
- return codes
  - post-processing 184
  - preprocessing 183
- GLD.SGLDLNK 35
- global ARL (CA revocation list) 156
- grace period, certificate suspension 56
- groups
  - authorizing 242
  - deleting 242
- GSK.SGSKLOAD 35
- gskkyman 389

## H

- HIGHTRUST attribute 243
- HoldInstructionCode (CRL entry extension) 269
- host identity mapping 8
- HostIdMap (named field in pkiserv.tmpl) 96
- HostIdMappings extension
  - administering 242, 243
  - field (on end-user Web pages) 202
  - PKITP support 268
- httpd.conf 371
  - application domains
    - adding 161
  - editing
    - to add application domains 161
- httpd.envvars 181
- httpd1443.conf 67
  - editing log directives in 70
- httpd2.conf 372

## I

- ICL
  - base cluster
    - VSAM data set name for 52
  - certificates maintained in 231
  - requestor
    - VSAM data set name for 53
  - space considerations 78
  - status alternate index
    - VSAM data set name for 53
  - time period before automatic deletion
    - expired certificates 53
- ICL data sets and indexes
  - creating 79
- ICLDSN (parameter in pkiserv.conf) 52
- ICLRequestorDSN (parameter in pkiserv.conf) 53
- ICLStatusDSN (parameter in pkiserv.conf) 53
- iclview
  - examples 303
  - format 302
  - parameters 302
  - purpose 302

## ICSF

- authorizing PKI Services 27
- configuring 22
- description of PKI Services component 5
- installing 22
- PKI Services component
  - description 5
  - public key data set (PKDS) 22
- ICSF programmer
  - installing and configuring ICSF 22
  - skills 12
  - tasks 12
  - team member 11
- IDCAMS 78
- IdenTrust, configuring PKI Services for compliance
  - adding an intermediate CA 392
  - adjusting general settings 392
  - defining templates 393
  - introduction 391
  - roadmap 393
  - samples for IdenTrust compliance
    - browser certificate template 397
    - configuration file 397
    - server certificate template 400
  - steps
    - adjusting general settings 395
    - creating templates 395
    - modifying pkiserv.conf 394
  - system prerequisites 392
  - task overview 392
  - task roadmap 393
- iecert substitution variable 92
- IKY8B CSECT 290
- IKYALLOC 341
- IKYCVSAM
  - copying 79
  - creating VSAM data sets 77
  - extent allocations 77
  - in SAMPLIB 341
  - sample 373
  - updating 79
- IKYDDDEF 341
- IKYISMKD 341
- IKYMKDIR 341
- IKYMVSAM
  - copying 80
  - in SAMPLIB 341
  - sample 377
  - updating 80
- IKYP0N CSECT 289
- IKYP81 CSECT 289
- IKYP8A CSECT 289, 290
- IKYP8B CSECT 289
- IKYPKID 341
- IKYPRTM 341
- IKYRVSAM
  - copying 81
  - in SAMPLIB 341
  - sample 380
  - updating 81
- IKYSCHDR CSECT 291

- IKYSETUP REXX exec
  - actions 351
  - code sample 357
  - decision tables
    - for key\_backup 32
    - for restrict\_surrog 33
    - for unix\_sec 33
    - key\_type 32
  - in SAMPLIB 341
  - parts 27
  - RACF administration 27
    - actions 351
    - steps for 39
  - sample log data set 42
  - structure and divisions 28
  - variables
    - backup\_dsn 36
    - bpx\_userid. 34
    - ca\_dn 29
    - ca\_domain 37
    - ca\_expires 37
    - ca\_keysize 34
    - ca\_label 30
    - ca\_ring 37
    - changes based on setup 31
    - changes optional 36
    - changes required 29
    - csfkeys\_profile 34
    - csfserv\_profile 34
    - csfusers\_grp 35
    - daemon 37
    - daemon\_uid 30
    - export\_dsn 37
    - key\_backup 32, 35
    - key\_type 32, 35
    - log\_dsn 38
    - pgmcntl\_dsn. 35
    - pki\_gid 30
    - pkigroup 38
    - pkigroup\_mem. 30
    - ra\_backup\_dsn 38
    - ra\_dn 30
    - ra\_label 31
    - restrict\_surrog 33, 36
    - signing\_ca\_label 38
    - surrog 39
    - surrog\_uid 31
    - unix\_sec 33, 36
    - vsamhlq 39
    - web\_dn 31
    - web\_expires 39
    - web\_label 39
    - web\_ring 31
    - webserver 39
- IKYSPROC 341
- IKYSTART CSECT 291
- IKYTIMER CSECT 292
- implementation plan
  - creating 14
  - tasks 14
- IMWEBSRV started procedure 70
- include subdirectory 341
- INETD 86
- informational messages, logging 349
- InitialThreadCount (parameter in pkiserv.conf) 58
- inquiry access, authorizing users for 241, 242
- INSERT sections of pkiserv.tmpl 93, 120
- INSERTs
  - AdditionalHeadIE 94
  - preregok 94
  - renewrevokebad 94
  - renewrevokeok 94
  - requestbad 94
  - requestok 94
  - return10cert 94
  - AltDomain 95
  - AltEmail 95
  - AltIPAddr 95
  - AltOther 95
  - AltURI 95
  - ChallengePassPhrase 95
  - ClientName 95
  - CommonName 95
  - Country 95
  - Email (deprecated) 96
  - EmailAddr 96
  - ExtKeyUsage 96
  - HostIdMap 96
  - KeyProt 96
  - KeyUsage 97
  - Label 97
  - Locality 97
  - Mail 97
  - NotAfter 97
  - NotBefore 97
  - NotifyEmail 97
  - Org 97
  - OrgUnit 97
  - OrgUnit2 97
  - PassPhrase 98
  - PostalCode 98
  - PublicKey 98
  - PublicKeyIE 98
  - PublicKeyNS 98
  - Requestor 98
  - returnbrowsercertIE 94
  - returnbrowsercertNS 94
  - SerialNumber 98
  - SignWith 98
  - StateProv 99
  - Street 99
  - Title 99
  - TransactionId 99
  - UnstructAddr 99
  - UserId 99
- install\_pkityp 269, 270
- install-dir 9
- installation directory 9
- installing
  - CA certificate 199, 218
  - ICSF 22
  - LDAP 20

- installing (*continued*)
  - OCSF 19
  - PKI Services
    - skills 13
    - SMP/E 9
  - prerequisite products
    - directions 17
    - skills 12
  - z/OS HTTP Server 17
- intermediate CA
  - certificate
    - description 103
    - description of template 103
    - fields 111
    - template description 103
  - establishing PKI Services as 247
- Internet Explorer
  - key protection field on end-user Web page 202
  - requesting a certificate 204
  - selecting a key size 205
  - supported standard 6
  - verifying certificate installed correctly 210
- Internet Protocol Security standard (IPSEC) 3
- interval
  - before certificate expiration 56
  - between certificate revocation lists 55, 58
  - certificate suspension grace period 56
  - scanning database for approved requests 54
  - warning message about certificate expiration 56
- InvalidityDate (CRL entry extension) 269
- IP address
  - AltIPAddr field 95
  - field in end-user Web pages 202
  - format 95
  - LDAP fully qualified domain name 65
- IPSEC
  - certificate format 6
  - certificates 7
  - supported standard 3
- IRR.DIGTCERT.ADD 258, 353
- IRR.DIGTCERT.CERTIFAUTH.\* 34
- IRR.DIGTCERT.EXPORT 258
- IRR.DIGTCERT.GENCERT 258, 353, 355
- IRR.DIGTCERT.GENRENEW 258, 353
- IRR.DIGTCERT.LIST 355
- IRR.DIGTCERT.LISTRING 355
- IRR.DIGTCERT.REQCERT 259, 352
- IRR.DIGTCERT.REQRENEW 259, 352
- IRR.DIGTCERT.RESPOND 259
- IRR.DIGTCERT.REVOKE 259
- IRR.DIGTCERT.SCEPREQ 259
- IRR.DIGTCERT.VERIFY 259
- IRR.PROXY.DEFAULTS 72, 260, 261
- IRR.RPKISERV 257
- IRR.RPKISERV.PKIADMIN 259, 353
- IRRSPX00 callable service
  - controlling the use 257
- IRRSPX00 SAF callable service
  - exits 181
  - PKI Services component description 5
- issued certificate list (ICL) 231

- IssuerAltName
  - certificate extension 268
- IssuerAltName (CRL extension) 269
- IssuingDistributionPoint (CRL extension) 269

## J

- JCL
  - creating VSAM data sets
    - alternate indexes 377
    - not using RLS 373
    - PATH data sets 377
    - using RLS 380
  - example
    - certificate serial number incrementer, restoring 252
    - IKYCVSAM 374
    - IKYMVSAM 378
    - IKYRVSAM 380
    - PKISERVD 384
  - EXEC card, PARM= operand limitation 48
  - VSAM data sets
    - alternate indexes 377
    - not using RLS 373
    - PATH data sets 377
    - using RLS 380
- JOB card 79, 80

## K

- key protection (field in end-user Web pages) 202
- key ring
  - associating Web server and CA certificates with 27
  - creating 27
  - locating 244
- key size (field in end-user Web pages) 202
- key usage (field in end-user Web pages) 202
- key\_backup (variable in IKYSETUP)
  - decision table 32
  - default value 35
  - description 35
- key\_type (variable in IKYSETUP)
  - decision table 32
  - default value 35
  - description 35
- keyboard 403
- keyfile directive 69
- KeyProt (named field in pkiserv.tmpl) 96
- KeyRing (parameter in pkiserv.conf) 59
- KEYSMSTR class
  - activating 261
  - profile, defining 261
- KeyUsage (certificate extension) 268
- KeyUsage (named field in pkiserv.tmpl) 97

## L

- label (field in end-user Web pages) 202
- Label (named field in pkiserv.tmpl) 97
- LDAP
  - adminDN keyword 73



## LDAP (continued)

- administrator's distinguished name
  - description 65, 68
- administrator's password
  - description 65, 68
- attributes
  - mapped to DN fields 388
  - mapped to object identifiers 388
  - PKI Services requires 387
- backend 20
- bind passwords
  - encrypted 260
  - in the clear 260
- configuring 20
- description of PKI Services component 5
- directory server requirements 387
- distinguished name
  - administrator's 65, 68
  - for binding 73
- domain name
  - description 65, 68
  - Server1 parameter 73
- encrypted passwords
  - BindProfile1 description 75
  - BindProfile1, updating 75
  - LDAPBIND class profile 71
  - RACF administration tasks for 260
  - storing information for 71
  - updating LDAP section of pkiserv.conf 72
- fully qualified domain name
  - description 65, 68
  - for LDAP server 73
  - Server1 parameter 73
- installing 20
- IP address
  - for LDAP server 73
- IP address and port 73
- objectclasses required by PKI Services 387
- OU attribute 74, 76
- password
  - administrator's 65, 68
  - encrypted 260
  - for binding 74
  - in the clear 260
- PKI Services component
  - description 5
- PKI Services objectclasses and attributes requirements 387
- port
  - description 65, 68
  - for LDAP server 73
- profile name 75
- retrying post requests 74
- servers available (number of) 72
- standard 7
- subcomponent for message logging 348
- suffix, description 65
- tailoring configuration for PKI Services 65
- TDBM DB2 backend 20
- time interval for scanning for items to post 72
- version 7

## LDAP programmer

- skills 12, 13
- tasks
  - configuration, tailoring LDAP 65
  - configuring LDAP 20
  - installing LDAP 20
  - LDAP configuration, tailoring 65
  - LDAP, installing and configuring 20
  - schema.user.ldif, updating 65
  - summary 12, 13
  - tailoring LDAP configuration 65
  - updating schema.user.ldif 65
- team member 11
- LDAP section (of pkiserv.conf)
  - default value 72
  - description 50
  - excerpt 71
  - information needed 72
  - tailoring 72
- LDAP server configuration file
  - adminPW 65, 68
- LDAPBIND class
  - displaying information about 261, 262
  - profile
    - creating 261
    - specifying name when configuring 72
- ldapmodify 66
- ldif2tdbm load utility 21, 65, 68
- legal statement about certificate issuance and use 58
- lib 342
- libraries
  - AIEALNKE 341
  - APROCLIB 341
  - ASAMPLIB 341
  - PROCLIB 341
  - SAMPLIB 341
  - SIEALNKE 341
- load libraries 35
- load utility (ldif2tdbm) 21, 65, 68
- loading
  - schema.user.ldif 66
  - sendmail configuration file 23
- local PKI certificate authority 30
- Local PKI RA (default RA certificate label) 59
- local PKI registration authority 31
- locality (field in end-user Web pages) 202
- Locality (named field in pkiserv.tmpl) 97
- locating
  - CA certificate 244
  - key ring 244
  - PKI Services certificates 244
  - RA certificate 244
- log data set
  - from running IKYSETUP 42
- log directives in httpd1443.conf, editing 70
- log options
  - changing 299
  - displaying 300
- log\_dsn (variable in IKYSETUP) 38
- logging message level 348

- LOGREC
  - description 289
  - sample data 292
- logs
  - changing options for 299
  - IKYSETUP data set sample 42
  - using information from 295
- LookAt message retrieval tool xviii

## M

- Mail (named field in pkiserv.tmpl) 97
- Makefile.pkixit 181
- Makefile.pkitpsamp 269
- mapping
  - access control for certificates to CGI directories 133
  - DN fields to LDAP attributes 388
  - host identity 8
- MaxSuspendDuration (parameter in pkiserv.conf) 56
- MD2 149
- MD5 7, 149
- members
  - connecting
    - to group 241
    - to new group 242
  - deleting 241, 242
- message form
  - certificate expiring 59
  - certificate ready 59
  - certificate rejected 59
- message levels
  - \_PKISERV\_MSG\_LEVEL 299
  - for logging 348
  - logging
    - diagnostic 349
    - error 349
    - informational 349
    - severe 349
    - verbose diagnostic 349
    - warning 349
- message logging
  - CORE subcomponent 348
  - DB subcomponent 348
  - LDAP subcomponent 348
  - PKID subcomponent 348
  - POLICY subcomponent 348
  - SAF subcomponent 348
  - TPOLICY subcomponent 348
- message numbers
  - components identified 315
- message retrieval tool, LookAt xviii
- message types 315
- messages
  - new in z/OS Version 1 Release 5 xxvi
  - new in z/OS Version 1 Release 7 xxiv
  - new in z/OS Version 1 Release 8 xxiii
- Microsoft Internet Explorer
  - key protection field on end-user Web page 202
  - requesting a certificate 204
  - selecting a key size 205
  - supported standard 6

- Microsoft Internet Explorer (*continued*)
  - verifying certificate installed correctly 210
- migrating
  - private key 27
- migrating from z/OS V1R3
  - e-mail notifications
    - customizing forms 137
    - ExpireWarningTime, updating 61
    - ReadyMessageForm, updating 62
    - updating ExpireWarningTime 61
    - updating ReadyMessageForm 62
  - encrypted passwords for LDAP servers
    - LDAPBIND class profile 71
    - RACF administration 260
    - storing information for 71
    - updating LDAP section of pkiserv.conf 72
  - LDAP servers, encrypted passwords for
    - LDAPBIND class profile 71
    - RACF administration 260
    - storing information for 71
    - updating LDAP section of pkiserv.conf 72
  - sysplex support
    - RLS, enabling VSAM data sets for 81
    - RLS, setting up, preliminary steps 78
    - SharedVSAM, updating 61
    - updating SharedVSAM 61
  - user notifications
    - customizing forms 137
- migrating to ICSF
  - CA certificate and private key 354
  - RA certificate and private key 354
- MODIFY command
  - change log options 299
  - display logging options 300
  - stop PKI Services daemon 87
- modifying
  - certificate request 224
- MVS programmer
  - installation of PKI Services 9
  - skills 13
  - tasks
    - creating VSAM data sets 77
    - enabling VSAM data sets for RLS 81
    - establishing RLS, preliminary steps 78
    - RLS, enabling VSAM data sets for 81
    - RLS, preliminary steps for establishing 78
    - starting PKI Services daemon 85
    - stopping PKI Services daemon 86
    - VSAM data sets, creating 77
  - team member 11
- MyPolicy (parameter in pkiserv.conf) 51, 147

## N

- n-year PKI certificate for extensions demonstration
  - description 103
- name
  - certificate templates
    - alias names 103
    - nicknames 103
    - short names 103



- name (*continued*)
  - certificate templates (*continued*)
    - table summarizing 103
    - true names 103
  - field (in end-user Web pages) 203
- named fields (in pkiserv.tmpl) 93
- Netscape
  - key size field on end-user Web page 202
  - requesting a certificate 204
  - selecting a key size 205
  - supported standard 6
  - verifying certificate installed correctly 210
- nickname
  - certificate template 103
- NOPASSWORD attribute 351
- normal operating mode of z/OS HTTP Server 355
- normalmode directive 68, 69
- not after date (field in end-user Web pages) 202
- not before date (field in end-user Web pages) 202
- NotAfter (named field in pkiserv.tmpl) 97
- NotBefore (named field in pkiserv.tmpl) 97
- notice
  - legal 58
  - number 57
- Notices 405
- notification e-mail address (field in end-user Web pages) 203
- notification forms
  - copying 47
  - customizing 135
- notifications
  - customizing 137
  - retrieving your certificate 209
- NotifyEmail
  - deleting 125
  - must match MAIL 212
- NotifyEmail (named field in pkiserv.tmpl)
  - description 97
- notifying users
  - copying files for 47
  - customizing forms 137
  - forms for
    - expiringmsg.form 136
    - readymsg.form 135
    - rejectmsg.form 136
- NumServers (parameter in pkiserv.conf) 72

## O

- Object ID
  - for policy 57
  - signing algorithm 58
- object identifiers
  - mapping to LDAP attributes 388
- object store
  - space considerations 78
- objectclasses
  - LDAP, that PKI Services requires 387
- ObjectDSN (parameter in pkiserv.conf) 51
- ObjectRequestorDSN (parameter in pkiserv.conf) 52
- ObjectStatusDSN (parameter in pkiserv.conf) 52

- ObjectStore
  - DB subcomponent for message logging 348
  - enabled for sysplex 54
  - section of pkiserv.conf
    - default value 51
    - description 50
    - excerpt 50
    - information needed 51
  - sysplex enabled 54
  - time period before automatic deletion
    - completed requests 53
    - inactive requests 53
    - incomplete requests 53
    - unsuccessful requests 53
  - VSAM data set names
    - requestor alternate index 52
    - status alternate index 52
    - TID alternate index 52
- ObjectTidDSN (parameter in pkiserv.conf) 52
- OCEP
  - configuring for PKITP 269
  - data library (DL) 272
  - optional installation 11
  - programmer
    - skills 12
    - team member 11
  - Trust Policy 265, 267, 272
- OCSF
  - configuring 19
  - data sets, providing access to 42
  - functions, enabling PKI Services daemon to call 356
  - installing 19
  - programmer 11
    - installing and configuring OCSF 19
    - skills 12
  - Trust Policy
    - module 270
    - overview 265
    - plug-in 265
- OCSF programmer
  - installing OCSF 19
- OCSFREGDIR environment variable 49
- OCSPTYPE (parameter in pkiserv.conf) 56
- OIDs section (of pkiserv.conf)
  - default value 51
  - description 50
  - excerpt 50
  - information needed 51
- OMVSKERN 34
- one-year PKI S/MIME browser certificate
  - description 102
  - fields 110
- one-year PKI SSL browser certificate
  - description 102
  - fields 110
- one-year SAF browser certificate
  - description 102
  - fields 113
- one-year SAF server certificate
  - description 101

- one-year SAF server certificate *(continued)*
  - fields 113
- optfield substitution variable 92
- Org (named field in pkiserv.tmpl) 97
- organization (field in end-user Web pages) 203
- organization name
  - for CertificatePolicies extension 57
- organizational unit (field in end-user Web pages) 203
- organizationalUnit objectclass 74
- OrgUnit (named field in pkiserv.tmpl) 97
- OrgUnit2 (named field in pkiserv.tmpl) 97
- OtherName (field in end-user Web pages)
  - customizing 137
  - description 203
- OU attribute 74, 76

## P

- Parallel Sysplex support
  - prerequisites 9
  - requirements 9
- parameters
  - changing 180
  - validating 180
- pass directive 69
- passphrase (field in end-user Web pages) 203
- PassPhrase (named field in pkiserv.tmpl) 98
- passwords
  - binding 260
  - encrypted, for LDAP servers
    - LDAPBIND class profile 71
    - RACF administration for 260
  - for LDAP binding 74
  - for LDAP servers, encrypted
    - storing information for 71
    - updating LDAP section of pkiserv.conf 72
  - LDAP
    - encrypted 260
    - in the clear 260
  - LDAP administrator's 65, 68
  - RACF administration for 260
- PATH statement 49
- pathname
  - certificate expiring message form 59
  - certificate ready message form 59
  - certificate reject message form 59
  - configuration file 347
  - exit program 347
- PDS 341
- pending approval (status of certificate request) 220
- pgmcntl\_dsn. (variable in IKYSETUP) 35
- PKCS #10
  - browser certificate format 6
  - certificate request 201
  - SCEP certificate request 175
  - server certificate format 6
- PKDS (public key data set) 22
- PKI (public key infrastructure)
  - defined 4
- PKI Authenticode-code signing server certificate
  - fields 111

- PKI browser certificate for authenticating to z/OS
  - description 102
  - fields 110
- PKI certificate for extensions demonstration
  - description 103
- PKI exit
  - arguments 182
  - PKI Services component
    - description 5
  - post-processing
    - EXPORT 188
    - GENCERT 184
    - GENRENEW 184
    - REQCERT 186
    - REQRENEW 186
    - REVOKE 190
  - preprocessing
    - EXPORT 187
    - GENCERT 183
    - GENRENEW 183
    - REQCERT 185
    - REQRENEW 185
    - REVOKE 189
  - scenarios 191
  - using 180
- PKI extensions demonstration certificate
  - fields 110
- PKI intermediate CA certificate
  - description 103
  - fields 111
- PKI IPSEC server (firewall) certificate
  - description 102
  - fields 111
- PKI S/MIME browser certificate
  - description 102
  - fields 110
- PKI Services
  - administering RACF 241
  - administration
    - changing log options 299
    - displaying log options settings 300
    - log options, changing 299
    - log options, displaying 300
    - starting PKI Services 85
    - stopping PKI Services daemon 86
    - using Web pages 217
  - administration group PKIGRP 38
  - administration Web application
    - component 4
  - authorizing for ICSF 27
  - CA 3
  - CA certificate
    - locating 244
  - certificate authority 3
  - certificate authority certificate, renewing 249
  - certificate types 7
  - certificates
    - locating 244
  - changing
    - environment variables 49
  - changing log options 299

## PKI Services *(continued)*

- compliance with IdenTrust 391
- component diagram 5
- components
  - administration Web application 4
  - diagram 5
  - end-user Web application 5
  - exit 5
  - ICSF 5
  - IRRSPX00 5
  - LDAP 5
  - list 4
  - PKI Services daemon 5
  - R\_PKIServ callable service 5
  - RACF 5
  - z/OS HTTP Server 5
- configuration file
  - overview 49
  - updating 51, 60
- configuration, testing 85
- cryptographic standards 6
- customizing
  - administration Web pages 141
  - advanced 145
  - end-user Web pages 91
- daemon
  - component 5
- daemon user ID
  - authorizing for CA functions 27
  - creating 27
  - PKISRVD 37
- directory structure 341
- end-user Web application 5
- environment variables
  - updating 49
- exit 5
- extensions supported 8
- fields supported 8
- file directory structure 341
- ICSF
  - component 5
  - installing 22
- IdenTrust CA, configuring PKI Services as 391
- implementation plan 14
- installing
  - skills 13
  - SMP/E 9
- intermediate certificate authority 246
- introduction 3
- IRRSPX00 5
- key ring
  - locating 244
- key rollover 254
- LDAP
  - attributes requirements 387
  - component 5
  - objectclasses requirements 387
  - tailoring configuration for PKI Services 65
  - tailoring pkiserv.conf 71
- log options, changing 299
- logs 295

## PKI Services *(continued)*

- OCSF Trust Policy plug-in 265
- overview 3
- PKI exit
  - component 5
  - using 180
- planning 9
- prerequisite products
  - ICSF 10
  - installing and configuring 17
  - LDAP server 10
  - OCEP 11
  - OCSF 10
  - planning for 10
  - z/OS HTTP Server 10
- private key rollover 254
- protecting administrative and end-user functions 27, 351
- R\_PKIServ callable service (IRRSPX00)
  - component 5
- RA certificate
  - locating 244
- RACF
  - administration 241
  - component 5
  - using IKYSETUP 27
- rekeying CA certificate 254
- related products 4
- renewing certificate authority certificate 249
- roadmap
  - for IdenTrust compliance 393
  - for implementing PKI Services 14
- rollover of private key 254
- SAF key ring 59
- skill requirements 11
- standards 6
- starting 85
- stopping 85, 86
- subordinate certificate authority 246
- surrogate user ID PKISERV 39
- task roadmap
  - for IdenTrust compliance 393
  - for implementing PKI Services 14
- team members 11
- testing configuration 85
- UNIX utilities 301
- updating
  - certificate templates file 124, 126, 129
  - configuration file 60
  - environment variables 49
- uses 3
- using
  - administration Web pages 217
  - end-user Web pages 197
- utilities 301
  - iclview 302
  - pkiprereg 309
  - vosview 305
- Web pages
  - customizing 91, 141
  - using 197, 217

- PKI Services *(continued)*
  - z/OS HTTP Server
    - component 5
    - installing 17
    - updating configuration 67
  - z/OS product libraries 341
- PKI Services administration
  - deleting certificate requests 226
  - deleting certificates 233
  - deleting preregistered requests 226
  - group, setting up 27
  - modifying certificate request 224
  - processing certificate requests
    - multiple 227
    - selected 229
    - single 222
    - using searches 226
  - processing certificates 231
    - multiple 234
    - selected 236
    - single 232
  - processing preregistered requests
    - single 222
  - rejecting certificate requests 226
  - resuming certificates 233
  - revoking certificates 233
  - searching
    - certificate requests 226
    - certificates 234
  - selected certificate requests 229
  - selected certificates 236
  - setting up group 27
  - suspending certificates 233
- PKI Services daemon
  - enabling OCSF functions 356
  - starting 85, 384
  - user ID 59
- PKI Services daemon user ID
  - creating 351
- PKI Services OCSF Trust Policy
  - API
    - CSSM\_TP\_PassThrough 270
    - overview 265
- PKI Services started procedure
  - associating user ID with 27
- PKI SSL browser certificate
  - description 102
  - fields 110
- PKI SSL server certificate
  - description 102
  - fields 111
- PKI Windows logon certificate
  - fields 110
- pki\_gid (variable in IKYSETUP) 30
- PKID
  - subcomponent for message logging 348
- pkixit.c
  - description 181
  - scenarios 191
  - updating sample code 181
- pkigroup (variable in IKYSETUP) 38
- pkigroup\_mem. (variable in IKYSETUP) 30
- PKIGRP 38
- pkiprereg
  - examples 311
  - format 309
  - input data file 310
  - parameters 309
  - purpose 309
- PKISERV
  - application name 99, 114
  - runtime user ID 132
  - surrogate user ID 39, 352
  - z/OS HTTP Server operating modes required 355
- PKIServ subdirectory 342
- pkiserv.conf
  - application sections
    - adding 159
  - CertPolicy section
    - default values 54
    - description 50
    - excerpt 50
    - information needed 54
  - changing
    - signature algorithm 149
    - to add application sections 159
  - code sample 343
  - copying 47
  - distribution point CRLs, customizing 153
  - editing
    - for configuring PKI Services 51
    - to add application sections 159
    - to change signature algorithm 149
    - to create CertificatePolicies extension 147, 148
    - to customize distribution point CRLs 153
    - to test configuration 85
  - General section
    - default values 58
    - description 50
    - excerpt 50
    - information needed 58
  - LDAP section
    - default value 72
    - description 50
    - excerpt 71
    - information needed 72
  - ObjectStore section
    - default value 51
    - description 50
    - excerpt 50
    - information needed 51
  - OIDs section
    - default value 51
    - description 50
    - excerpt 50
    - information needed 51
  - parameters
    - ARLDist 54, 155
    - AuthName1 73
    - AuthPwd1 74
    - BindProfile1 75
    - CPS1 54, 148

- pkiserv.conf *(continued)*
  - parameters *(continued)*
    - CreateInterval 54
    - CreateOUValue 74
    - CRLDistDirPath 54, 154
    - CRLDistName 55, 154
    - CRLDistSize 55, 153
    - CRLDistURI 55, 154
    - CRLDuration 55
    - EnableSCEP 55
    - ExpireWarningTime 56
    - ExpiringMessageForm 59
    - ICLDSN 52
    - ICLRequestorDSN 53
    - ICLStatusDSN 53
    - InitialThreadCount 58
    - KeyRing 59
    - MaxSuspendDuration 56
    - MyPolicy 51, 147
    - NumServers 72
    - ObjectDSN 51
    - ObjectRequestorDSN 52
    - ObjectStatusDSN 52
    - ObjectTidDSN 52
    - OCSPTYPE 56
    - Policy1Notice1 57, 147
    - Policy1Notice2 57
    - Policy1Org 57, 147
    - PolicyCritical 56, 147
    - PolicyName1 57, 147
    - PolicyRequired 57, 147, 148
    - PostInterval 72
    - RA\_label 59
    - ReadyMessageForm 59
    - RejectMessageForm 59
    - RemoveCompletedReqs 53
    - RemoveExpiredCerts 53
    - RemoveInactiveReqs 53
    - RetryMissingSuffix 74
    - Server1 73
    - SharedVSAM 54
    - SigAlg1 58, 150
    - TimeBetweenCRLs 58
    - UserNoticeText1 58, 148
  - purpose 46
  - SAF section
    - default value 59
    - description 50
    - excerpt 50
    - information needed 59
  - sections
    - CertPolicy section 50
    - General section 50
    - LDAP section 50
    - ObjectStore section 50
    - OID section 50
    - SAF section 50
  - signature algorithm
    - changing 149
  - steps for updating LDAP section 75
  - updating 60

- pkiserv.conf *(continued)*
  - overview 49
  - steps 51
- pkiserv.envars
  - code sample 349
  - purpose 46
  - updating 48
- pkiserv.tmpl
  - APPLICATION section 114
    - ADMINFOOTER subsection 143
    - ADMINHEADER subsection 143
    - ADMINSCOPE subsection 100, 143
  - APPLICATION sections
    - ADMINFOOTER subsection 100
    - ADMINHEADER subsection 100
    - CONTENT subsection 100
    - RECONTENT subsection 100, 101
    - REFAILURECONTENT subsection 100, 101
    - RESUCCESSCONTENT subsection 100, 101
    - subsections 99
  - certificate templates 101
  - copying 47
  - customizing
    - customization, additional first-time 126
    - customizing the OtherName field 138
    - minimally 124
    - retrofitting release changes 129
  - debug flag 91
  - description 91
  - editing
    - administration Web pages 144
    - end-user Web pages 124
  - INSERT sections 93, 120
  - INSERTs
    - AdditionalHeadIE 94
    - preregok 94
    - renewrevokebad 94
    - renewrevokeok 94
    - requestbad 94
    - requestok 94
    - return10cert 94
    - AltDomain 95
    - AltEmail 95
    - AltIPAddr 95
    - AltOther 95
    - AltURI 95
    - ChallengePassPhrase 95
    - ClientName 95
    - CommonName 95
    - Country 95
    - Email (deprecated) 96
    - EmailAddr 96
    - ExtKeyUsage 96
    - HostIdMap 96
    - KeyProt 96
    - KeyUsage 97
    - Label 97
    - Locality 97
    - Mail 97
    - NotAfter 97
    - NotBefore 97

- pkiserv.tmpl (continued)
  - INSERTs (continued)
    - NotifyEmail 97
    - Org 97
    - OrgUnit 97
    - OrgUnit2 97
    - PassPhrase 98
    - PostalCode 98
    - PublicKey 98
    - PublicKeyIE 98
    - PublicKeyNS 98
    - Requestor 98
    - returnbrowsercertIE 94
    - returnbrowsercertNS 94
    - SerialNumber 98
    - SignWith 98
    - StateProv 99
    - Street 99
    - Title 99
    - TransactionId 99
    - UnstructAddr 99
    - UserId 99
  - named fields 93
    - %%-renewrevokebad%% 101
    - %%-renewrevokeok%% 101
    - %%-requestok%% 108
    - %%SelectCADomain%% 101
  - purpose 46
  - sections 91
  - substitution variables 91
  - TEMPLATE section 117
    - ADMINAPPROVE subsection 106
    - APPL subsection 105
    - CONSTANT subsection 105
    - CONTENT subsection 104
    - FAILURECONTENT subsection 108
    - PREREGISTER subsection 109
    - RETRIEVECONTENT subsection 108
    - RETURNCERT subsection 109
    - subsections 104
    - SUCCESSCONTENT subsection 107
  - updating
    - customization, additional first-time 126
    - customizing the OtherName field 138
    - minimally 124
    - retrofitting release changes 129
- PKISERVD
  - code sample 384
  - in PROCLIB 341
  - updating environment variables 48
- PKISRVD
- PKI Services daemon user ID 37, 59
- PKITP
  - API called CSSM\_TP\_PassThrough 270
  - certificate extensions supported 268
  - certificate policies supported 267
  - configuring 269
  - files 269
  - overview 265
  - PKI Services Trust Policy plug-in for OCSF 265
- pkitp\_ivp 269, 270
- pkitp.h 269
- pkitp.so 269
- pkitpsamp.c
  - description and directory 269
  - editing 274
  - sample code 275
- PKIX
  - support for interoperability 4
  - supported by PKI Services 3
- planning
  - for PKI Services 9
- policy
  - notice number 57
  - Object ID for 57
  - usage 51
- POLICY
  - subcomponent for message logging 348
- Policy1Notice1 (parameter in pkiserv.conf) 57, 147
- Policy1Notice2 (parameter in pkiserv.conf) 57
- Policy1Org (parameter in pkiserv.conf) 57, 147
- PolicyCritical (parameter in pkiserv.conf) 56, 147, 267
- PolicyName1 (parameter in pkiserv.conf) 57, 147
- PolicyRequired (parameter in pkiserv.conf) 57, 147, 148, 267
- ports
  - 1443, port 67
  - 443, port 67
  - 80, port 67
  - for HTTP traffic 67
  - for SSL traffic 67
  - LDAP 65, 68
- post requests
  - retrying for LDAP 74
- post-processing
  - exit 181
  - EXPORT 188
  - GENCERT 184
  - GENRENEW 184
  - REQCERT 186
  - REVOKE 190
- postal code
  - DN field supported 388
  - field in end-user Web pages 203
- PostalCode (named field in pkiserv.conf) 98
- PostInterval (parameter in pkiserv.conf) 72
- preprocessing
  - exit 181
  - EXPORT 187
  - GENCERT 183
  - GENRENEW 183
  - REQCERT 185
  - REVOKE 189
- PREREGISTER subsection (in TEMPLATE section of pkiserv.conf) 109
- preregistered (status of certificate request) 221
- preregistered requests
  - deleting 226
  - processing
    - single 222
- preregistering
  - steps for 214



- preregistration
  - certificate for SCEP 103
  - enabling in pkiserv.conf 55
- preregistration certificate
  - certificate
    - fields 113
- prerequisite products
  - configuring 17
  - installing 17
  - skills 12
  - planning for 10
- prerequisites
  - for sysplex support 9
  - products 17
- printablecert substitution variable 92
- private key
  - backing up 27
  - creating 27, 353
  - migrating to ICSF 27
  - storing
    - in ICSF 22
    - in RACF 5
- private key of PKI Services
  - rekeying 254
  - replacing 254
  - retiring 254
  - rollover 254
- problems, diagnosing 289
- processing
  - certificate requests
    - actions 221
    - introduction 217
    - multiple 228
    - selected 229
    - single 222
    - using searches 227
  - certificates
    - actions 232
    - introduction 217
    - multiple 234
    - overview 231
    - selected 236
    - single 232
    - using searches 234
  - preregistered requests
    - single 222
- PROCLIB 341
- product libraries 341
- profile
  - CA certificate, recovering 251
  - FACILITY class
    - IRR.PROXY.DEFAULTS 260, 261
    - IRR.DIGTCERT.ADD 258
    - IRR.DIGTCERT.EXPORT 258
    - IRR.DIGTCERT.GENCERT 258
    - IRR.DIGTCERT.GENRENEW 258
    - IRR.DIGTCERT.REQCERT 259
    - IRR.DIGTCERT.REQRENEW 259
    - IRR.DIGTCERT.RESPOND 259
    - IRR.DIGTCERT.REVOKE 259
    - IRR.DIGTCERT.SCEPREQ 259

- profile (*continued*)
  - IRR.DIGTCERT.VERIFY 259
  - IRR.RPKISERV.PKIADMIN 259
  - KEYSMSTR class
    - LDAP.BINDPW.KEY 261
  - LDAPBIND class profile
    - defining 261
- protect directive 69
- PROTECTED attribute 352
- protection directive 69
- protocols
  - supported in PKI Services 6
- province 203
- public key cryptography
  - standards supported 6
- public key data set (PKDS) 22
- Public Key Infrastructure for X.509 3
- publications
  - on CD-ROM xvii, xviii
  - softcopy xvii, xviii
- PublicKey (named field in pkiserv.tmpl) 98
- PublicKeyIE (named field in pkiserv.tmpl) 98
- PublicKeyNS (named field in pkiserv.tmpl) 98

## R

- R\_PKIServ
  - end-user functions 257
- R\_PKIServ callable service
  - controlling the use 257
  - description of PKI Services component 5
  - PKI Services component description 5
  - protected by FACILITY class resources 352
- RA certificate
  - creating 353
  - using IKYSETUP 27
  - locating 244
- ra\_backup\_dsn (variable in IKYSETUP) 38
- ra\_dn (variable in IKYSETUP) 30
- RA\_label (parameter in pkiserv.conf) 59
- ra\_label (variable in IKYSETUP) 31
- RACF
  - administering PKI Services 241
  - authorizing
    - READ access 242
    - users for inquiry access 241
  - connecting members
    - to group 241
    - to new group 242
  - deleting groups 242
  - deleting members 241, 242
  - description of PKI Services component 5
  - PKI Services component
    - description 5
  - publications
    - on CD-ROM xvii, xviii
    - softcopy xvii, xviii
  - setting up PKI Services 27
- RACF administration
  - for PKI Services, ongoing 241
  - for setting up PKI Services using IKYSETUP 27

- RACF administration (*continued*)
  - steps for 39
  - using IKYSETUP 351
- RACF administrator
  - ongoing administration for PKI Services 241
  - running IKYSETUP
    - overview 27
    - steps 41
  - skills 13
  - tasks 13
    - IKYSETUP, running 27
    - ongoing administration for PKI Services 241
    - performed by IKYSETUP 351
    - running IKYSETUP 27
    - setting up PKI Services using IKYSETUP 39
  - team member 11
- RACF group
  - providing access 27
- RDB (request database) 238
- RDN attribute 387
- READ access
  - authorizing 242
  - IRR.DIGTCERT.GENCERT 355
  - IRR.DIGTCERT.GENRENEW 353
  - IRR.DIGTCERT.LIST 355
  - IRR.DIGTCERT.LISTRING 355
  - IRR.DIGTCERT.REQCERT 352
  - IRR.DIGTCERT.REQRENEW 352
  - IRR.RPKISERV.PKIADMIN 353
- ready message form for certificate 59
- ReadyMessageForm (parameter in pkiserv.conf) 59
- readymsg.form
  - code sample 135
  - copying 47
  - customizing 137
  - in samples directory 342
  - purpose 46
- recent activity (field in administration Web pages) 220
- RECONTENT subsection (in APPLICATION section of pkiserv.tmpl) 100
- RECONTENT subsection (in APPLICATION sections of pkiserv.tmpl) 101
- record level sharing (VSAM RLS) 79
- recording
  - errors 289
- recovering
  - CA certificate profile 251
- redirect directive 69
- REFAILURECONTENT subsection (in APPLICATION section of pkiserv.tmpl) 100
- REFAILURECONTENT subsection (in APPLICATION sections of pkiserv.tmpl) 101
- registration authority (CA)
  - certificate
    - creating 27
- registry directory for OCSF 19
- reject (action on certificate request) 221
- reject message form for certificate 59
- rejected (status of certificate request) 221
- rejected, user notified (status of certificate request) 221
- rejecting
  - certificate requests
    - multiple 229
    - selected 229
    - single 226
- RejectMessageForm (parameter in pkiserv.conf) 59
- rejectmsg.form
  - code sample 136
  - copying 47
  - customizing 137
  - in samples directory 342
  - purpose 47
- rekeying, PKI Services private key 254
- relationship between certificate requests and certificates 238
- relocate section
  - variable data for type 80 SMF records 385
- RemoveCompletedReqs (parameter in pkiserv.conf) 53
- RemoveExpiredCerts (parameter in pkiserv.conf) 53
- RemoveInactiveReqs (parameter in pkiserv.conf) 53
- removing
  - groups 242
  - members 242
- renew (action for certificate) 232
- Renew or revoke a browser certificate Web page 101
- renewing
  - certificate
    - steps for 210
  - PKI Services certificate authority certificate 249
- replacing, PKI Services private key 254
- REQCERT
  - accesses required 259
  - exit scenario use 191, 192
  - parameters
    - post-processing 186
    - preprocessing 185
  - R\_PKIServ function 352
  - return codes
    - post-processing 186
    - preprocessing 185
- REQRENEW
  - accesses required 259
  - exit scenario use 192
  - parameters
    - post-processing 186
    - preprocessing 185
  - R\_PKIServ function 352
  - return codes
    - post-processing 186
    - preprocessing 185
- request database (RDB) 238
- requesting
  - certificate
    - steps for 204
  - preregistration
    - steps for 214
- requestor
  - alternate index
    - VSAM data set name for 52
- Requestor (named field in pkiserv.tmpl) 98



- requestor name (field in administration Web pages) 220
- requirements
  - LDAP directory server 387
  - prerequisite products 10
  - skills 11
  - sysplex support 9
- RESPOND
  - accesses required 259
  - R\_PKIServ function 352
- restoring
  - certificate serial number incrementer 252
- restrict\_surrog (variable in IKYSETUP)
  - decision table 33
  - default value 36
  - description 36
- RESTRICTED attribute 352
- RESUCCESSCONTENT subsection (in APPLICATION section of pkiserv.tmpl) 100
- RESUCCESSCONTENT subsection (in APPLICATION sections of pkiserv.tmpl) 101
- resume (action for certificate) 232
- resuming
  - certificates
    - selected 236
    - single 233
- resuming certificates
  - by administrator
    - multiple 236
- retiring, PKI Services private key 254
- RETRIEVECONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 108
- retrieving
  - certificate
    - steps for 209, 210
- retrofitting release changes into pkiserv.tmpl 129
- retrying
  - LDAP post requests 74
- RetryMissingSuffix (parameter in pkiserv.conf) 74
- return codes
  - CSSM\_TP\_PassThrough 273
- EXPORT
  - post-processing 188
  - preprocessing 187
- GENCERT
  - post-processing 184
  - preprocessing 183
- GENRENEW
  - post-processing 184
  - preprocessing 183
- recording 289
- REQCERT
  - post-processing 186
  - preprocessing 185
- REQRENEW
  - post-processing 186
  - preprocessing 185
- REVOKE
  - post-processing 190
  - preprocessing 189
- returnbrowsercertIE 94
- returnbrowsercertNS 94
- RETURNCERT subsection (in TEMPLATE section of pkiserv.tmpl) 109
- revocation list, certificate authority (ARL) 156
- REVOKE
  - accesses required 259
  - parameters
    - post-processing 190
    - preprocessing 189
  - R\_PKIServ function 352
  - return codes
    - post-processing 190
    - preprocessing 189
- revoke (action for certificate) 232
- revoked (status of certificate) 231
- revoked, expired (status of certificate) 231
- revoking
  - certificates
    - selected 236
- revoking certificates
  - by administrator
    - multiple 236
    - single 233
  - by user 213
- RFC2587.Idif 66
- RLS
  - enabling VSAM data sets for 81
  - MVS programmer task 13
  - preliminary steps for establishing 79
  - setting up, preliminary steps 78
- RLS, VSAM record-level sharing 380
- roadmaps
  - configuring PKI Services for IdenTrust compliance 393
  - for implementing PKI Services 14
- roles 11
- rollover, PKI Services private key 254
- RSA
  - signature algorithm
    - SigAlg1 parameter 58
    - updating 149
  - standard supported 7
- runtime directory 85
- runtime environment
  - configuring 45
- runtime user ID
  - changing 132
  - for requesting certificates 133
  - for retrieving certificates 134
- runtime-dir 9

**S**

- S/MIME
  - certificate format 6
  - description of certificate 102
  - fields of certificate 110
  - supported standard 3
  - use of certificate 7

- SAF
  - browser certificate
    - description 102
    - fields 113
  - key ring
    - creating 27, 353
    - KeyRing parameter 59
  - section (of pkiserv.conf)
    - default value 59
    - description 50
    - excerpt 50
    - parameter description 59
  - server certificate
    - description 101
    - fields 113
  - subcomponent for message logging 348
- SAMPLB 103
- samples
  - \_PKISERV\_CA\_DOMAIN 347
  - \_PKISERV\_MSG\_LEVEL 348
  - added in z/OS Version 1 Release 5 xxvi
  - browser certificate template 397
  - configuration directives 371, 372
  - configuration file 343
  - configuration file for IdenTrust compliance 397
  - directives 371, 372
  - environment variables file 349
  - expiringmsg.form 136
  - httpd.conf 371
  - httpd2.conf 372
  - IKYCVSAM 373
  - IKYMVSAM 377
  - IKYRVSAM 380
  - IKYSETUP 357
  - JCL
    - certificate serial number incrementer, restoring 252
    - IKYCVSAM 374
    - IKYMVSAM 378
    - IKYRVSAM 380
    - PKISERVD 384
  - log data set from IKYSETUP 42
  - LOGREC data 292
  - pkiserv.conf 343
  - pkiserv.envars 349
  - pkiserv.tmpl
    - APPLICATION section 114
    - INSERT section 120
    - TEMPLATE section 117
  - PKISERVD sample procedure 384
  - pkitpsamp.c 275
  - readymsg.form 135
  - rejectmsg.form 136
  - server certificate template for IdenTrust compliance 400
- samples subdirectory 342
- SAMPLIB 341
- scenarios
  - PKI exit 191
    - allowing only selected users to request certificates 191
  - scenarios (*continued*)
    - PKI exit (*continued*)
      - maintaining customized certificate repository 192
      - providing customized TITLE 191
      - renewal only within 30 days of expiration 192
- SCEP certificates
  - requesting 214
- SCEP preregistration certificate
  - fields 113
- SCEPREQ
  - accesses required 259
- schema.user.ldif
  - editing 66
  - loading 66
- searching
  - certificate requests 226
  - certificates 234
- sections of pkiserv.conf
  - CertPolicy section 50
  - General section 50
  - LDAP section 50
  - ObjectStore section 50
  - OIDs section 50
  - SAF section 50
- secure
  - e-mail 3
- Secure Multipurpose Internet Mail Extensions (S./MIME) 3
- Secure Sockets Layer (SSL) 3, 6
- security topics for RACF
  - classroom courses xviii
- selected certificate requests
  - processing 229
- selected certificates
  - processing 236
- sendmail
  - configuring 23
  - not using default 48
  - planning for 11
- serial number
  - field in administration Web pages 220
  - incrementer, restoring 252
- SerialNumber (named field in pkiserv.tmpl) 98
- SERVAUTH class 243
- server certificates
  - aliases 103
  - five-year PKI intermediate CA certificate 103
  - five-year PKI IPSEC server (firewall) certificate 102
  - five-year PKI SSL server certificate 102
  - generating 27
  - IdenTrust compliant template 400
  - installing 209, 210
  - one-year SAF server certificate 101
  - retrieving 209, 210
  - supported types 7
- Server1 (parameter in pkiserv.conf) 73
- setting up
  - access control 27, 351
  - PKI Services 25
  - PKI Services administration group 27
  - prerequisite products 17

- setting up (*continued*)
  - RLS
    - enabling VSAM data sets for 81
    - preliminary steps 78
  - var directory 64
  - z/OS HTTP Server for surrogate operation 27, 356
- settings
  - contained in pkiserv.conf 46
  - displaying log options 300
  - IKYP025I displays 332
  - log options, displaying 300
  - permission, changing with chmod 64
- severe messages, logging 349
- SHA1 7, 149
- SharedVSAM (parameter in pkiserv.conf) 54
- sharing control data sets (SHCDS) 79
- SHCDS 79
- shortcut keys 403
- SIEALNKE 341
- SigAlg1 (parameter in pkiserv.conf) 58, 150
- signature algorithm
  - Object ID 58
  - updating 149
- signing key 22
- signing\_ca\_label (variable in IKYSETUP) 38
- SignWith (named field in pkiserv.tmpl) 98
- Simple Certificate Enrollment Protocol (SCEP)
  - certificate
    - description 103
    - description of template 103
    - template description 103
  - certificate fields 113
  - enabling 175
  - enabling in pkiserv.conf 55
  - fingerprint checking 178
  - overview 175
  - preregistration 176
    - overview 175
  - RA certificate
    - creating using IKYSETUP 27
  - record 176
  - request processing, overview 176
  - request states 176
  - steps for enabling 178
  - variables 176
- single certificate, processing 232
- single request, processing 222
- skill requirements 11
- skills
  - ICSF programmer 12
  - installing PKI Services 13
  - installing prerequisite products 12
  - LDAP programmer 12, 13
  - MVS programmer 13
  - OCEP programmer 12
  - OCSF programmer 12
  - RACF administrator 13
  - UNIX programmer 12, 13
  - Web server programmer 12, 13, 14
- smart cards 3
- SMP/E 9
- SMS base configuration 79
- space considerations
  - for ICL 78
  - for object store 78
  - for VSAM data sets 77
- SPANNED statements 80
- square brackets (in substitution variables) 91
- SSL
  - certificate
    - creating 27, 356
    - use 7
  - delivering certificates through 3
  - enabled 85
  - supported standards 6
  - two modes 67
  - with client authentication operating mode of z/OS
    - HTTP Server 355
  - without client authentication operating mode of z/OS
    - HTTP Server 355
- sslclientauth directive 69
- sslmode directive 68, 69
- sslport directive 68, 69
- SSLring 31
- SSLX500CARoots directive 69
- SSLX500Host directive 69
- SSLX500Password directive 69
- SSLX500Port directive 69
- SSLX500UserID directive 69
- standards
  - certificate extensions supported 8
  - LDAP 7
  - public key cryptography, supported 6
- starting
  - PKI Services 85
  - PKI Services daemon 85
  - z/OS HTTP Server 70
- state (field in end-user Web pages) 203
- StateProv (named field in pkiserv.tmpl) 99
- status
  - alternate index
    - VSAM data set name for 52
- statuses
  - certificate requests 220
  - certificates 231
- STDERR\_LOGGING 348
- STDOUT 181
- EXPORT
  - preprocessing 187
- GENCERT
  - post-processing 184
  - preprocessing 183
- GENRENEW
  - post-processing 184
  - preprocessing 183
- REQCERT
  - post-processing 186
  - preprocessing 185
- REQRENEW
  - post-processing 186
  - preprocessing 185
- STDOUT\_LOGGING 348

## steps

- access to administration pages, changing 144
- accessing
  - administration home page 217
  - end-user Web pages 198
- adding
  - alternate indexes 80
  - application sections 159
  - VSAM alternate indexes 80
- adding new certificate template 132
- administering HostIdMappings extensions 243
- administration Web pages
  - changing access to 144
  - customizing 143
- alternate indexes, adding 80
- approving single request 222
- authorizing users for inquiry access 242
- bind passwords encrypted for LDAP
  - IRR.PROXY.DEFAULTS profile 261
  - LDAPBIND class 261
- building sample application 274
- CA certificate
  - renewing 249
- CA certificate profile, recovering 251
- CA certificate, locating 244
- certificate templates file
  - customization, additional first-time 126
  - customization, minimal 124
  - customizing the OtherName field 138
  - retrofitting release changes 129
- certificates, locating 244
- changing
  - administration Web pages 143
  - end-user Web pages 124, 126, 129
  - environment variables 49
  - fields in requests 222, 224
  - pkiserv.conf configuration file 51
  - runtime user ID for requesting certificates 133
  - runtime user ID for retrieving certificates 134
  - signature algorithm 149
- configuration file, updating 51, 60
- configuring
  - ICSF 22
  - LDAP 20
  - OCSF 19
  - PKITP 270
  - z/OS HTTP Server 17
- copying files
  - expiringmsg.form 47
  - pkiserv.conf 47
  - pkiserv.envvars 47
  - pkiserv.tmpl 47
  - readymsg.form 47
  - rejectmsg.form 47
- creating
  - application sections 159
  - CertificatePolicies extension 147
  - ICL data sets 79, 80
  - ICL indexes 80
  - VSAM object store 79, 80

## steps (continued)

- customizing
  - administration Web pages 143
  - distribution point CRLs 153
  - e-mail notifications 137
  - end-user Web pages 124, 126, 129
  - pkiserv.tmpl 124, 126, 129
- deleting
  - multiple certificates 234
  - selected certificates 236
  - single certificate 232
  - single request 222
- e-mail notifications, customizing 137
- enabling Simple Certificate Enrollment Protocol (SCEP) 179
- encrypted passwords for LDAP servers 261
- environment variables, updating 49
- establishing PKI Services as an intermediate CA 247
- establishing your CA and RA certificates 353
- gskkyman for certificate store 389
- HostIdMappings extensions, administering 243
- ICL data sets, creating 79
- IdenTrust compliance
  - adjusting general settings 395
  - creating templates 395
  - modifying pkiserv.conf 394
- IKYSETUP, using 39
- inquiry access, authorizing users for 242
- installing
  - ICSF 22
  - LDAP 20
  - OCSF 19
  - z/OS HTTP Server 17
- intermediate certificate authority, making PKI Services 247
- invoking the certificate validation service 274
- key ring, locating 244
- LDAP
  - schema.user.Idif, updating 65
  - section of PKI Services configuration file, tailoring 72
  - updating schema.user.Idif 65
- LDAP bind passwords, encrypted
  - IRR.PROXY.DEFAULTS profile 261
  - LDAPBIND class 261
- LDAP servers, encrypted passwords for 261
- locating
  - CA certificate 244
  - key ring 244
  - PKI Services certificates 244
  - RA certificate 244
- modifying single request 222
- passwords, encrypted LDAP binding
  - IRR.PROXY.DEFAULTS profile 261
  - LDAPBIND class 261
- performing RACF administration using IKYSETUP 39
- PKI Services certificate authority certificate, renewing 249
- PKI Services certificates, locating 244

- steps (*continued*)
  - PKI Services daemon
    - starting 85
    - stopping 86
  - PKI Services RA certificate, renewing 250
  - pkiserv.conf
    - copying 47
    - updating 51, 60
  - pkiserv.tmpl
    - copying 47
    - customization, additional first-time 126
    - customization, minimal 124
    - customizing the OtherName field 138
    - retrofitting release changes 129
  - PKITP, configuring 270
  - pkitsamp.c 274
  - preregistering a SCEP client 214
  - processing
    - multiple certificates 234
    - multiple requests through searches 227
    - selected certificates 236
    - selected requests 229
    - single certificate 232
    - single request 222
  - RA certificate
    - renewing 250
  - RA certificate, locating 244
  - RACF administration using IKYSETUP 39
  - recovering a CA certificate profile 251
  - rejecting single request 222
  - renewing
    - certificate 210
    - PKI Services certificate authority certificate 249
    - PKI Services RA certificate 250
  - requesting a certificate 204
  - retrieving a certificate
    - from bookmarked Web page 209
    - from PKI Services home page 210
  - retrofitting changes into certificate templates 129
  - revoking
    - certificate (by user) 213
    - multiple certificates 234
    - selected certificates 236
    - single certificate 232
  - RLS
    - enabling VSAM data sets for 81
    - preliminary steps for establishing 79
  - running IKYSETUP 39
  - searching for requests 227
  - sendmail configuration, testing 23
  - setting up the var directory 64
  - Simple Certificate Enrollment Protocol (SCEP)
    - enabling 179
  - starting
    - PKI Services 85
    - PKI Services daemon 85
    - z/OS HTTP Server 70
  - stopping PKI Services daemon 86
  - suspending
    - certificate (by user) 213

- steps (*continued*)
  - tailoring LDAP section of PKI Services configuration file 72
  - testing sendmail configuration 23
  - updating
    - configuration file 51, 60
    - environment variables 49
    - exit code sample 181
    - LDAP section of pkiserv.conf 75
    - pkixit.c 181
    - pkiserv.conf 51, 60
    - signature algorithm 149
    - single request 222
    - z/OS HTTP Server configuration files 67, 68
  - user ID for requesting certificates, changing 133
  - user ID for retrieving certificates, changing 134
  - using
    - gskkyman 389
    - IKYSETUP 39
  - using encrypted passwords for LDAP servers 261
    - creating IRR.PROXY.DEFAULTS profile 261
    - creating LDAPBIND class profile 261
  - var directory, setting up 64
  - viewing Web pages 85
  - VSAM alternate indexes, adding 80
  - VSAM object store, creating 79
  - z/OS HTTP Server configuration files, updating 67
- STOP command 87
- stopping
  - PKI Services 85, 86
- storage needs
  - for ICL 78
  - for object store 78
- STORCLAS statements 80
- store
  - creating 79
  - determining size requirements 78
- storing
  - certificate requests 238
  - certificate revocation lists 5
  - certificates 5
  - encrypted password information for LDAP servers 71
  - in LDAP 10
  - LDAP server encrypted password information 71
  - password for LDAP server
    - encrypted 71
  - private key
    - in ICSF 5, 10, 22
    - in RACF 5
- street (field in end-user Web pages) 203
- Street (named field in pkiserv.tmpl) 99
- subcomponent level
  - for logging 348
- subdirectory
  - bin 341
  - include 341
  - lib 342
  - PKIServ 342
  - samples 342
- SubjectAltName (certificate extension) 268

- SubjectKeyIdentifier (certificate extension) 268
- subordinate certificate authority
  - using PKI Services as 246
- subsections in certificate templates, summary 109
- substitution variables
  - base64cert 92, 94
  - browsertype 92
  - cadomain 92
  - errorinfo 92
  - iecert 92
  - optfield 92
  - pkiserv.tmpl 91
  - printablecert 92
  - tmplname 92
  - transactionid 92
- SUCCESSCONTENT subsection (in TEMPLATE section of pkiserv.tmpl) 107
- suffix
  - LDAP 65
- summary of changes xxiii
- superuser authority 47
- surrog (variable in IKYSETUP) 39
- surrog\_uid (variable in IKYSETUP) 31
- surrogate operation
  - setting up 27, 356
- surrogate user ID 132
  - creating 27
  - PKISERV 39, 352
- suspend (action for certificate) 232
- suspended (status of certificate) 231
- suspending
  - certificates
    - selected 236
- suspending certificates
  - by administrator
    - multiple 236
    - single 233
  - by user 213
- SYS1.CSSLIB 35
- SYS1.LINKLIB 35
- SYS1.LOGREC 289
- SYS1.SAMPLIB(IKYSETUP) 27
- SYSOUT
  - records, contents 298
  - viewing information 295
- sysplex enabled
  - in ObjectStore 54
- sysplex support
  - daemon, PKI Services, starting 85
  - PKI Services daemon, starting 85
  - pkiserv.conf
    - updating, overview 49
    - updating, steps for 60
  - prerequisites 9
  - requirements 9
  - RLS
    - enabling VSAM data sets for 81
    - preliminary steps for establishing 78
  - SharedVSAM
    - description 54
    - updating 61

- sysplex support (*continued*)
  - starting PKI Services daemon 85
  - updating pkiserv.conf 49
  - updating SharedVSAM 61
- system architecture diagram 5

## T

- tailoring
  - LDAP configuration 65
  - LDAP section of PKI Services configuration file 72
- task roadmaps
  - configuring PKI Services for IdenTrust
    - compliance 393
  - for implementing PKI Services 14
- TCPIP.SEZALOAD 35
- TDBM 10, 20
  - specifying password as entry 65, 68
- team members 11
- TEMPLATE section of pkiserv.tmpl
  - ADMINAPPROVE subsection 106
  - APPL subsection 105
  - CONSTANT subsection 105
  - CONTENT subsection 104
  - examining contents 117
  - FAILURECONTENT subsection 108
  - PREREGISTER subsection 109
  - RETRIEVECONTENT subsection 108
  - RETURNCERT subsection 109
  - subsections 104
  - SUCCESSCONTENT subsection 107
- templates
  - adding 132
  - customizing
    - additional first-time changes 126
    - minimal 124
    - OtherName field 138
    - retrofitting release changes 129
- testing
  - PKI Services configuration 85
- threads
  - created at initialization 58
- TID
  - alternate index
    - VSAM data set name for 52
- time interval
  - before certificate expiration 56
  - between certificate revocation lists 55, 58
  - certificate suspension grace period 56
  - for scanning for items to post 72
  - scanning database for approved requests 54
  - warning message about certificate expiration 56
- time period
  - in ICL before automatic deletion 53
  - in ObjectStore before automatic deletion 53
- TimeBetweenCRLs (parameter in pkiserv.conf) 58
- Title 99
- title (field in end-user Web pages) 203
- tmplname substitution variable 92
- TPOLICY
  - subcomponent for message logging 348



- transaction ID
  - field in administration Web pages 220
  - field in end-user Web pages 203
- TransactionId (named field in pkiserv.tmpl) 99
- transactionid substitution variable 92
- true name of certificate templates 103
- Trust Policy
  - API called CSSM\_TP\_PassThrough 270
  - overview 265
- trusting PKI Services 243
- two-year PKI Authenticcode-code signing server certificate
  - description 102
  - fields 111
- two-year PKI browser certificate for authenticating to z/OS
  - description 102
  - fields 110
- two-year PKI Windows logon certificate
  - description 102
  - fields 110
- type 80 SMF record
  - table of event codes and qualifiers 385
  - table of relocate section variable data 385
- TZ environment variable 48

## U

- unencrypted, LDAP bind passwords 260
- Uniform Resource Identifier (URI)
  - Certification Practice Statement 54
  - field in end-user Web pages 203
- UNIX programmer
  - skill planning 12, 13
  - task planning 12, 13
  - tasks
    - configuring sendmail 23
    - configuring UNIX runtime environment 45
    - LDAP section of pkiserv.conf, updating 71
    - runtime environment, configuring 45
    - sendmail, configuring 23
    - UNIX runtime environment, configuring 45
    - updating LDAP section of pkiserv.conf 71
  - team member 11
- UNIX runtime environment
  - configuring 45
- UNIX utilities
  - PKI Services 301
- unix\_sec (variable in IKYSETUP)
  - decision table 33
  - default value 36
  - description 36
- UnstructAddr (named field in pkiserv.tmpl) 99
- UPDATE access
  - IRR.DIGTCERT.ADD 353
  - IRR.RPKISERV.PKIADMIN 353
- updating
  - access to administration pages 144
  - certificate request 224
  - certificate templates file
    - customization, additional first-time 126
- updating (*continued*)
  - certificate templates file (*continued*)
    - customizing the OtherName field 138
    - minimal 124
    - retrofitting changes 129
  - configuration file
    - overview 49
    - steps 51
  - e-mail notifications 135, 137
  - environment variables
    - overview 48
    - steps 49
  - exit 181
  - expiringmsg.form 137
  - forms for e-mail notifications 135
  - IKYCVSAM 79
  - IKYMSAM 80
  - IKYRVSAM 81
  - LDAP section of pkiserv.conf 75
  - notification forms 135
  - pkixit.c 181
  - pkiserv.conf
    - overview 49
    - steps 51
  - pkiserv.tmpl
    - customization, additional first-time 126
    - customizing the OtherName field 138
    - minimal 124
  - pkitpsamp.c 274
  - readymsg.form 137
  - rejectmsg.form 137
  - runtime user ID 132
    - for requesting certificates 133
    - for retrieving certificates 134
  - signature algorithm 149
  - z/OS HTTP Server configuration files 67
- URI 203
  - containing CPS 54
- usage policy 51
- user ID
  - associating with PKI Services started procedure 27, 351
  - changing
    - requesting certificates 133
    - retrieving certificates 134
  - PKI Services daemon 59
  - runtime
    - changing 132
- user notifications
  - copying files 47
  - customizing forms 137
- UserId (named field in pkiserv.tmpl) 99
- userid directive 68, 69
- UserNoticeText1 (parameter in pkiserv.conf) 58, 148
- userPassword attribute 65, 68
- using
  - administration home page 221
  - administration Web pages 217
  - certificate policies 146
  - end-user Web pages 197
  - exit 180

## utilities

- executables 341
- PKI Services 301
  - iclview 302
  - pkiprereg 309
  - vosview 305

## V

### validating

- parameters 180

### var directory

- setting up 64

### variables

- in IKYSETUP REXX exec
  - change based on setup 31
  - change optionally 36
  - change required 29
  - configurable section 28
- in notification forms
  - %%dn%% 136
  - %%notafter%% 136
  - %%requestor%% 136
  - %%transactionid%% 136

### variables-dir 9

### verbose diagnostic messages, logging 349

### VERIFY

- accesses required 259
- R\_PKIServ function 352

### viewing

- SYSOUT information 295
- VSAM ICL data set records 303
- VSAM ObjectStore data set records 306

### virtual private network (VPN) devices 3

### VOL statements 79, 80

### vosview

- example 306
- format 305
- output 306
- parameters 305
- purpose 305

### VPN devices 3, 8

### VSAM

- data set name for ICL data 52
- data set name for ICL requestor alternate index 53
- data set name for ICL status alternate index 53
- data set name for ObjectStore base cluster 51
- data set name for ObjectStore requestor alternate index 52
- data set name for ObjectStore status alternate index 52
- data set name for ObjectStore TID alternate index 52
- RLS
  - enabling data sets for 81
  - preliminary steps for establishing 79

### VSAM alternate indexes, adding 80

### VSAM data sets

- creating 77
  - alternate indexes 377
  - not using RLS 373

### VSAM data sets *(continued)*

#### creating *(continued)*

- PATH data sets 377
- using RLS 79, 81, 380
- giving administrators access to 27
- RLS, enabling for 81

### VSAM object store, creating 79

### vsamhlq (variable in IKYSETUP) 39

## W

### warning message before certificate expiration 56

### warning messages, logging 349

### Web pages

- accessing 198, 217

### Web server

- daemon user ID WEBSRV 39

### Web server programmer

- installing and configuring z/OS HTTP Server 17
- skills 12, 13
- starting the z/OS HTTP Server 70
- tasks 12, 14

- configuring z/OS HTTP Server 17

- installing z/OS HTTP Server 17

- starting z/OS HTTP Server 70

- updating z/OS HTTP Server configuration files 67

- z/OS HTTP Server, installing and configuring 17

- z/OS HTTP Server, starting 70

- z/OS HTTP Server, updating configuration files 67

- team member 11

- updating the z/OS HTTP Server configuration files 67

### web\_dn (variable in IKYSETUP) 31

### web\_expires (variable in IKYSETUP) 39

### web\_label (variable in IKYSETUP) 39

### web\_ring (variable in IKYSETUP) 31

### webserver (variable in IKYSETUP) 39

### WEBSRV 39

### Windows logon certificate

- description 102

## X

### X.509v3 certificates 7, 8

## Y

### your name (field in end-user Web pages) 203

## Z

### z/OS

- PKI browser certificate for authenticating to
  - description 102

- fields 110

- PKI Windows logon certificate

- fields 110



- z/OS *(continued)*
  - V1R5
    - deleted information xxvii
    - moved information xxvi
    - new information xxv
    - new messages xxvi
    - updated information xxvi
  - V1R6
    - new information xxv
    - updated information xxv
  - V1R7
    - new information xxiv
    - new messages xxiv
    - updated information xxiv
  - V1R8
    - new information xxiii
    - new messages xxiii
    - updated information xxiii
- z/OS HTTP Server
  - configuration files
    - updating 67
  - configuring 17
  - description of PKI Services component 5
  - installing 17
  - operating modes PKISERV requires 355
  - PKI Services component
    - description 5
  - setting up for surrogate operation 27, 356
  - starting 70
- z/OS product libraries
  - AIEALNKE 341
  - APROCLIB 341
  - ASAMPLIB 341
  - PROCLIB 341
  - SAMPLIB 341
  - SIEALNKE 341
- z/OS UNIX level security 33
- z/OS V1R4
  - e-mail notifications
    - customizing forms 137
    - ExpireWarningTime, updating 61
    - ReadyMessageForm, updating 62
    - updating ExpireWarningTime 61
    - updating ReadyMessageForm 62
  - encrypted passwords for LDAP servers
    - LDAPBIND class profile 71
    - RACF administration 260
    - storing information for 71
    - updating LDAP section of pkiserv.conf 72
  - LDAP servers, encrypted passwords for
    - LDAPBIND class profile 71
    - RACF administration 260
    - storing information for 71
    - updating LDAP section of pkiserv.conf 72
  - notifications for users
    - customizing forms 137
  - sysplex support
    - RLS, enabling VSAM data sets for 81
    - RLS, preliminary steps for establishing 78
    - SharedVSAM, updating 61
    - updating SharedVSAM 61

zip code 203



---

# Readers' Comments — We'd Like to Hear from You

**z/OS**  
**Cryptographic Services PKI Services**  
**Guide and Reference**

**Publication No. SA22-7693-08**

**Overall, how satisfied are you with the information in this book?**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**How satisfied are you that the information in this book is:**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
Department 55JA, Mail Station P384  
2455 South Road  
Poughkeepsie, NY  
12601-5400



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold  
Along Line





Program Number: 5694-A01, 5655-G52

Printed in USA

SA22-7693-08

