IBM Z Batch Resiliency
Version 1 Release 1

User's Guide and Reference

GC27-9554-01

Note

Before using this information and the product it supports, read the information in Notices.

This edition applies to version 1, release 1 of IBM Z® Batch Resiliency (program number 5698-BR1) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: December 2020

© Copyright International Business Machines Corporation 2019, 2020.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
© Teracloud S.A. 1993, 2020.

Contents

Figures ..... **ix**

Tables ..... **xvii**

Preface ..... **xxi**
Who should read this book xxi
What this book contains xxi
Publications xxi
Accessibility xxi
Support information xxi
Contacting IBM Support xxii
Conventions used in this book xxiii
Typeface conventions xxiii
What's new in this edition xxiv

Chapter 1. Overview **1**
IBM Z Batch Resiliency Features 1

Chapter 2. Installation and Setup **5**
System requirements 5
Installing IBM Z Batch Resiliency 5
Modifying installation parameters 8
Allocating IBM Z Batch Resiliency libraries and data sets 10
Defining IBM Z Batch Resiliency GDG data set bases 11
Creating customized JCL and parameters 12
Creating IBM Z Batch Resiliency VSAM databases 13
Allocating the backup method JCL libraries 17
Setting up the system PROCLIB 19
Setting up the execution EXEC 19
Running IBM Z Batch Resiliency across multiple LPARS 20
Checking for the REXX Runtime Libraries 20
IBM Z Batch Resiliency implementation 20
APF authorizing load libraries 20
TimeLiner component setup 20

Chapter 3. Configuration **29**
Building the APPTABLE 29
Building the APPTABLE using ISPF panels 29
Building the APPTABLE by PDSE direct edit 32
Building the APPTABLE using job scheduler utilities 34
Removing or excluding data sets and jobs from the collection process 46
Removing data set names (EXCLDSN) 46
Removing job names (EXCLJOB) 47
Setting up the RT/SMF Extractor 47
Setting up the RT/SMF Extractor on multiple systems 48
Enabling the SMF exits 48
Defining Real-Time SMF subtasks (VFISTASK) for the Real-Time Dispatcher 49
Starting the Real-Time SMF Dispatcher (RTSMFP) 50
Setting up TimeLiner for ISPF 50

Configuring the IBM Z Batch Resiliency zDP interface....51

Chapter 4. SMF Data Extraction....53
The Real-Time SMF Extractor....53
SMF Extractor JCL....53
SMF record types and exit support....54
Batch SMF data collection....55
SMF Extractor parameters....56
SMF Extractor console commands....60

Chapter 5. Real-Time SMF Dispatcher....61
Real-Time Dispatcher subtasks....62
Refreshing IBM Z Batch Resiliency tables....67
Definition of test or alternate configurations for parameter sets (VFIRTD00)....67
SMF parameters (SMFPARMS)....68
Real-Time critical file analysis....69

Chapter 6. Generating IBM Z Batch Resiliency Backup JCL....73
Backup cycle....73
Backing up critical application data sets....73
BACKJCL parameters....74
BKUPJCL Exits....75
Defining backup methodologies....75
Adding data sets to the Include list....75
Load balancing backups using application splitting....80
Backup methodology utilities....80
Backup Dynamic Data set Scheduling....97

Chapter 7. Monitoring Non-IBM Z Batch Resiliency Generated Backups....111
Configuring IBM Z Batch Resiliency to monitor non-IBM Z Batch Resiliency generated backups....111
Building the LEGTABLE....111
Setting up SCANSMF....114
Adding data to the Backup Table for non-standard utilities....116
Activating SCANSMF diagnostic output....117
Setting up SCANSMF for utilities that do not produce SMF records....118
How SCANSMF produces SMF records for data movers....119
Modifying interface modules for ADRDSSU or FDR/FDRDSF....120
Tracking backups (SMFDSSU)....122
Listing options (VDUOPT)....122
Enabling the VFIDYNEX dynamic exit....123
Adding DSS or FDR backup entries to BKUPTBL....125
Adding DSS full volume backup entries to BKUPTBL (VFIFDSS)....125
Adding FDR backup entries to BKUPTBL....127
Generating JCL for Full Volume Backups (BKUPFULL)....128
INCLVOLS Format and Fields....129

Chapter 8. Virtual Recovery Simulator....131
Activating the Virtual Recovery Simulator subtask....131
Starting the Virtual Recovery Simulator....133
Starting the Virtual Recovery Simulator as an RT/SMF subtask....133
Starting the Virtual Recovery Simulator from the Command Processor....133
Stopping the Virtual Recovery Simulator....134
Mirror Volume Support....134
Reporting....135
Virtual Recovery Simulator Reports....135
Virtual Recovery Simulator (Report 1)....135
DSN Recovery Report (Report 2)....136

DSN Recover Report (Report 3)..... 136

Chapter 9. IBM Z Batch Resiliency Maintenance.....139
Maintaining the Inventory database (INVMAINT)..... 139
Running Inventory maintenance as a batch job (INVMAINT).....139
Setting up R/T Inventory maintenance within the Real-Time started task (RTINVPRM)..... 140
Scheduling Real-Time Inventory maintenance using ICP.....141
Removing specific data sets from the Inventory database (INVREMOV).....142
Maintaining the Backup Table (BKUPMNT).....145
Setting up maintenance for the Backup Table..... 145
BKUPMNT processing options.....146
Maintaining the REPORTS data set (RPTMAINT)..... 147
RPTMAINT options.....148
Maintaining the HISTORY and AUDIT data sets (HISMAINT)..... 148
Changing the retention period for the HISTORY data set..... 148
Changing the retention period for the AUDIT data set.....149
Critical file auditing..... 149
Maintaining the Keyed History databases 149
Defining Real-Time Keyed History data collection.....149
Defining Keyed History data collection using a batch job.....150
Running the Keyed History maintenance jobs..... 150
Updating the critical file list (MAINTLST).....152
Building and updating the critical file list (MAINTLST)..... 152
MAINTLST parameters.....153
MANTLSTP and MANTLSTN Reports 155
Removing obsolete records (MANTREMV) 156
Defining exclusions..... 156
Running the MAINTLST program using the IBM Z Batch Resiliency ISPF Menu..... 157
Reporting non-critical data sets..... 157
Maintaining the InstaSnap zDP audit data set (ZDPMAINT) 157

Chapter 10. IBM Z Batch Resiliency Utilities..... 159
MEMPLATE – Grouping data sets by patterns..... 159
Activating MEMPLATE Processing..... 159
MEMPLATE Definition Rules.....160
DB2 – Auditing the Db2 environment..... 161
DB2READ 161
DB2BACK.....163
IMS – Auditing the IMS environment.....164
BRNVGENR – Using the high-speed tape copy utility 164
BRNMGJBB – Backing up DFSMShsm migrated data sets.....165
BRNMGJBB Options.....165
BRNMGJBB Considerations.....166
MOVEJCL – Moving critical data sets to other volumes..... 168
MOVEJCL data sets..... 168
BRNVCAT8 – INCLDSN testing utility.....171
CRITREMV – Removing critical data sets entries..... 171
BKUPRELD, BKUPRBLD - Recovering a corrupted Backup Table.....172
Reloading the Backup Table (BKUPRELD).....172
Recreating entries in the Backup Table (BKUPRBLD)..... 172
IBM z/OS Dataset Mobility Facility Integration Installation and Use.....172
Installation and Customization.....173
Using the interface to IBM z/OS Dataset Mobility Facility..... 173
Generating the JCL and output reports..... 173
Inventory Data Set Report..... 174
Resolving Application Assignments for Data Set Encryption.....175

Chapter 11. Viewing IBM Z Batch Resiliency Databases and Reports..... 177
Viewing IBM Z Batch Resiliency databases using ISPF....177
Accessing the ISPF Panels....177
Viewing the Critical File List....178
Viewing the Backup Table....179
Viewing the Inventory database....181
Viewing History information.... 183
Displaying Backup Dynamic Dataset Scheduling information....185
Generating IBM Z Batch Resiliency reports....187
Accessing reports using ISPF panels....188
Outputting reports using BRNYPRNT....188
Creating a CSV data set via the IBM Z Batch Resiliency ISPF panels 189
Downloading the sample Excel spreadsheet reports from the IZBR installation package 191
Updating your spreadsheet reports 192
Jobs report (JOBRPT)....193
Critical data set report (CRITREPT).... 194
HISTORY report (HISTRPT).... 194
Concatenated files report (CONCATRP)....195
Cross dependencies report (CROSSDEP)....195
Inventory records report (INVTYRPT)....195
Cycle management report (CYCLEMGT).... 196
Dormant report (DORMANT).... 197

Chapter 12. Restoring Critical Data Sets..... 199
Restoring specific data sets using ISPF panels.... 199
Restoring all critical data sets using RESTJCL.... 200
Recovery job flow process.... 201
Generating the RESTJCL.... 201
Restoring specific applications or data sets using the XRESTJCL Exit.... 201
Deleting critical files ($DELETE$).... 202
Building a data set previously restored using RESTTBL.... 203
Creating jobs to allocate data sets (RESTALC).... 203
Backup and restore methodologies.... 204
Restoring DASD data sets backed up by DF/DSS.... 204
Restoring online DASD data sets backed up by ABARS.... 205
Restoring DASD data sets backed up by FDR/ABR.... 207
Restoring VSAM data sets backed up by FAVER.... 208
Restoring DASD data sets backed up by CA Disk (DMS/OS) 209
Restoring data sets from BRNVCPY backups 210
Restoring data sets for TMS (CA-1).... 210
Restoring data sets for CA Dynam/TLMS.... 210
Restoring data sets for Control-M/Tape (CTT).... 211
Restoring data sets for RMM....212
Restoring data sets for ASG-Zara.... 212

Chapter 13. Using TimeLiner..... 215
TimeLiner overview....215
Using Log Stream information.... 215
About the zDP Restore interface.... 215
Starting TimeLiner.... 216
Using the TimeLiner Job Record Viewer.... 217
Recovering zDP snapsets.... 218
Generate a cascade report from a selected job, step or data set.... 219
Generate a Safeguarded Copy report....220
TimeLiner maintenance tasks.... 221
TimeLiner messages.... 221

TimeLiner commands....221
Stopping TimeLiner....222
Reloading the Application Table....223
Reloading the SVC or SMF exit....224

Appendix A. IBM Z Batch Resiliency Parameters....225

Appendix B. Data Sets and Libraries....229

Appendix C. USERMAC Member Description....233

Appendix D. IBM Z Batch Resiliency Defined Enqueues....245

Appendix E. BRNVCPY Alternate Return Code Table....247
How to Change TCALTTBL....247

Appendix F. SMF.MANx Data Sets and System Logger....251
Real-Time SMF Extraction....251
Running Multiple SMF Extractors....252
Example: Running Multiple SMF Extractors....253
SMFEXT - LOGR Support....254

Appendix G. Disaster Recovery Planning Process Overview....257
Step 1: Obtain Top Management Commitment....257
Step 2: Develop Project Plan....258
Step 3: Perform Risk Analysis....258
Step 4: Perform Business Impact Analysis....258
Step 5: Develop Disaster Recovery Plan....259
Step 6: Test the Plan....262
Step 7: Maintain Plan....262

Appendix H. IBM Z Batch Resiliency Messages....263
ARCSKEXT – ABARS Backup and Restore Interface Messages....263
Backup Grouping Messages....268
Backup Grouping Error Codes....275
BRNYPRNT Messages....277
Cycle Management Messages....279
In-Flight Processing Messages....280
INVMAINT Messages....282
BRNVSPLT Messages....291
BRNADHSM Messages....292
Real-Time Dispatcher Messages....293
REQ$TASK Messages....321
RPTMAINT Messages....322
SMFEXT Messages....326
SCANSMF Messages....328
Simulated Recovery Messages....334
SMFDSSU Messages....336
BRNVB020 Messages....343
BRNVB031 Messages....343
BRNVB032 (TABLEVFY) Messages....344
BRNVB034 Messages....345
BRNVB044 Messages....346
BRNVBCAT8 Messages....346
BRNVBDMS Messages....350
BRNVBDSS Messages....350

BRNVBFDR Messages....352
BRNVBFVR Messages....356
BRNVYNEX Messages....356
BRNVLEGX Messages....359
BRNVGENR Messages....360
HISMAINT Messages....364
IZBR/RTS Messages....365
BRNVSAMI Messages....376
BRNVSAMU Messages....378
ICP Messages....379
TimeLiner Messages....380

Appendix I. IBM Z Batch Resiliency Abend Codes....401
RTSMF Abend Codes....401
SMF Extractor Abend Codes....407
VFIGENR/VTCCMP Abend Codes....408
Backup Grouping (BGMAIN) Abend Codes....408
DC/Optimize Abend Codes....408
IZBR-TLMS Abend Codes....408
BRNVSAMX Abend Codes....409
Batch Utility Abend Codes....410
Miscellaneous IBM Z Batch Resiliency Abend Codes....411

Notices....413
Trademarks....414
Glossary....415

Index....417

Figures

1. IBM Z Batch Resiliency Overview..... 1

2. Menu Options View Utilities..... 6

3. IBM Z Batch Resiliency Primary Panel.....6

4. IBM Z Batch Resiliency Installation Panel.....6

5. Site Information panel.....8

6. Names of CTT Files panel.....8

7. High-level qualifier option.....9

8. File Definition Panel.....9

9. IBM Z Batch Resiliency Allocation Panel.....10

10. Data Sets to be allocated option.....10

11. Data Sets to allocate option.....11

12. Generate IBM Z Batch Resiliency JCL panel..... 12

13. Operational Status Display panel..... 13

14. Create VSAM Databases panel..... 13

15. Create Inventory database JCL..... 14

16. Create Backup Table JCL..... 15

17. Create Backup Audit file JCL..... 16

18. Create CMDQUEUE data set JCL.....17

19. Allocate Files for Backup Methods panel.....18

20. Backup Methodology Panel..... 18

21. Choose from Backup Method Files panel.....19

22. IBM Z Batch Resiliency Application Panel..... 30

23. IBM Z Batch Resiliency Job Edit Panel.....30

24. ISV Scheduling Utilities....35
25. Input to the ESPJOB utility....41
26. Output from the ESPJOB utility....42
27. Step JS40 contains the APPTABLE update....42
28. Modifying USERMAC(EXCLJOB)....47
29. SMF Extractor and SMFXU83 exit job flow....53
30. SMF Extractor started task....54
31. Sample batch collection process JCL....56
32. Example of the Real-Time SMF Dispatcher RTSMFP JCL....61
33. VFISTASK member....62
34. Sample DISPLAY SUBTASKS report....67
35. VFIRTD00 Real-Time Startup parameters....67
36. Backup cycle flowchart....73
37. BACKJCL program....74
38. VFIMEMDL JCL....79
39. BRNVFMMC JCL....79
40. APPLDEF JCL....80
41. BKUPJCL for DF/DSS....80
42. CNTL(XDSSBKUP)....81
43. ISPF Primary Option Menu for z/OS DFSMS....82
44. Aggregate Group Application Selection panel....83
45. Aggregate Group Define panel....83
46. Aggregate Group Define panel (Page 1 of 2)....84
47. Aggregate Group Define panel (Page 2 of 2)....84
48. ABARVFY JCL sample....85

49. ABARVFY JCL 86
50. BKUPBARS JCL 87
51. BKUPJCL for FDR/ABR 88
52. FDR/ABR procedure 89
53. BKUPJCL for FAVER 90
54. APPLID XFVRBKUP procedure 90
55. BKUPJCL for CA-DISK DMS OS 91
56. CA-DISK DMS OS XDMSBKUP procedure 92
57. JCL Created in BKUPJCL BRNVCPY 93
58. Sample JCL for backing up critical tapes 94
59. SCHDREQ member sample 100
60. DISPLAY STATUS command example 104
61. DISPLAY ARRAYS command example 104
62. DUMP IRKA command example 105
63. VFISUB JCL sample 106
64. VFISUBR JCL sample 107
65. Backup Dynamic Data set Scheduling Batch 107
66. VFIMEMDL JCL Sample 108
67. Create Application Data set Backup jobs 108
68. Change APPL01 in CNTL(VFISUB) to reflect the relevant Application Name 108
69. Add CNTL(VFISUB) JCL 109
70. VFITABLS member 114
71. RTSSPPRM member 114
72. Sample LEGTABLE 118
73. Sample FDRZAPOP to Enable Exits 119

74. VFIDYNEX JCL Example....123
75. VRSALLOC job....131
76. Virtual Recovery Simulator Report 1....136
77. Virtual Recovery Simulator Report 2....136
78. Virtual Recovery Simulator Report 3....137
79. RTINVPRM Real-Time Inventory maintenance....140
80. RTDICPIN Sample....141
81. KEYLOAD JCL Member....150
82. Keyed History maintenance job deletion selection criteria 1....151
83. Keyed History maintenance job deletion selection criteria 2....151
84. Critical File Report....153
85. MAINTLST Update Critical File Report....153
86. Critical File Report by Data Set, sample report....155
87. MAINTLST Sample....157
88. BRNSWEPT Successful Execution Example....162
89. BRNSWEPC Successful Execution Example....162
90. DB2READ Report Sample....162
91. Db2 Bind Error Output....163
92. DB2BACK sample report 1....163
93. DB2BACK sample report 2....163
94. DB2BACK sample report 3....164
95. DB2BACK sample report 4....164
96. BRNVGENR JCL sample....164
97. Sample JCL to execute BRNMGJBB....166
98. JCL Generated by BRNMGJBB....167

99. BRNVCAT8 utility 171

100. INVREPT sample report 175

101. IBM Z Batch Resiliency Primary Panel 178

102. IBM Z Batch Resiliency Online Inquiry Menu 178

103. IBM Z Batch Resiliency Online Inquiry Menu 179

104. IBM Z Batch Resiliency Online Inquiry Menu 180

105. Backup Table Filter 180

106. IBM Z Batch Resiliency Backup Table Display 181

107. IBM Z Batch Resiliency Backup Detail Viewer 181

108. IBM Z Batch Resiliency Online Inquiry Menu panel 182

109. Inventory Component Table Filter panel 182

110. Inventory Component Detail Viewer panel 183

111. Online Inquiry Menu 184

112. Data Set History Table Filter panel 184

113. Job History Data Set Display 184

114. History Data Record Display 185

115. Online Inquiry Menu 185

116. Dynamic Data Set Scheduling Filters panel 186

117. Dynamic Data Set Scheduling Records panel 186

118. Dynamic Scheduling Component Record panel 187

119. Reports panel 188

120. IBM Z Batch Resiliency Online Inquiry Menu 189

121. Inventory Component Table Filter 190

122. Export Tool 190

123. View Display for SBRNUMAC 191

124. Action for Member BRNGRAPH....191
125. Spreadsheet report example....193
126. JOBRPT report....194
127. Critical data sets by APPLID sample report....194
128. CROSSDEP report....195
129. INVTYRPT JCL....196
130. DORMANT sample report....197
131. Backup Table Filter panel....199
132. Backup Table Display....200
133. IBM Z Batch Resiliency Restore JCL Generator....200
134. RESTJCL example....200
135. Recovery job flow process diagram....201
136. JCL to run RESTALC....204
137. RESTJCL for DF/DSS....205
138. YBARS Procedure....206
139. JCL RESTJCL for FDR/ABR....208
140. YFDRREST Procedure....208
141. RESTJCL generated for FAVER....209
142. TimeLiner Started Task....216
143. Job panel showing steps of job in tree table....217
144. TimeLiner Snapsets....219
145. TimeLiner zDP Custom Recovery....219
146. DSACTIVE Safeguarded Copy JCL....220
147. Safeguarded Copy report....221
148. USERMAC(VFIPARMS) with default values....225

149. HLQ.USERMAC(TCALTTBL) with alternate return codes..... 248

150. SMF.MANx data sets and system logger..... 251

151. CKKIALLO example.....252

152. Sample IBM Z Batch Resiliency log job.....254

Tables

1. Installation worksheet..... 7

2. Job run TimeLiner commands..... 22

3. TimeLiner parameters..... 26

4. APPTABLE format and field descriptions..... 33

5. Overriding job masks..... 34

6. Multiple APPLIDs..... 34

7. Job scheduler interface programs provided with IBM Z Batch Resiliency..... 35

8. OPCJOB parameters..... 43

9. Available XTWSJOB1 exit variables..... 44

10. IZWS APPLIDs / APPLID job assignment differences..... 44

11. SMF record types used by basic RT/SMF subtasks..... 49

12. SMF record types..... 54

13. SMFPARMS options..... 56

14. R/T SMF Dispatcher subtasks..... 62

15. RTCMDB job parameters..... 66

16. SMFPARMS parameters..... 68

17. RTSMFP DD statements..... 69

18. BACKJCL parameters..... 74

19. Include list record syntax..... 76

20. DF/DSS information..... 80

21. ABARS information..... 82

22. ABARVFY options..... 85

23. FDR/ABR information..... 88

24. Backing up VSAM to tape....89
25. Backing up online DASD....91
26. XTAPECPY defaults....94
27. Creating OUTTAPE (TAPEPARS)....95
28. IZBR/CKK Startup parameters....98
29. SCHDREQ parameters....100
30. Coding SCHDREQ parameters....102
31. Example SCHDREW statement scheduling criteria....103
32. Backup dynamic Data set scheduler commands....104
33. Parameters for invoking VFISUB....105
34. Files used by VFISUB....105
35. LEGTABLE entries....112
36. ADRDSSU options....120
37. VFIFDSS parameters....126
38. BRNVIFDR execution parameters and default values....127
39. BKUPFULL parameters....128
40. INCLVOLS format and fields....129
41. Virtual recovery simulator parameters....132
42. RTDICPIN parameters....142
43. DSB, DSN, DSNAME parameters....144
44. BKUPMNT processing options....146
45. RPTMAINT options....148
46. MAINTLST parameters....154
47. Place holder characters for MEMPLATE definition rules....160
48. Default parameter values for BRNMGJBB options....165

49. MOVEJCL data sets....168
50. INCLMOVE parameters....169
51. MOVEJCL options....171
52. HLQ.CNTL(ZDMFIZBR) output....173
53. HLQ.CNTL(ZDMFBSIZ) report output....174
54. Inventory data set report....174
55. Report JCL descriptions....187
56. COPYPRINT options....188
57. BRNYPRNT PARM values....189
58. MODE option parameters....202
59. RESTJCL.ABARS members....205
60. RESTBARS parameters....206
61. Information for FDR/ABR....207
62. RESTJCL.FAVER members....209
63. DMS information....209
64. TimeLiner start parameters....216
65. TimeLiner messages....221
66. TimeLiner commands....222
67. VFIPARMS parameters and values....225
68. Data sets and libraries....229
69. USERMAC member descriptions....233
70. IBM Z Batch Resiliency defined enqueues....245
71. Default return codes....247
72. Backup grouping error codes....275
73. Message ID format....282

74. INVSPLT messages..... 291

75. RTSMF abend codes..... 401

76. SMF Extractor abend codes..... 407

77. VFIGENR/VTCCMP abend codes..... 408

78. BGMAIN abend codes..... 408

79. DC/Optimize abend codes..... 408

80. TLMS abend codes..... 408

81. VFIVSAMX abend codes..... 409

82. Batch utility abend codes..... 410

83. Miscellaneous abend codes..... 411

Preface

IBM Z Batch Resiliency protects data within IBM z/OS environments ensuring continued operational resiliency and availability of critical business applications through automated backup and recovery processes.

Who should read this book

The *IBM Z Batch Resiliency User Guide and Reference* is to be used by your System Administrator to perform the customization, and administration of the system.

It is also to be used by your System Operators for general daily usage.

Readers should be familiar with the following:

- Db2® and its utilities
- Time Sharing Option Extensions (TSO/E)
- Restructured Extended Executor (REXX) language
- Job control language (JCL)
- Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) and its dialog manager functions

What this book contains

This book contains information about how to install, maintain, and use IBM Z Batch Resiliency.

Publications

IBM posts publications for this and other products, as they become available and whenever they are updated, to the IBM Knowledge Center website:

https://www.ibm.com/support/knowledgecenter/en/SSY3NH

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, refer to the IBM Accessibility website:

https://www.ibm.com/accessibility

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

© Copyright IBM Corp. 2019, 2020

Contacting IBM Support

This topic describes how to contact IBM Support if you have been unable to resolve a problem with IBM Z Batch Resiliency.

Before contacting IBM Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have. For more information, refer to the IBM Support website at the following links:

IBM Support
https://www.ibm.com/mysupport/s/

IBM Z Support
https://www.ibm.com/support/pages/ibm-enterprise-support-and-preferred-care-options-ibm-z

To contact IBM Support to report a problem (*open a case*), follow these steps:

1. Determine the business impact.
2. Describe the problem and gather information.
3. Submit the problem report.

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1
The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3
The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing the problem and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Support specialists can help you solve the problem efficiently. To save time, know the answers to the following questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, product-specific customization, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting the problem

You can submit your problem to IBM Support in either of the following ways:

Online

Go to https://www.ibm.com/mysupport/s/, click **Open a case**, and enter the relevant details into the online form.

By email or phone

For the contact details in your country, go to the IBM Support website at https://www.ibm.com/support/. Look for the tab on the right and click **Contact and feedback** > **Directory of worldwide contacts** for a list of countries by geographic region. Select your country to find the contact details for general inquiries, Technical Support, and customer support.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website, so that other users who experience the same problem can benefit from the same resolution.

Conventions used in this book

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

The following terms are used interchangeably throughout this book:

- MVS, OS/390®, and z/OS.
- VM and z/VM.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip**, and **Operating system considerations**)
- Column headings in a table
- Keywords and parameters in text

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

What's new in this edition

The changes in this edition relate to IBM Z Batch Resiliency V1.1.0 new functions and enhancements.

This edition is an update of the previous edition of the same book. Changes in this edition include:

- New SMF Extractor features have been documented in “SMF Extractor JCL” on page 53
- The “Maintaining the InstaSnap zDP audit data set (ZDPMAINT) ” on page 157 section has been added, outlining the procedure to remove obsolete records from the TimeLiner zDP Audit data set
- A new spreadsheet templates feature is documented in:
  - “Creating a CSV data set via the IBM Z Batch Resiliency ISPF panels ” on page 189
  - “Downloading the sample Excel spreadsheet reports from the IZBR installation package ” on page 191
  - “Updating your spreadsheet reports ” on page 192
- New graph generation capability is documented in “Updating your spreadsheet reports ” on page 192
- The “Inventory records report (INVTYRPT)” on page 195 section has been added, outlining the options for the IBM Z Batch Resiliency IBM z/OS Dataset Mobility Facility Integration INVREPT report
- The new Cascade report has been documented in “Using the TimeLiner Job Record Viewer” on page 217 and “Generate a cascade report from a selected job, step or data set” on page 219
- The new Safeguarded Copy report has been documented in “Generate a Safeguarded Copy report” on page 220
- SMF Extractor messages have been further updated
- TimeLiner commands have been documented in “TimeLiner commands” on page 221
- General editorial changes throughout

Technical updates are marked in the PDF with a vertical bar to the left of the change. Editorial changes are not marked.

Changes in the previous edition

Changes in the previous edition included:

- Restructure of the book and reorder of a number of chapters:
  - Chapter 6, “Generating IBM Z Batch Resiliency Backup JCL,” on page 73
  - New topic in Chapter 6, “Generating IBM Z Batch Resiliency Backup JCL,” on page 73 on the “Backup Dynamic Data set Scheduling batch process” on page 107
  - Chapter 8, “Virtual Recovery Simulator,” on page 131
  - Chapter 9, “IBM Z Batch Resiliency Maintenance,” on page 139
  - Chapter 13, “Using TimeLiner,” on page 215
- IZBR zDMF integration enhancements
- TimeLiner messages have been updated and explained
- SMF Extractor messages have been updated
- The Index has been generated

Changes from the previous edition are not marked.

Chapter 1. Overview

IBM Z Batch Resiliency is intended for operations staff to improve the resilience of the installation's batch processes.

Improved resilience is accomplished by monitoring processing to ensure that critical backups are completed before any data sets are used. Then, should a data corruption occur, IBM Z Batch Resiliency is able to report on which jobs created and read the relevant data set, and assist with the recovery of the corrupted data set.

The jobs that the product needs to know about are generally those being submitted by the installation's batch job scheduler. The program extracts the required information from the scheduler's data sets. There is, of course, the ability to manually edit the list of jobs.

After this is done, the critical data sets need to be identified; that is, the ones that are created by one job and then used by subsequent jobs. This is done via analysis of SMF data, either historical or in real-time. Once the data sets are identified, the product monitors them using real-time SMF analysis of both the batch jobs and the backup utilities; the former to keep track of which jobs have used the critical data set and the latter to maintain a list of current backups for any possible recovery.

One final part of the product is known as TimeLiner. This component dynamically monitors the selected jobs and keeps track of which data sets are being used during execution. This is done to enhance resilience by capturing the data in flight; so that if the job fails, the program is aware of which data sets were in use at the time. TimeLiner records its data in storage and replicates that into a journal, both of which are searchable online.

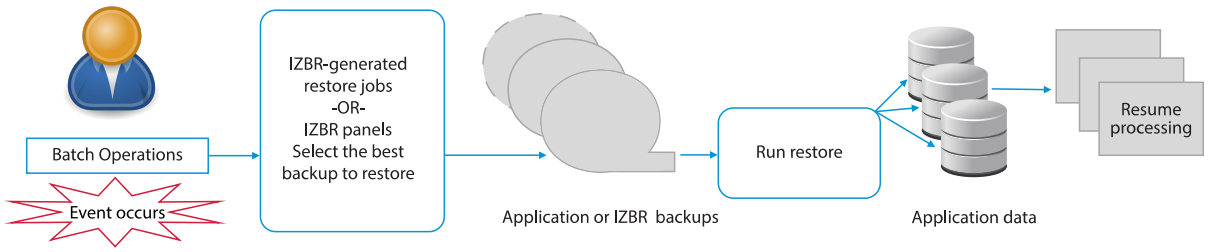


Figure 1. IBM Z Batch Resiliency Overview

Data consistency is provided at the recovery site. The database management subsystem restart takes care of backing out in-flight transactions ready to commence online transaction processing (OLTP) workloads.

IBM Z Batch Resiliency gives you journal-like recovery for batch. It provides an auditable, actionable, repeatable recovery process that can help to prove your compliance because it provides a three-pronged data management approach to know:

- How the data is used
- Who is using it
- Where the data is located

IBM Z Batch Resiliency Features

IBM Z Batch Resiliency protects data within IBM® z/OS® environments, ensuring the continued operational resiliency and availability of critical business applications through automated backup and recovery processes.

Identifies critical application data sets for backup

IBM Z Batch Resiliency uses advanced analysis of SMF data to determine the criticality of application data sets requiring backup. The SMF extractor continuously collects SMF data from multiple LPARs and the

© Copyright IBM Corp. 2019, 2020

Real-Time SMF Dispatcher analyzes SMF data from multiple LPARs to determine which applications are critical to your business operations.

IBM Z Batch Resiliency backup features offer the following benefits:

- Comprehensive backups of critical files based on Real-Time SMF analysis determinations.
- Multiple backup methodologies from different backup utilities. For example, you can use the DF/DSS utility to back up critical sequential data and use FAVER to back up VSAM data.
- Updates to the changed date/time in the Backup Table for non-VSAM DASD data sets that are modified to reflect whether a tracked data set still has a valid backup. Both full-volume and logical backups are tracked to allow restore of data at the data set level, application level, or volume level.
- Aggregate Backup and Recovery Support (ABARS) processing including parameter generation, ABARS verification, invocation processes, and log reading utilities.
- Monitoring of backups generated with utilities such as IBM IDCAMS, IEBGENER and IEBCOPY, using the SCANSMF utility. SCANSMFupdates the backup table to provide a comprehensive inventory of all available data set backups.
- An optional feature, Backup Dynamic Dataset Scheduling, that reduces backup windows by scheduling backups of critical data sets to be run at intervals throughout the day.
- For non-virtual tape installations, BRNVCPY copies critical tapes to backup volumes for offsite rotation. BRNVCPY simplifies the backup of critical tapes by using standard copy utilities such as IEBGENER.

Automated recovery of critical data sets

In addition to the templates that are provided for standard utilities, such as DFSMSdss and IDCAMS, you can create JCL templates for any methodology, including internally written utilities.

IBM Z Batch Resiliency provides automated recovery of critical application data sets to minimize overall application downtime and quickly resume full application operations.

If you need to restore a data set, you can use the ISPF interface which identifies the appropriate backup that is needed, and automatically generates the restore JCL to match the original backup method.

TimeLiner feature

The IBM Z Batch Resiliency TimeLiner feature monitors operational production job and data set usage while jobs are processing. Using TimeLiner you can identify potential corrupted data sets by date and time of the event, to help determine the point-in-time recovery period and quickly return an application to full production.

TimeLiner also integrates with the Dell EMC Data Protector for Z Systems zDP™ product SnapVX feature providing automated management of zDP snapsets.

Comprehensive reporting

You can access IBM Z Batch Resiliency database information and generate reports that detail the usage and relationship of your batch application data. The IBM Z Batch Resiliency ISPF interface provides access to comprehensive and updated database information and reports. Key reports identify resiliency and audit gaps to address compliance mandates by automatically providing audit reporting for recovery and data sharing regulations.

Using IBM Z Batch Resiliency reports you can identify:

- Critical data sets that are not being backed up
- Critical data sets that are shared between production and non-production jobs
- Duplication of backups
- Application data dependencies
- Db2 Database image copies audit

- IMS Database backup audit
- Data sets impacted downstream of failed jobs

Maintenance utilities

IBM Z Batch Resiliency provides a series of utilities to maintain the list of critical data sets contained in the Inventory database, backups of all critical data sets and where they reside in the backup table, as well as reporting, history and auditing data maintenance. To ensure the databases are maintained and contain up-to-date information, you can set up a daily batch process that runs maintenance to ensure the integrity of key IBM Z Batch Resiliency databases.

Chapter 2. Installation and Setup

Before you can use IBM Z Batch Resiliency, you must install and set it up.

System requirements

Before you install IBM Z Batch Resiliency, be sure that the following system requirements are met:

- z/OS V2R2 (and later)
- Access to the system catalog
- A tape management system (CA-1®, CA Dynam/TLMS, Control-M/Tape (previously known as Control-T), RMM, or ASG-Zara™)
- A DASD Backup/Restore system (for example, ABARS, DFDSS, FDR, DMS)

Installing IBM Z Batch Resiliency

A customized JCL control library is needed to run IBM Z Batch Resiliency in batch mode. This library is created during the interactive installation. The customized JCL is generated automatically.

Before you begin

Complete the installation process described in the *Program Directory for IBM Z Batch Resiliency Infrastructure V1.1.0* to install the IBM Z Batch Resiliency infrastructure.

About this task

To begin the interactive installation of IBM Z Batch Resiliency, perform the following steps.

Procedure

1. Access ISPF selection 3.4.
2. At the `DSNAME LEVEL ===>` prompt, enter the high-level qualifier you chose and press the Enter key.
   A listing of all data sets beginning with that high-level qualifier is displayed.
3. On the line that contains *hlq*.SBRNUMAC, enter the following command in column 1:
   `EX /(BRNUDIST)`
   This command creates the *hlq*.DIST.LIBRARY data set.
4. On the line that contains *hlq*.SBRNUMAC, enter the following command in column 1:
   `EX /(BRNUUMAC)`
   This command creates the *hlq*.USERMAC data set.
5. On the line that contains *hlq*.SBRNEXEC, enter the following command beginning in column 1:
   `EX /(BRNVSTRT)`
   Your screen should look similar to the following figure.

© Copyright IBM Corp. 2019, 2020

```
Menu   Options   View   Utilities   Compilers   Help
------------------------------------------------------------------------------
DSLIST - Data Sets Matching Hlq.V1R1M0                        Row 32 of 39
Command ===>                                                Scroll ===> CSR
Command - Enter "/" to select action                 Message          Volume
------------------------------------------------------------------------------
         Hlq.V1R1M0.USERMAC                                           BRNV102
ex /(BRNVSTRT) V1R1M0.SBRNEXEC                                        BRNV102
         Hlq.V1R1M0.SBRNLOAD                                          BRNV101
         Hlq.V1R1M0.SBRNMLIB                                          BRNV102
         Hlq.V1R1M0.SBRNPLIB                                          BRNV102
         Hlq.V1R1M0.SBRNSLIB                                          BRNV102
         Hlq.V1R1M0.SBRNTLIB                                          BRNV102
```

Figure 2. Menu Options View Utilities

6. Press the Enter key to begin the IBM Z Batch Resiliency online session.

   This is a panel-driven session with a help facility that you can invoke by pressing the PF1 key.

   After the logo screen, the first panel displayed is the IBM Z Batch Resiliency Primary Panel.

```
---------------------------- IZBR 1.1.0 Primary Panel -------------------------
OPTION  ===>


   .  0 PROFILE       - Define IZBR Profile Information       MVSID: HBB77A0
   .  1 INSTALLATION  - Perform IZBR Installation             SYSID: SOW1
   .  2 OVERVIEW      - Review IZBR System Documentation      TIME : 07:30:30
   .  3 EXECUTE       - Execute IZBR Utilities and Reports
   .  4 REPORTS       - Access IZBR Reports Online
   .  5 MAINTENANCE   - Maintain IZBR Parameters and Tables
   .  6 INQUIRY       - IZBR Online Inquiry Functions
   .  7 MISCELLANEOUS - Miscellaneous Functions
   .  8 TLN           - TimeLiner Functions

                                          IIII  ZZZZZZ  BBBBB   RRRRR
                                          II   Z   ZZ   B  BB   R  RR
                                         II      ZZ    B  BB   R  RR
                                        II     ZZ     BBBB    RRRR
                                       II    ZZ      B  BB   R RR
                                      II   ZZ   Z   B  BB   R  RR
                                     IIII  ZZZZZZ  BBBBB   RRR  RR

     5698-BR1 Copyright (C) Teracloud S.A. 1998-2019 and IBM Corp. 2019
```

Figure 3. IBM Z Batch Resiliency Primary Panel

7. On the IBM Z Batch Resiliency Primary Panel, type 1 at the OPTION prompt and press the Enter key.

   The IBM Z Batch Resiliency Installation Panel is displayed.

```
------------------------ IZBR 1.1.0 Installation Panel ----------------------
OPTION  ===>


   .  0 OVERVIEW          - Review Installation Documentation
   .  1 MODIFY            - Modify Installation Parameters
   .  2 ALLOCATE          - Allocate IZBR Datasets
   .  3 DEFINE GDGs       - Define IZBR GDG Dataset Bases
   .  4 GENERATE JCL      - Generate IZBR Execution JCL
   .  5 CREATE VSAM FILES - Create IZBR VSAM Databases
   .  6 BKUPMETH          - Allocate Files for Backup Methods
   .  7 AUTHORIZE IZBR    - Edit and Apply IZBR Product Password
   .  A CONVERSION        - Convert files from previous versions
   .  R REXX Build        - Environment test and runtime library build
```

Figure 4. IBM Z Batch Resiliency Installation Panel

8. Optional: To preview the Installation Overview, type 0 at the OPTION prompt and press the Enter key.

   The remainder of the installation is completely panel driven, with a help facility to assist you along the way. To invoke the help facility, press PF1 from any screen.

   The following Installation Worksheet lists the parameters you must supply in the subsequent panels.

Installation Worksheet

Table 1. Installation worksheet	
Company Name (16 characters maximum)	

Installation Information

Tape Management System (For CA1 version 5 enter 'CA1'; for CA Dynam/TLMS enter 'TLMS'; for RMM enter 'RMM'; for Control-M/Tape (previously known as Control-T) enter ‘CTT’)	
File Name of Tape Catalog for Tape Management System (For CA1 use the name of the TMC; for CA Dynam/TLMS use the name of the VMF; for RMM use the name of the RMM Control File; for Control-M/Tape (previously known as Control-T) enter the names of the CTT master file)	
CTT Index File (if TMS is CTT)	
CTT Trace File (if TMS is CTT)	
Default Disk Management System to be used when generating backup and restore JCL • For DFDSS enter 'DSS' • For FDR/ABR enter 'FDR' • For ABARS enter 'ABARS' • For DMS/OS enter 'DMS'	
High-Level Qualifier (HLQ) to use when generating backup and restore JCL for DASD files	
Name of Daily SMF File. If you are planning to run in batch mode (as opposed to Real Time), provide the name of the file that holds each day's SMF data. SMF Record types required are 14, 15, 17, 30, 60, 61, 62, 63, 64, and 65. You may need to create a new set of SMF files to gather this data.	

IBM Z Batch Resiliency hlq

High-Level Qualifier for IBM Z Batch Resiliency Libraries	

Job Scheduler

Job Scheduler. Click here to select a scheduler. This is the Job Scheduling Package that is being used, for example, CA7, IZWS, or CONTROL-M.	

9. Complete each function in the order presented, using the information from the worksheets.

Modifying installation parameters

About this task

To modify installation parameters:

Procedure

1. From the IBM Z Batch Resiliency Installation Panel, select option 1, MODIFY.
2. Enter parameters and other variables used both in the installation process and for IBM Z Batch Resiliency operation.
   For example, enter your:
   - Tape and Disk Management System names
   - SMF dump file data set name
   - Tape Catalog data set name

   Refer to the Installation Worksheet for the parameters you must supply.
3. Verify parameters that are provided by the system.
   The first panel displayed prompts you for the items on the Installation Worksheet. Type in the items or accept the entries and press the PF3 key.

```
OPTION  ===>

 Please validate this information.  In addition, please enter the site
 specific information below.  Press <ENTER> to continue or <PF3> to escape.

              Company Name:  COMPANY NAME

 Site Info:
 (Dataset names are fully qualified. No quotes allowed)
    Name of daily SMF file:  SMF.&SYSNAME..BACKUP.LOG
    Tape management system:  RMM   (CA1, CTT, RMM, TLMS or ZARA)
      Tape master filename:  SYS2.RMM.CONTROL
    Disk management system:  DSS   (enter either DMS, FDR, DSS, or ABARS)
      hlq for disk backups:  IZBRA.DR              (High Level Qualifier)
```

Figure 5. Site Information panel

If your tape management system is CTT, the pop-up window prompts you for CTT index and trace.

```
                Names of CTT Files

 Master CTT file is:

         IOA.CTT.MASTER

 Enter name of CTT index file:

 INDEX:  IOA.CTT.INDEX

 Enter name of CTT trace file:

 TRACE:  IOA.CTT.TRC

    Press <ENTER> to continue or <PF3> to escape
```

Figure 6. Names of CTT Files panel

4. When the next panel prompts you to enter a default high-level qualifier for IBM Z Batch Resiliency to use in allocating the remaining data sets, add a new high-level qualifier, or use the qualifier provided.
   - Type in a high-level qualifier and press the Enter key. In certain cases, such as when you are distributing IBM Z Batch Resiliency to multiple remote sites, you may want to have different high-level qualifiers for each installation, and another for the distribution libraries.
   - If you want to use the default name presented in the panel, press Enter.

```
OPTION  ===>

The following high level qualifier will be used to allocate the rest of
the IZBR datasets. If you want to use a different qualifier than the
one below, then type over the field.
The System ID is used to allocate and setup the IZBR SMF Extractor.
This field should be unique for each system that runs IZBR.

  High level qualifer ===> IZBRA.V110
  System ID           ===> S0W1


  Press <ENTER> to continue or <PF3> to escape
```

Figure 7. High-level qualifier option

The final panel displays the names of the remaining files that IBM Z Batch Resiliency allocates during the online installation process.

```
   ------------------------ IZBR 01.01.00 File Definition Panel ------------------
   OPTION  ===>

    Verify  IZBR                          <PF7/PF8> to scroll  Scroll ===> PAGE
    Press <PF3> to save
                                                                 More:     +
               Audit Database:  IZBRA.V110.AUDITDB
                Audit Dataset:  IZBRA.V110.AUDTDS
     Audit Level 2(GDG Base):  IZBRA.V110.AUDITL2
               Backup JCL PDS:  IZBRA.V110.BKUPJCL
      Backup Methodology Table:  IZBRA.V110.BKUPMETH

          RT/SMF Command Queue:  IZBRA.V110.RTCMDQ
            RT/SMF Check Point:  IZBRA.V110.RTACKPT

                  Backup Table:  IZBRA.V110.BKUPTBL
            Backup Table Audit:  IZBRA.V110.BKUPAUD
      Backup Unload (GDG Base):  IZBRA.V110.BKUPFLAT
    Backup AUD Unload(GDGBase):  IZBRA.V110.BAUDFLAT
         Online Keyed Datasets:  IZBRA.V110.KEYDHIST
             Online Keyed Jobs:  IZBRA.V110.KEYJHIST
      Checkpoint File (SMFEXT):  IZBRA.V110.CHKPOINT.SYSS0W1
          IZBR JCL Control PDS:  IZBRA.V110.CNTL
            Critical File List:  IZBRA.V110.CRITICAL
             DB2 Backup Copies:  IZBRA.V110.BRN2FILG
     DB2 Databases/Tablespaces:  IZBRA.V110.DB2VFICR
     IMS RECON Selected Output:  IZBRA.V110.IMS.RECONOUT
              IMS RESLIB Print:  IZBRA.V110.IMS.RESPRT
            Inventory Database:  IZBRA.V110.INVENTRY
             IZBR Load Library:  IZBRA.V110.SBRNLOAD
         IZBR STC Load Library:  IZBRA.V110.DIST.LIBRARY
               IZBR Model DSCB:  IZBRA.V110.MODEL
               ODT Check Point:  IZBRA.V110.ODTCKPT
              IZBR ISPF Panels:  IZBRA.V110.SBRNPLIB
               IZBR ISPF Execs:  IZBRA.V110.SBRNEXEC
            IZBR ISPF Messages:  IZBRA.V110.SBRNMLIB
          IZBR ISPF Skeletons:  IZBRA.V110.SBRNSLIB
             IZBR ISPF Tables:  IZBRA.V110.SBRNSLIB
       Production SMF Datasets:  IZBRA.V110.PRODDS
           IZBR Report Library:  IZBRA.V110.REPORTS
               Restore JCL PDS:  IZBRA.V110.SBRNSLIB
                Restore Tables:  IZBRA.V110.RESTTBL
     DC Optimize 2000 database:  IZBRA.V110.SCANJCL
               IZBR Source PDS:  IZBRA.V110.SBRNSRCE
           Tape Backup JCL PDS:  IZBRA.V110.TAPEJCL
            IZBR User Code PDS:  IZBRA.V110.USERMAC
```

Figure 8. File Definition Panel

5. If you want to change any individual data set names, change the names listed in the File Definition Panel.

   Note: The File Definition Panel is scrollable.
6. After making the changes, press PF3 to save and exit.

Allocating IBM Z Batch Resiliency libraries and data sets

Use ALLOCATE, option 2 of the IBM Z Batch Resiliency Installation Panel, to allocate the data sets listed in MODIFY (option 1 of the IBM Z Batch Resiliency Installation Panel).

About this task

The panel shown sets up a template allocation parameter used for all subsequent allocations.

Procedure

1. Modify the installation size, volume and, device based on your installation standards.
   - SMALL – yields an allocation capable of handling the discovery of 46,000 data sets per day.
   - MEDIUM – yields an allocation capable of handling the discovery of 92,000 data sets per day.
   - LARGE – yields an allocation capable of handling the discovery of 192,000 data sets per day.

```
------------------------- IZBR 1.1.0 Allocation Panel -----------------------
OPTION  ===>

 The following parameters will be used as a template to allocate the
 IZBR libraries. The next panel will use these parameters for all
 libraries that have not been allocated. You may change the Volume
 and Device values here for the template or selectively change the
 parameters in the next panel. You may blank out the Volume parameter
 and code an acceptable esoteric device such as SYSDA if desired.

     Installation size ===> MEDIUM   (SMALL, MEDIUM, LARGE)
     Volume            ===> BRNV102
     Device            ===> 3390


 Press <ENTER> to continue
```

Figure 9. IBM Z Batch Resiliency Allocation Panel

Note: The data sets to be allocated, along with the device and volume information, are shown in this example.

```
OPTION  ===>

 Press <ENTER> to start the allocation process, <PF3> to escape.
       CANCEL exits immediately. Use PF7/PF8 to scroll.

     -> You may update DASD Volume, Unit (generic/esoteric) and Tracks
        information below.
     -> Status field will display the results of each dataset allocation.
     -> Investigate and correct any unsuccessful allocations.

Dataset Name                                Volume  Unit      Tracks  Status
                                                                 More:     +
IZBRA.V110.AUDITDB                          SW0006  3390        1000  NOALLOC
IZBRA.V110.AUDTDS                           SW0006  3390        1000  NOALLOC
IZBRA.V110.BKUPJCL                          SW0006  3390          75  NOALLOC
IZBRA.V110.BKUPMETH                         SW0006  3390           1  NOALLOC
IZBRA.V110.CHKPOINT.SYSS0W1                 SW0006  3390           1  NOALLOC
IZBRA.V110.CNTL                             SW0006  3390          45  NOALLOC
IZBRA.V110.CRITICAL                         SW0006  3390          75  NOALLOC
IZBRA.V110.DB2VFICR                         SW0006  3390          75  NOALLOC
IZBRA.V110.DB2VFILG                         SW0006  3390          75  NOALLOC
IZBRA.V110.DMS                              SW0006  3390          15  NOALLOC
IZBRA.V110.HISTORY                          SW0006  3390        1000  NOALLOC
IZBRA.V110.HSMACT                           SW0006  3390          75  NOALLOC
```

Figure 10. Data Sets to be allocated option

Note: Verify the free space on the specified volumes before beginning this step and review the allocations afterward. The allocations for the BKUPJCL, RESTJCL, PRODDS, and HISTORY are highly specific to each individual installation.

2. Press the ENTER key to begin the allocation. (Press PF3 if you need to exit this process.)
3. Investigate and fix any unsuccessful allocations.
4. Verify that the status of each allocation at completion is ALLOC.
5. Change the VOLUME and UNIT fields as required on each individual file.
6. If any allocations are unsuccessful because of space restrictions, change the volume and process the panel again.
7. When the files have been successfully allocated, press the PF3 key to continue with the online installation.

Defining IBM Z Batch Resiliency GDG data set bases

The allocated GDG data sets are listed when you select MODIFY (option 1 of the IBM Z Batch Resiliency Installation Panel). Typically these values are set to 5 and store five days worth of data between them. You may store more data if you wish.

Procedure

1. To define GDGs, select option 3 from the IBM Z Batch Resiliency Installation Panel.

   Each data set to be allocated along with the Limit and Scratch information is displayed, as shown in the following example.

```
OPTION  ===>

 Press <ENTER> to start the allocation process, <PF3> to escape.
       CANCEL exits immediately. Use PF7/PF8 to scroll.

    -> Update Limit and Scratch fields for GDG allocation attributes.
    -> Status field will display the results of each dataset allocation.
    -> Investigate and correct any unsuccessful allocations.

  Dataset Name Base                          Limit    Scratch   Status

  IZBRA.V110.AUDITL2                         5        SCRATCH   NOALLOC
  IZBRA.V110.BKUPFLAT                        5        SCRATCH   NOALLOC
  IZBRA.V110.BAUDFLAT                        5        SCRATCH   NOALLOC
  IZBRA.V110.HISTRYL2                        5        SCRATCH   NOALLOC
```

   Figure 11. Data Sets to allocate option

2. Press the ENTER key to begin the allocation or alteration.

   Press PF3 if you need to exit this process.

3. Investigate and correct any unsuccessful allocations.

   Here is an example of unsuccessful allocation messages:

```
IKJ56893I DATA SET IZBRA.V110.CNTL NOT ALLOCATED+
IGD17273I ALLOCATION HAS FAILED FOR ALL VOLUMES SELECTED FOR DATA SET
IZBRA.V110.CNTL
IGD17277I THERE ARE (5) CANDIDATE VOLUMES OF WHICH (5) ARE ENABLED OR QUIESCED
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:SGIZBR
IGD17279I 5 VOLUMES WERE REJECTED BECAUSE OF INSUFF TOTAL SPACE
***
```

4. Ensure the status of each allocation is ALLOC at completion.
5. When you complete the allocations, press the PF3 key to continue with the online installation.

Creating customized JCL and parameters

Procedure

To generate the JCL required for the entire IBM Z Batch Resiliency process, based on the parameters you supplied, select option 4, GENERATE JCL from the IBM Z Batch Resiliency Installation Panel.

In addition to the JCL, several parameter members are generated in the CNTL and USERMAC libraries as well as the Backup Methodology (BKUPMETH) libraries.

```
------------------------- IZBR 1.1.0 Generate IZBR JCL ----------------------
OPTION  ===>


   .  1 Edit JOBCARD - Edit IZBR JOBCARD for Execution JCL

   .  2 Generate JCL - Generate IZBR Execution JCL
```

Figure 12. Generate IBM Z Batch Resiliency JCL panel

Editing JOBCARD for IBM Z Batch Resiliency

IBM Z Batch Resiliency uses the job statement to submit jobs and reports.

About this task

To edit the job card:

Procedure

1. From the Generate IBM Z Batch Resiliency JCL panel, select option 1, EDIT JOBCARD.
2. Edit the JOBCARD member with a valid job statement for your installation.
   This job statement is copied to all JCL members in the IBM Z Batch Resiliency CNTL library (hlq.CNTL).

   The supplied JOBCARD member includes a JCLLIB reference as follows:

   ```
   //IZBRPROC JCLLIB ORDER=(%CNTL%)
   ```

   The *%CNTL%* is replaced with the name of the control library when the JCL is created.
3. Press PF3 when you finish editing the JOBCARD.

What to do next

Continue the process of creating customized JCL and parameters using the information in the following section.

Generating JCL

Procedure

1. Select option '2' from the Generate IBM Z Batch Resiliency JCL panel to begin the process of generating the JCL.

   The Operational Status Display is displayed.

```
---------------------- IZBR Operational Status Display ----------------------
COMMAND  ===>

   Enter pattern or '*' for creation list or <PF3> to exit

   PATTERN  ===> *


   +---------------------------------------------------------------------+
   .                                                                     .
   .                                                                     .
   .                                                                     .
   +---------------------------------------------------------------------+
   .                                                                     .
   +---------------------------------------------------------------------+
```

Figure 13. Operational Status Display panel

2. Press Enter to begin the generation process.

   Note: You may be instructed by IBM Service to modify this field when enhancements are made to product releases.

   The pattern asterisk (*) causes IBM Z Batch Resiliency to copy all library members to their proper location.

Creating IBM Z Batch Resiliency VSAM databases

About this task

To create IBM Z Batch Resiliency VSAM databases:

Procedure

From the IBM Z Batch Resiliency Installation Panel, select Option 5.

The next panel lets you edit and submit the JCL to create the IBM Z Batch Resiliency Inventory, Backup Table, Backup Audit File, and Real-Time Command Queue VSAM databases, as shown in the following figure.

```
---------------------- IZBR 1.1.0   Create VSAM Databases  -----------------
OPTION  ===>



         Enter H next to member for Documentation
         Enter S next to member to Edit JCL
         Enter Z next to member to Submit JCL



   .  INVTDEFN  - Create VSAM INVENTORY
   .  BKUPDEFN  - Create VSAM BKUPTBL
   .  BAUDDEFN  - Create VSAM BKUPAUD
   .  CMDQDEFN  - Create VSAM RTCMDQ for RT/SMF
```

Figure 14. Create VSAM Databases panel

Defining the Inventory database

The Inventory database is a VSAM KSDS file that is allocated with the batch job CNTL(INVTDEFN).

Procedure

1. Enter S next to INVTDEFN to modify the JCL.

The JCL is then displayed.

```
//*===================================================================
//* THIS JCL IS USED TO CREATE  THE RT/SMF INVENTRY DATABASE
//*  CHANGE ALL 'VVVVVV' TO VOLSER OF DASD TO PLACE INVENTRY CLUSTER
//*===================================================================
//*
//IDCAMS1  EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSIN     DD  *
 DEFINE CLUSTER( NAME(Hlq.V1R1M0.INVENTRY)  -
                 SHAREOPTIONS(2 3)           SPEED    -
                 CYLINDERS(50 50)                     -
                 BUFFERSPACE(53248)                   -
                 KEYS(64 0)                           -
                 FREESPACE(020 010)                   -
                 RECORDSIZE(372 2000)   VOL(VVVVVV) ) -
          DATA( CISZ(08192)     BUFFERSPACE(36864) ) -
         INDEX( CISZ(02560)                         )
 REPRO INFILE(DUMMYREC) -
   OUTDATA SET(Hlq.V1R1M0.INVENTRY)
 LISTCAT LEVEL(Hlq.V1R1M0.INVENTRY) ALL
/*
//DUMMYREC  DD   *
Z999999999999999999999999999999999999999999999999999999999999999999999999999
//*
```

Figure 15. Create Inventory database JCL

2. Change the test string *VVVVVV* to the appropriate DASD volume serial number where the Inventory File resides.
3. Submit the JCL to create the Inventory database.
4. Review the output to verify its successful completion.

Defining the Backup Table

The Backup Table is a VSAM KSDS file that is allocated with the batch job CNTL(BKUPDEFN).

Procedure

1. Enter S next to BKUPDEFN to modify the JCL.
   The JCL is displayed.

```
//IZBR110A JOB (),CLASS=A,MSGCLASS=X,REGION=128M,NOTIFY=&SYSUID
//*
//IZBRPROC JCLLIB ORDER=(Hlq.V1R1M0.CNTL)
//*====================================================================
//* THIS JCL IS USED TO CREATE THE IZBR BACKUP TABLE
//*  CHANGE ALL 'VVVVVV' TO VOLSER OF PACK TO PLACE VSAM BKUPTBL
//*====================================================================
//* BUILD BKUPTBL PRIMING RECORD
//*====================================================================
//JS10    EXEC PGM=IEBDG
//SYSPRINT DD  SYSOUT=*
//BKUPP    DD  DSN=&&TEMP,DISP=(NEW,PASS),UNIT=SYSDA,
//             DCB=(RECFM=F,LRECL=70,BLKSIZE=70),SPACE=(TRK,(1))
//SYSIN    DD  *
 DSD OUTPUT=(BKUPP)
 FD NAME=A,LENGTH=44,STARTLOC=1,PICTURE=44,                          X
   'Z99999999999999999999999999999999999999999999'
 CREATE NAME=A
 END
/*
//*====================================================================
//*  DEFINE BKUPTBL & PRIME W/REPRO
//*====================================================================
//JS20    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//BKUPP    DD  DSN=&&TEMP,DISP=(OLD,PASS)
//BKUPTBL  DD  UNIT=SYSDA,DISP=SHR,VOL=SER=BRNV100
//SYSIN    DD  *
  DELETE Hlq.V1R1M0.BKUPTBL CLUSTER
  SET MAXCC=0
  DEFINE CLUSTER (  NAME(Hlq.V1R1M0.BKUPTBL)      -
                    OWNER(IZBR)                   -
                    INDEXED                       -
                    RECORDSIZE(284 1808)          -
                    SHAREOPTIONS(2 3)             -
                    KEYS(70 0)                    -
                    VOL(VVVVVV)                   -
                    NOREUSE UNIQUE)               -
         DATA    (  NAME(Hlq.V1R1M0.BKUPTBL.DATA) -
                    FREESPACE(10 10)              -
                    CYLINDERS(10 5) CISZ(8192) )  -
         INDEX   (  NAME(Hlq.V1R1M0.BKUPTBL.INDEX) CISZ(2560))
  IF LASTCC = 0 THEN -
           REPRO ODS('Hlq.V1R1M0.BKUPTBL') INFILE(BKUPP)
/*
//*====================================================================
//* DELETE PRIMED RECORD
//*====================================================================
//JS30    EXEC PGM=BRNVVSMD
//STEPLIB  DD  DSN=Hlq.V1R1M0.LIBRARY,DISP=SHR
//DDIN     DD  DSN=&&TEMP,DISP=SHR
//BKUPTBL  DD  DSN=Hlq.V1R1M0.BKUPTBL,DISP=(SHR),
//             AMP=('BUFNI=12,BUFND=49')
//VFIOUT   DD  SYSOUT=*
//
```

Figure 16. Create Backup Table JCL

2. Change *VVVVVV* to the appropriate DASD volume serial number where the Backup Table resides.
3. Submit the JCL to create the Backup Table.
4. Review the output to verify its successful completion.

Defining the Backup Audit data set

The Backup Audit data set is a VSAM ESDS file that is allocated with the batch job CNTL(BAUDDEFN).

Procedure

1. Enter S next to BAUDDEFN to modify the JCL.

   The JCL is displayed.

```
//*===================================================================
//* THIS JCL IS USED TO CREATE THE BACKUP AUDIT FILE  (BKUPAUD)
//*  CHANGE ALL 'VVVVVV' TO VOLSER OF PACK TO PLACE VSAM BKUPAUD
//*===================================================================
//*
//*===================================================================
//* BUILD BKUPAUD PRIMING RECORD
//*===================================================================
//JS10    EXEC PGM=IEBDG
//SYSPRINT DD  SYSOUT=*
//BAUDP    DD  DSN=&&TEMP,DISP=(NEW,PASS),UNIT=SYSDA,
//             DCB=(RECFM=F,LRECL=70,BLKSIZE=70),SPACE=(TRK,(1))
//SYSIN    DD  *
  DSD OUTPUT=(BAUDP)
  FD NAME=A,LENGTH=44,STARTLOC=1,PICTURE=44,                          X
   'Z99999999999999999999999999999999999999999999'
  CREATE NAME=A
  END
/*
//*===================================================================
//*  DEFINE BKUPAUD ESDS VSAM FILE
//*===================================================================
//JS20    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//BAUDP    DD  DSN=&&TEMP,DISP=(OLD,PASS)
//BKUPAUD  DD  UNIT=SYSDA,DISP=SHR,VOL=SER=BRNV100
//SYSIN    DD  *
  DELETE Hlq.V1R1M0.BKUPAUD CLUSTER
  SET MAXCC=0
  DEFINE CLUSTER (  NAME(Hlq.V1R1M0.BKUPAUD) -
                    OWNER(IZBR)               -
                    NONINDEXED                -
                    RECORDSIZE(284 1808)      -
                    SHAREOPTIONS(2 3)         -
                    MEGABYTES(10 10)          -
                    VOL(VVVVVV)               -
                    UNIQUE)
  IF LASTCC = 0 THEN -
     REPRO ODS('Hlq.V1R1M0.BKUPAUD') INFILE(BAUDP)
  LISTCAT LEVEL(Hlq.V1R1M0.BKUPAUD) ALL
/*
//
```

Figure 17. Create Backup Audit file JCL

2. Change *VVVVVV* to the appropriate DASD volume serial number where the Backup Audit File resides.
3. Submit the JCL to create the Backup Audit data set.
4. Review the job output to verify its successful completion.

Defining the Real-Time Command Queue

The Real-Time Command Queue is a VSAM RRDS file that is allocated with the batch job CMDQDEFN, which is located in the CNTL library.

Procedure

1. Enter S next to CMDQDEFN to modify the JCL.

   The JCL is displayed.

```
//*===================================================================*
//* THIS JCL IS USED TO CREATE  THE RT/SMF CMDQUEUE DATASET           *
//*  CHANGE ALL 'VVVVVV' TO VOLSER OF DASD TO PLACE CMDQUEUE CLUSTER *
//*===================================================================*
//*
//*====================================================================
//* JS10: BUILD CMDQUEUE PRIMING RECORD
//*====================================================================
//JS10     EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=*
//CMDQP    DD DSN=&&TEMP,DISP=(NEW,PASS),UNIT=SYSDA,
//    DCB=(RECFM=F,LRECL=120,BLKSIZE=120),SPACE=(TRK,(1))
//SYSIN    DD *
 DSD OUTPUT=(CMDQP)
 FD NAME=A,LENGTH=1,STARTLOC=1,FILL=X'00',PICTURE=1,                  X
   B'1'
 CREATE NAME=A
 END
/*
//*====================================================================
//* JS20: DEFINE CMDQUEUE DATASET & PRIME W/REPRO
//*====================================================================
//JS20     EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//CMDQP    DD DSN=&&TEMP,DISP=(OLD,PASS)
//SYSIN    DD *

 DELETE TECH.IZBR.V1R1M0.RTCMDQ
 SET MAXCC=0

 DEFINE CLUSTER( NAME(TECH.IZBR.V1R1M0.RTCMDQ)     -
                 SHAREOPTIONS(4 4)                 -
                 RECORDS(128 0)                    -
                 RECORDSIZE(120 120)               -
                 VOL(VVVVVV)                       -
                 NUMBERED   )

 IF LASTCC = 0 THEN -
         REPRO ODS('TECH.IZBR.V1R1M0.RTCMDQ') INFILE(CMDQP)
/*
```

Figure 18. Create CMDQUEUE data set JCL

2. Change *VVVVVV* to the appropriate DASD volume serial number where the Command Queue resides.
3. Submit the JCL to create the Command Queue.
4. Review the output to verify its successful completion.

Allocating the backup method JCL libraries

About this task

Use Option 6 of the IBM Z Batch Resiliency Installation Panel, BKUPMETH to:

- Allocate the files required for backing up and recovering critical files.
- Edit job card parameters for Backup and Restore JCL for your installation.

Procedure

1. From the IBM Z Batch Resiliency Installation Panel, select option 6, BKUPMETH.

   The Allocate Files for Backup Methods panel is displayed.

```
------------------ IZBR 1.1.0 Allocate Files for Backup Methods ---------------
OPTION  ===>


   .  1 BKUPMETH    - Allocate Files for Backup Methods

   .  2 BKUPJCL     - Edit Jobcard for BKUPJCL

   .  3 RESTJCL     - Edit Jobcard for RESTJCL
```

Figure 19. Allocate Files for Backup Methods panel

2. From the Allocate Files for Backup Methods panel, enter option 1, BKUPMETH, to allocate, rename, edit, compress or delete files for the backup methodologies defined to IBM Z Batch Resiliency.

```
---------------------- IZBR 1.1.0 Backup Methodology Panel -- Row 1 to 4 of 18
COMMAND ===>                                                  SCROLL ===> CSR

 Enter  (A)DD - to Add New Backup Method on the command line above.
 Select an existing backup methods with the following line commands:

  C - Comment Switch         S - Edit File(s)       X - Edit Driver(s)
  A - Allocate File(s)       Z - Compress File(s)
  R - Rename File(s)         D - Delete Method
Allocation  Method   File - Backup  or  Restore        Volser
------------------------------------------------------------------------------
NOALLOC     ABARS    BKUP  IZBRA.V110.BKUPJCL.ABARS
NOALLOC              REST  IZBRA.V110.RESTJCL.ABARS
------------------------------------------------------------------------------
ALLOC       ALLOCAT  BKUP  IZBRA.V110.BKUPJCL.ALLOCAT  SW0006
ALLOC                REST  IZBRA.V110.RESTJCL.ALLOCAT  SW0006
------------------------------------------------------------------------------
ALLOC       CATBKUP  BKUP  IZBRA.V110.BKUPJCL.CATBKUP  SW0006
ALLOC                REST  IZBRA.V110.RESTJCL.CATBKUP  SW0006
------------------------------------------------------------------------------
ALLOC       CAT2MIG  BKUP  IZBRA.V110.BKUPJCL.HRECALL  SW0006
ALLOC                REST  IZBRA.V110.RESTJCL.HRECALL  SW0006
------------------------------------------------------------------------------
```

Figure 20. Backup Methodology Panel

When IBM Z Batch Resiliency is installed to this point, two backup methodologies are defined to the system:

- DASD – for DSS, FDR, or DMS
- IBM Z Batch Resiliency BRNVCPY – process which is installed by default.

The status of the data sets for each backup methodology file is displayed to the right of the method and its data set names.

- If the files have a status of ALLOC, the volume serial numbers of the devices, on which they are allocated, is also displayed.
- If the allocation status is NOALLOC, then the files are not cataloged and must be allocated.

3. From the Backup Methodology Panel, type A next to a methodology to choose which file(s), Backup or Restore or both, should be allocated.

   The following pop-up window is displayed.

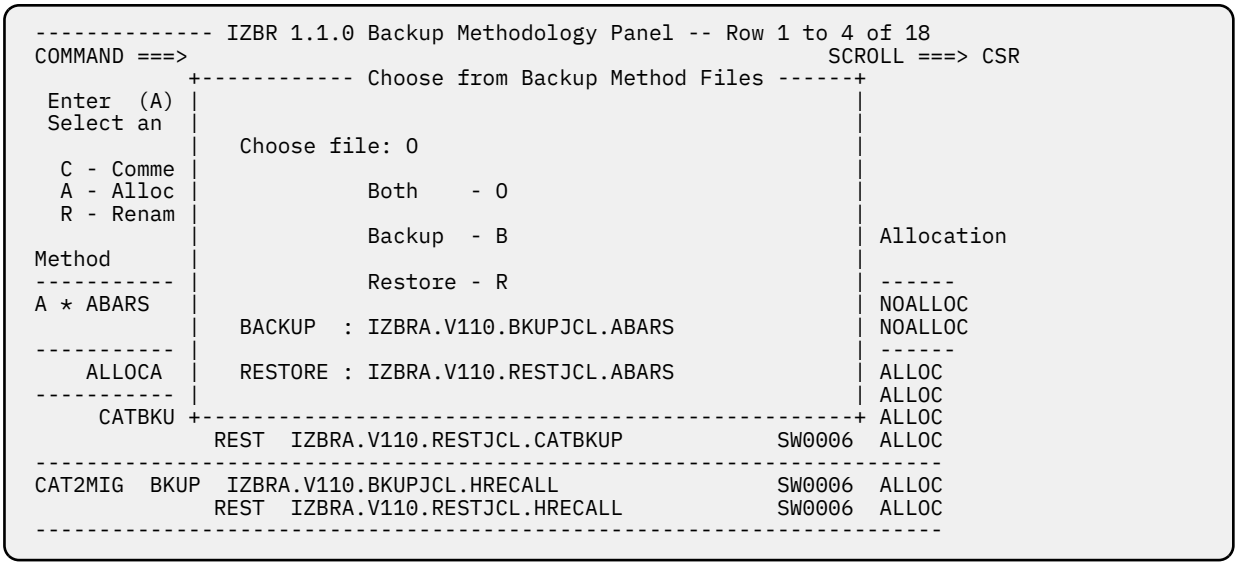


Figure 21. Choose from Backup Method Files panel

The same pop-up window is shown when you select the Rename, Edit or Compress option. During the process of allocating the files, a standard ISPF panel is displayed to allow the allocation to proceed.

4. If the files for the Backup Methodologies that you want to use are not allocated, allocate them with these panels or alternatively, as PDS files suitable for JCL, typically as RECFM=FB and LRECL=80.

   You can also use these panels to periodically compress the Backup JCL files.

Setting up the system PROCLIB

The next task is to create an IBM Z Batch Resiliency cataloged procedure in a suitable PROCLIB library.

About this task

Note: This step is not needed if using a JCLLIB statement that points to the IZBR CNTL data set where the IZBR PROC resides.

Procedure

1. Ask your system programmer which system cataloged procedure library to use.
2. Copy member PROCVFI from the CNTL library into a system cataloged procedure library.

Setting up the execution EXEC

One of the members created in the Create JCL process is BRN110, a REXX EXEC.

Procedure

1. Copy HLQ.CNTL(BRN110) to a SYSEXEC library (or SYSPROC library if no SYSEXEC library exists in your TSO logon procedure).
2. To begin an IBM Z Batch Resiliency online session from any ISPF screen, type the following command:

   ```
   TSO BRN110
   ```

3. If you do not start IBM Z Batch Resiliency from the TSO ready prompt, you can begin an IBM Z Batch Resiliency session by invoking the TSO Command Processor (option 6 on the ISPF Primary Option Menu) and then entering the following command:

   ```
   EX 'HLQ.CNTL(BRN110)'
   ```

Running IBM Z Batch Resiliency across multiple LPARS

Procedure

If you plan to run IBM Z Batch Resiliency in different LPARS on the same machine and share IBM Z Batch Resiliency data sets between them, define the QNAMES with the parameter **SYSTEMS** in the SCOPE column (shown in Defined Enqueues) to GRS or MIM.

Checking for the REXX Runtime Libraries

About this task

The REXX runtime libraries or the REXX Alternate runtime library are required.

This job tests for the availability of the REXX runtime library and, if not available, enables the REXX alternate runtime library. If the REXX runtime library has already been located, the Installation panel may indicate that this task is not required:

```
R REXX build - Not applicable
```

Procedure

1. Select Option R from the IBM Z Batch Resiliency Installation Panel and submit the JCL.
2. Review the job's output to confirm that the job executed successfully.

IBM Z Batch Resiliency implementation

Administrator or disaster recovery coordinator must implement IBM Z Batch Resiliency after it is installed.

APF authorizing load libraries

Procedure

APF authorize the following load libraries:

- DSN=*hlq*.SCKKLOAD
- DSN=*hlq*.SBRNLOAD

If you plan to run TimeLiner, APF authorize these libraries:

- DSN=*hlq*.SBRNTLOD

TimeLiner component setup

You must set up the TimeLiner component.

TimeLiner hardware and software requirements

- Many of the TimeLiner installation tasks require assistance from a systems programmer. Coordinate with a systems programmer to install TimeLiner.
- The TimeLiner linear data set requires 3100 contiguous cylinders of space. Ensure that there is a volume with sufficient space.
- The TimeLiner log stream and staging data sets require 100 contiguous cylinders and 500 contiguous cylinders, respectively.
  - If you plan to manage the data sets with SMS, ensure that you have SMS classes that direct the data sets to volumes with sufficient space.

  - If you do not want to manage the data sets with SMS, ensure you have volumes with sufficient space.
- The user who defines the log stream needs authority to run the IXCMIAPU utility. Use of this utility is controlled via the FACILITY class resource MVSADMIN.LOGR.
  - Ask your security administrator to issue this authority to the user ID.

    For example, if you use RACF®, the security administrator can issue the following commands:

    ```
    permit mvsadmin.logr class(facility) id(xxxx) access(read)
    setr refresh raclist(facility)
    ```

Determining changes to SMFPRMxx member

To help you determine if any changes need to be made to the SMFPRMxx parmlib member, use TSO EXEC BRNLMFCK.

Before you begin

Verify that the following requirements are met:

- This exec must be run in a TSO environment
- The user invoking the exec must be authorized to use the CONSOLE command
- The user invoking the exec must be authorized to issue the DISPLAY SMF,O command
- The user must not have an SDSF session active in the same TSO session. If an SDSF session is active, the REXX EXEC fails with a message indicating that the console could not be activated.

About this task

Note: Because TimeLiner dynamically adds its exit modules to the exit points, you do not need to make any changes to your PROGxx member to associate the TimeLiner exit modules to the SMF exit points.

Procedure

1. Issue the following command:

   ```
   TSO EXEC 'HLQ.SBRNEXEC(BRNLMFCK)'
   ```

   When the command is executed, the following results occur:

   - It produces output listing any exit points defined in the active SMF configuration for SYS, as well as any exits defined for the STC, TSO, JES2 and JES3 subsystems.
   - If exits are missing from any of the exit lists, the REXX output informs you of any changes that need to be made.
2. Examine the output to determine if you need to change your settings.

Defining TimeLiner log stream

About this task

The submitter of this job needs authority to run IXCMIAPU. Use of this utility is controlled by the FACILITY class resource MVSADMIN.LOGR. The JCL is provided in *hlq*.CNTL(TLNDEFLG).

Procedure

1. Edit the JCL as indicated in the comments at the start of the job.
2. Submit the job.

Adding the TimeLiner module BRNLCT00 to the PPT (Program Properties Table)

Procedure

Add the following lines to the appropriate SCHEDxx member in PARMLIB:

```
PPT PGMNAME(BRNLCT00) /* BRNLCT00 ADDRESS SPACE          */
KEY(4)                /* USE STORAGE PROTECT KEY FOUR(4) */
```

Updating security rules

You must update security rules to allow the users or started tasks with the authority to create, update, and read the DASD-only log stream used by the TimeLiner started task.

Procedure

1. Ensure the TimeLiner started task has authority to update and read the Data-In-Virtual (DIV) data set.
   You'll be using this authority in Creating the TimeLiner DIV linear data set.
2. Ensure all TimeLiner STCs are able to read the following resources:
   - APPTABLE file specified in the TimeLiner started task JCL.
   - PARMLIB data sets specified in the TimeLiner started task JCL.
3. For all instances of TimeLiner that are participating in a TimeLiner plex group, where the data is collected in one DIV, ensure that the:
   - DIV NAME is the same in the TLNPRM00 PARMLIB member as discussed in Creating the TimeLiner DIV linear data set.
   - For multiple instances of TimeLiner, ensure each instance of TimeLiner is operating independently. Specify a unique DIV NAME in the TLNPRM00 PARMLIB member for each instance.

Setting up security for TimeLiner commands

The TimeLiner started task accepts the MODIFY and STOP commands.

Procedure

Secure the MODIFY and STOP commands so that only a subset of users can issue them to the TimeLiner address space.

Commands to use with STOP and MODIFY when working with TimeLiner

When you are working with TimeLiner, there are commands that can be used to control the MODIFY and STOP commands, as well as START and CANCEL.

Although it is best practice to run TimeLiner as a started task, you can also run it as a batch job. Commands that are different if TimeLiner is run as a job rather than a started task are noted in the following table.

These rules are for user IDs that need to issue commands to the TimeLiner address space. These rules are not for the userid associated with the TimeLiner address space itself.

Table 2. Job run TimeLiner commands

Command/ Keyword	Authority	Resource name	Additional information
Started Task Command			

Table 2. Job run TimeLiner commands (continued)

Command/ Keyword	Authority	Resource name	Additional information
CANCEL *jobname.id* CANCEL *id*	UPDATE	MVS™.CANCEL.STC.*mbrname.id*	Command is for a started task for which an identifier is provided.
CANCEL *jobname.id* CANCEL *id*	UPDATE	MVS.CANCEL.STC.*mbrname.id*	For a started task for which an identifier was not provided. *mbrname* is the name of the member containing the JCL source.
FORCE *jobname.id* FORCE *id*	CONTROL	MVS.FORCE.STC.mbrname.id	For a started task for which an identifier was provided.
FORCE *jobname*	CONTROL	MVS.FORCE.STC.*mbrname.jobname*	For a started task for which an identifier was not provided. *mbrname* is the name of the member containing the JCL source.
MODIFY *jobname* MODIFY *jobname.id* MODIFY *id*	UPDATE	MVS.MODIFY.STC.*mbrname.id*	This command is for a started task for which an identifier was provided.
MODIFY *jobname*	UPDATE	MVS.MODIFY.STC.*mbrname.jobname*	For a started task for which an identifier was not provided. *mbrname* is the name of the member containing the JCL source.
START *mbrname*[.*identifier*]	UPDATE	MVS.START.STC.*mbrname*[.*id*]	Command for started tasks for which an identifier was provided. *mbrname* is the name of the member containing the JCL source.
START *mbrname*, JOBNAME=*jobname*	UPDATE	MVS.START.STC.*mbrname.jobname*	For a started task for which an identifier was not provided. *mbrname* is the name of the member containing the JCL source.
STOP *jobname* STOP *jobname.id* STOP *id*	UPDATE	MVS.STOP.STC.*mbrname.id*	Command is for a started task for which an identifier was provided. *mbrname* is the name of the member containing the JCL source.
STOP *jobname*	UPDATE	MVS.STOP.STC.*mbrname.jobname*	The previous command is for a started task for which an identifier was not provided. *mbrname* is the name of the member containing the JCL source.

Table 2. Job run TimeLiner commands (continued)			
Command/ Keyword	**Authority**	**Resource name**	**Additional information**
Batch Job Command			
CANCEL *jobname*	UPDATE	MVS.CANCEL.JOB.*jobname*	For a job that is not a started task.
FORCE *jobname*	CONTROL	MVS.FORCE.JOB.*jobname*	For a job that is not a started task.
MODIFY *jobname*	UPDATE	MVS.MODIFY.JOB.*jobname*	Command is for a job that is not a started task.
STOP *jobname*	UPDATE	MVS.STOP.JOB.*jobname*	Command for a job that is not a started task.

Setting up security for the CSVDYNEX dynamic exits

TimeLiner invokes CSVDYNEX requests to dynamically add its exit modules to SMF exits.

About this task

For information about the commands that control which users have authority to issue CSVDYNEX requests, refer to the IBM documentation: Providing Security for Dynamic Exits .

To control access to programs that create and call dynamic exits:

Procedure

1. Authorize specific users or groups of users to issue the various CSVDYNEX requests.
2. Ensure the *USERID* associated with the CSVDYNEX rules is that of the TimeLiner address space. TimeLiner uses the LIST and ADD request options of CSVDYNEX.

Setting up security for a DASD-only log stream

Follow these steps to define a log stream resource profile to RACF for a DASD-only log stream and give users access to the profile.

About this task

In the examples below, the log stream name is 'SYSPLEX.TIMELINR.JOURNAL', which is the default log stream name for TimeLiner. The user can override this name in the TimeLiner parameters JOURNALLOGSTREAM setting.

Apply the rules shown here to the USERID of the TimeLiner address space.

Procedure

1. Define a resource profile for the log stream to RACF in the LOGSTRM class.

   For example,this RACF command defines a resource profile named SYSPLEX.TIMELINER.JOURNAL:

   ```
   RDEFINE LOGSTRM (SYSPLEX.TIMELINR.JOURNAL) UACC(NONE)
   ```

   The name of the resource profile (SYSPLEX.TIMELINR.JOURNAL) must match the log stream name specified in the NAME parameter of the DEFINE LOGSTREAM statement in the LOGR policy.
2. Give system logger administrators, who run the IXCMIAPU utility, ALTER access to the log stream resource profile. IXCMIAPU is used to define, update, and delete log streams.

For example, this RACF command gives a user ALTER access to SYSPLEX.TIMELINR.JOURNAL

```
PERMIT SYSPLEX.TIMELINR.JOURNAL CLASS(LOGSTRM) ACCESS(ALTER) ID(userID)
```

3. Give operators, who start TimeLiner, UPDATE access to the log stream resource profile. This allows TimeLiner to read and write messages.

   For example,this RACF command gives the AOPOPER group UPDATE access to profile AOP.MSG:

   ```
   PERMIT SYSPLEX.TIMELINR.JOURNAL CLASS(LOGSTRM) ACCESS(UPDATE) ID(userID or RACF group
   profile)
   ```

4. Optional: If you copy profiles in the LOGSTRM class into virtual storage, refresh the class after you define new profiles or permit new users to profiles in the class.

   For example, this RACF command refreshes the LOGSTRM class:

   ```
   SETROPTS RACLIST(LOGSTRM) REFRESH
   ```

Updating security rules when using CA Top Secret

About this task

To work with CA Top Secret and TimeLiner, set up the TimeLiner started task as follows:

Procedure

1. Define the TimeLiner address space and the required SAF authorizations by doing one of the following steps:
   a) Define TLINER in the Started Task table to bypass SAF calls: TSS ADD(STC) PROCNAME(TLINER) ACID(BYPASS). This bypasses all SAF calls and, therefore, does not require additional resource permits.
   b) Define TimeLiner in the Started Task table: TSS ADD(STC) PROCNAME(TLINER) ACID(*userid*).
2. Permit access to the DASD log stream and staging data sets.

   ```
   TSS ADD(dept) DSNAME(IXGLOGR.SYSPLEX.TIMELINR.JOURNAL)
   TSS PER(userid) DSNAME(IXGLOGR.SYSPLEX.TIMELINR.JOURNAL) ACCESS(ALL)
   ```

   The above commands assume that the default high-level qualifier is IXGLOGR and that you are using the default log name of SYSPLEX.TIMELINER.JOURNAL.
3. Permit access to the systems SYS1.PARMLIB:

   ```
   TSS ADD(dept) DSNAME(SYS1.PARMLIB)
   TSS PER(userid) DSNAME(SYS1.PARMLIB) ACCESS(READ)
   ```

4. Control which applications can access the system logger resources.

   ```
   TSS ADD(dept) LOGSTRM(SYSPLEX.TIMELINR.JOURNAL)
   TSS PER(acid) LOGSTRM(SYSPLEX.TIMELINR.JOURNAL) ACCESS(UPDATE)
   ```

5. To authorize who can use the IXCMIAPU utility:

   ```
   TSS ADD(dept) IBMFAC(MVSADMIN.)
   ```

   To define the structure:

   ```
   TSS PER(acid) IBMFAC(MVSADMIN.) ACCESS(ALL)
   ```

   To execute a report:

   ```
   TSS PER(acid) IBMFAC(MVSADMIN.) ACCESS(READ)
   ```

The LOGSTRM resource class is already predefined in CA Top Secret.

6. TimeLiner accepts MODIFY commands. Consider adding a rule to ensure that only a subset of users can issue MODIFY commands to the TimeLiner address space.

Creating the TimeLiner DIV linear data set

Before you begin

Be sure that you have at least 3100 cylinders of space available to meet the requirements for the Data-In-Virtual (DIV) data space.

Procedure

1. Modify the JCL in `HLQ.CNTL(TLNDEFDV)`.
2. Change *VVVVVV* to the appropriate DASD volume serial number.

Creating the VSAM UFI file

Procedure

1. Modify the JCL in HLQ.CNTL(UFIDEF).
2. Change *VVVVVV* to the appropriate DASD volume serial number.

Modifying TimeLiner parameters

Procedure

Modify the TimeLiner parameters in *hlq*`.USERMAC(TLNPRM00)` for your installation, as described in the following table:

Table 3. TimeLiner parameters

Parameter	Example	Description
JOURNALLOGSTREAM (STREAMNAME)	JOURNALLOGSTREAM (SYSPLEX.TIMELINR.JOURNAL)	Name of the TimeLiner log stream that contains the JOB Information after the monitored job has completed. It must match the name used when defining the log stream.
MSGLEVEL	MSGLEVEL (INFO)	The minimum level for messages produced by dynamic allocation. Valid values are NONE, INFO, WARN, or SEVERE. The optimal setting is INFO.
OBJECTDSN / OBJDSNAME	OBJECTDSN (*HLQ.TLN.DIV*)	DIV object data set name.
TRANSACTIONDUMPINDEX	TRANSACTIONDUMPINDEX (SYS1.DDIR)	Dump index.
TRANSACTIONDUMPDSNS	TRANSACTIONDUMPSDSNS (TLN.ATLN110.DMP.D&&LYYMMDD..T && LHHMMSS..S&DS)	Transaction dump data set name.

Table 3. TimeLiner parameters (continued)		
Parameter	**Example**	**Description**
TRANSACTIONDUMPS	TRANSACTIONDUMPS (YES)	• YES - Specifies to take transaction dumps. • NO – (Default) Do not take transactions dumps.
UFIFILE	UFIFILE (*HLQ.UFIFILE*)	UFI file data set name.

Copying and updating the JCL for the TLINER address space

Procedure

1. Copy BRNLPR00 to PARMLIB data set of (TLINER) STC or Job (usually USERMAC). Tailor to suit after copying. Change PARM='MEMBER(BRNLPR00)' in the TLINER STC or job if you rename this member.

   This is a batch job that you can update to run as a started task in your environment.
2. Ensure that the APPTABLE DD specifies a member of a PDS and that member is the application table to be used.
3. Ensure that the following members exist in the TLINER PARMLIB:
   - RTSRTD00
   - SMFPARMS
   - VFIPARMS
   - VFISTASK
   - VFITABLS

Chapter 3. Configuration

After you complete IBM Z Batch Resiliency software installation and initial implementation, you must configure IBM Z Batch Resiliency for your environment and set up IBM Z Batch Resiliency database maintenance.

Building the APPTABLE

Before IBM Z Batch Resiliency can identify critical data sets, the names of the jobs that make up your critical applications must be known. IBM Z Batch Resiliency uses the APPTABLE, a table of job names or job masks and application identifiers, to associate a job to a particular application. Required information includes the application ID, the first jobs for the application, and the names of all jobs belonging to the application.

Procedure

Build the APPTABLE.

You can build the APPTABLE manually or use one of the job scheduler interfaces provided with IBM Z Batch Resiliency.

- If your environment has implemented consistent application related naming standards for job names, you can build the APPTABLE manually. Refer to the sample APPTABLES in USERMAC and follow the format. See “Building the APPTABLE using ISPF panels” on page 29 or “Building the APPTABLE by PDSE direct edit” on page 32.
- To build an APPTABLE for a larger environment, you can automate the APPTABLE build process by using one of the IBM Z Batch Resiliency job scheduler interface programs that corresponds with the job scheduler used at your installation.

  IBM Z Batch Resiliency provides interface programs for CA Workload Automation CA 7® Edition, CA Schedule, ASG-ZEKE, CA ESP, Control-M, JOBTRAC, and IZWS job schedulers. User exits, associated with the job scheduler interface programs, can be used to eliminate non-production jobs from the APPTABLE. For details about using the job scheduler interface utilities, see “Building the APPTABLE using job scheduler utilities” on page 34.

Building the APPTABLE using ISPF panels

To create the application table from the ISPF interface, follow these instructions.

Procedure

1. Invoke the IBM Z Batch Resiliency panels by running:

   ```
   TSO EX 'HLQ.CNTL(BRN110)'
   ```

2. From the IBM Z Batch Resiliency Primary Panel select option 5, MAINTENANCE, to display the System Maintenance panel.
3. From the System Maintenance panel, select option 2, APPTABLE - Application Table Maintenance.

   The Application Panel displays all current application entries in the application table. No entries are defined the first time you invoke this panel.

© Copyright IBM Corp. 2019, 2020

```
------------------------- IZBR 1.1.0 Application Panel ----------------------
COMMAND ===>                                                 SCROLL ===> PAGE

 Enter one of the following commands on the command line below to:
 (A)DD Add New Application (F)IND Find Application (M)OD Modify Description

 Command: ______ Applid: ________ Description: _________________

 Select an existing application below with the following line commands:
 B Browse Jobs' S Edit Jobs D Delete Application

         Applid   Description      Message
------------------------------------------------------------------------------
```

Figure 22. IBM Z Batch Resiliency Application Panel

4. To add a new application to the application table, enter an application identifier (APPLID) or application description for the application you want to add.
   - The APPLID is a maximum of eight characters with no embedded blanks. Additionally, the APPLID can contain letter, numbers or valid standard characters ($ @ #). The first character of an APPLID cannot be a number. It must also represent a valid PDS member name in MVS, since this APPLID is used as a member name in the backup and restores JCL PDS files for this application.

     Do not use the GLOBAL reserved word as the APPLID.
   - The application description formats the APPLID for reporting purposes. The description can be up to 16 alphanumeric characters. (Special characters such as ' * = are not allowed.)

   The Job Edit Panel is then automatically displayed so you can add job name or job mask entries for this application.

```
------------------------- IZBR 1.1.0 Application Panel ----------------------
COMMAND ===>                                                 SCROLL ===> PAGE
      +--------------------------- Job Edit Panel ----------------------------+
 Ente |                                                        Row 1 from 1  |
 (A)D | COMMAND ===>                                        SCROLL ===> PAGE  |
      |                                                                       |
 Comm |  Applid: GL        Application Description: GENERAL LEDGER            |
      |                                                                       |
 Sele |  Enter one of the following commands on the command line below to:    |
 B Br |    (A)DD Add a New Job or Mask  (F)IND  Find a Job or Mask.           |
      |                                                                       |
      |  Command: ADD    Job/Mask: GL*      First Job? (Y or N): N            |
----- |                                                                       |
***** |  Select a job name/mask below with the following line commands:       |
      |     D Delete Entry,  S Edit Entry                                     |
      |                                                                       |
      |          Job        First Job                                         |
      |          Name/Mask  Indicator        Message                          |
      | ------------------------------------------------------------------    |
      |       .  GL*                        *ADDED                            |
      | ********************** BOTTOM OF DATA ***************************** |
      +-----------------------------------------------------------------------+
```

Figure 23. IBM Z Batch Resiliency Job Edit Panel

After you have defined applications to IBM Z Batch Resiliency, you can use the FIND (F) and MOD (M) commands.

5. To find the next occurrence equal to or greater than the APPLID specified, type F in the COMMAND field and type the application ID in the APPLID field.

   This locates the application that has an APPLID equal to GL. If GL does not exist, the entry that most closely matches and is greater than GL is located.
6. To modify the application description for the chosen APPLID, type M in the COMMAND field, the APPLID, and a description. This changes the current description for the application.
7. To browse, edit, or delete specific application entries, type the line command B, S, or D, respectively, next to the specific application and press the Enter key.

Specifying job masks and wildcards in the Job Edit panel

The first time an application is added, or when you use the **Edit line** command to edit an existing application, the Job Edit pop-up panel is automatically invoked so that entries for the application can be added or modified.

Using job masks

In the JOB/MASK field, type the name of the jobs or job masks for the application you have selected. The job name can be up to eight characters with no embedded blanks. A job mask is a representation of an exclusive group of jobs that should be included as part of the application. Job masks consist of alphanumeric characters and special characters percent sign (%), question mark (?), and asterisk (*).

Using job mask wildcards

The question mark (?) and percent sign (%) characters are used as single position wildcard characters. For example, to include jobs that could have any character in position 1 of the job name but must end with "GLJOB", you could enter '?GLJOB' or '%GLJOB' in the JOB/MASK field. This would include jobs AGLJOB, BGLJOB, CGLJOB, and so on.

The asterisk (*) character is used to ignore any characters following the position where this character is placed. For example, to include jobs that could have any characters following position 5, but that start with PGLJB, you could enter PGLJB* in the JOB/MASK field. This would include jobs PGLJB1, PGLJB12, PGLJBXYZ, and so on. Note that an entry such as GL*JOB is invalid, as it would include any job that starts with GL (the "JOB" portion of this mask would be ignored). Also, note that the entry *GLJOB is invalid, as it would include all jobs starting with any character (the GLJOB position would be ignored).

You can also mix the three types of wildcards. For example, to include jobs that may start with any character in position 1, require GLJOB in positions 2-6, and could have any characters in positions 7 through 8, you could enter '%GLJOB*' in the JOB/MASK field. This would include jobs PGLJOB1, XGLJOB2, YGLJOBAB, and so on.

Specifying the first job

The FIRST JOB? field indicates to IBM Z Batch Resiliency whether the specified job name/job mask should be used in determining the start of the daily cycle for this application. At least one entry for this application must represent a "first job" so that the system can determine how files are referenced in normal daily processing within this application.

Type Y (Yes) or N (No) in the FIRST JOB? field to indicate if this is the first job (or one of multiple first jobs) for this application.

At times, there may be some confusion as to what constitutes a first job. If, for example, you run a backup as the first job of your application cycle, IBM Z Batch Resiliency considers all input files for the backup job to be critical (since they must exist before this job can run properly). Consequently, selecting the backup job as the first job would not be a wise choice because the backup would never be the first job to run at the recovery site in the event of a disaster.

The job considered as the first job should be the first jobs that run in an actual disaster for a given application after all of the critical files for this application have been restored. This would not include jobs used to recover the application at the recovery site, nor should it include backup jobs.

Specifying multiple first jobs

More than one job can mark the start of a daily application cycle. In this case, there are multiple entries with a first job indicator.

For example, if PGLXYZA1 starts the general ledger application on Mondays, Wednesdays, and Fridays, but on Tuesdays and Thursdays PGLXYZA2 is the first job, you would mark both PGLXYZA1 and PGLXYZA2 as a first job. IBM Z Batch Resiliency then automatically determines from the SMF data that a

new cycle is being run for this application whenever it encounters either one of these jobs on a different date.

Adding a dummy first job

In ad hoc applications, or in applications where the first job changes on a regular basis, it may be difficult to determine which jobs to mark as a first job. If this is the case, it is advisable to place a dummy job (for example an IEFBR14) in your job scheduling system as the first job run for an application. Use of a dummy first job permanently marks the start of an application cycle and prevents you from having to change the APPTABLE every time the first job changes.

If you use wildcards to represent multiple first jobs, make certain that all possible combinations represent valid first jobs. Otherwise, invalid critical file analysis may result. For example, if you used GL* as a first job entry, all jobs that start with GL must be valid jobs to start this application.

After you enter all of the first jobs and masks to define the application to IBM Z Batch Resiliency, press the PF3 key to save all entries.

Pressing the PF3 key returns you to the Application Panel so you can add more applications.

```
------------------------- IZBR 1.1.0 Application Panel ------ Row 1 to 1 of 1
COMMAND ===>                                                 SCROLL ===> PAGE

 Enter one of the following commands on the command line below to:
 (A)DD Add New Application (F)IND Find Application (M)OD Modify Description

 Command: ______ Applid: ________ Description: _________________

 Select an existing application below with the following line commands:
 B Browse Jobs S Edit Jobs D Delete Application

         Applid   Description      Message
-------------------------------------------------------------------------------
      .  GL       GENERAL LEDGER   *ADDED
```

When you finish adding applications, press the PF3 key to return to the System Maintenance Panel.

Building the APPTABLE by PDSE direct edit

The APPTABLE contains the job names of all critical applications. Building the APPTABLE manually is best practice if you have implemented consistent naming standards for job names.

If you build the APPTABLE manually, follow the format in USERMAC(APPTABLE). Each line consists of four fields that define the job name, or job mask, and application identifier that associates each job with an application, as shown in the following figure.

```
----+----1----+----2----+----3----+---
NEWDAY  *TESTAPP
#DAILY   TESTAPP  TEST APPLICATION
#*       TESTAPP  TEST APPLICATION
ENDDAY  #TESTAPP
```

APPTABLE format and field descriptions

This section describes the APPTABLE field descriptions and the use of job mask wildcards. The APPTABLE format is described below:

Table 4. APPTABLE format and field descriptions			
Column position	**Description**	**Maximum characters**	**Example**
1-8	Job name or job mask	8	NEWDAY
9	Job position indicator	1	(*) First job
10-17	Application ID	8	TESTAPP
19	Optional description	11	TEST APPLICATION

The APPTABLE is column sensitive. The job name or job mask is in columns 1 through 8, the first job indicator is in column 9 and the application name occupies columns 10 through 17.

Specifying a job name

The job name can be up to eight characters with no embedded blank spaces. A job mask is a representation of an exclusive group of jobs that should be included as part of the application. Job masks consist of alphanumeric characters and special characters percent (%), question mark (?) and asterisk (*).

Using a job position indicator

The following characters can be used as a job position indicator:

*** (asterisk)**
Job is the first job of an application cycle. At least one first job must be specified for each application.

@
Job indicates the start of an application cycle but is not part of the application.

#
Job is the last job of an application cycle. If last job is not specified, the next occurrence of the first job also indicates the end of the previous application cycle.

>
Job that indicates the start of the day's processing for all defined applications.

Blank
Job is part of the application but does not indicate any special position within the cycle.

Application ID: The APPLID is a maximum of eight characters with no embedded blank spaces. The APPLID can contain letters, numbers or the valid standard characters dollar ($), ampersand (@), or number (#).

Application Description: The optional description is used to format the APPLID for reporting purposes. The description can be up to 16 alphanumeric characters. (Special characters such as single quote ('), asterisk (*), equals (=), and so on, are not allowed).

Using job mask wildcards

The question mark (?) and percent sign (%) are used as single position wildcard characters. For example, to include jobs that could have any character in position 1 of the job name, but must end with GLJOB, you could enter ?GLJOB or %GLJOB in the JOB/MASK field. This would include jobs named AGLJOB, BGLJOB, CGLJOB.

The asterisk (*) is used to ignore any characters following the position where this character is placed. For example, to include jobs that could have any characters following position 5, but that start with "PGLJB", you could enter PGLJB* in the JOB/MASK field. This would include jobs named PGLJB1, PGLJB12, PGLJBXYZ.

The asterisk (*) mask is valid only at the end of a job mask.

Specifying multiple first jobs

More than one job can mark the start of a daily application cycle. In this case, there are multiple entries with a first job indicator.

For example, if PGLXYZA1 starts the general ledger application on Mondays, Wednesdays, and Fridays, but on Tuesdays and Thursdays PGLXYZA2 is the first job, you would mark both PGLXYZA1 and PGLXYZA2 as a first job. RTD then automatically determines from the SMF data that a new cycle is being run for this application whenever it encounters either one of these jobs on a different date.

Specifying jobs in multiple applications

Jobs, including first jobs, can be specified more than once with different Application IDs. In this case, these jobs are used in the determination of critical files for each of the APPLIDs specified.

Overriding job masks

If you find job masks work in the majority of cases but there are some job name exceptions, you can include fully-qualified job name references using the APPLID and identifier of another application.

For example, suppose you used AP* as a job mask in the Application Table to place all jobs that start with the letters AP in the Accounts Payable application. Later, you find that there were two jobs, APJOB039 and APJOB040, that should be considered as part of the general ledger application for reporting purposes.

You would then code these entries as follows:

Table 5. Overriding job masks

JOBNAME/MASK	**APPLID**
AP*	AP
APJOB039	GL
APJOB040	GL

In this case, the fully-qualified job names override the AP* wildcard.

If you want a job name or job mask that is a subset of a more general job mask to have more than one APPLID (as outlined in this section), code the more specific job name or mask for each APPLID. In the example above, APJOB039 and APJOB040 appear only in APPLID GL. If you want APJOB039 to appear in both APPLIDs, code that section of the APPTABLE as follows:

Table 6. Multiple APPLIDs

JOBNAME/MASK	**APPLID**
AP*	AP
APJOB039	GL
APJOB039	AP
APJOB040	GL

In this case, APJOB039 is used in critical file analysis for both AP and GL, APJOB040 is used for the APPLID GL only and all other jobs beginning with AP is used for the AP APPLID only.

Building the APPTABLE using job scheduler utilities

You can use the job scheduler interface programs to build the APPTABLE and keep it updated as new jobs and applications are added. This set of utilities helps ensure that the APPTABLE is defined according to information supplied by the job scheduler reports.

The following job scheduler interface programs are provided with IBM Z Batch Resiliency:

Table 7. Job scheduler interface programs provided with IBM Z Batch Resiliency	
Member name	**Utility function**
CA7JOB - CA 7 APPTABLE creation utility	Creates the APPTABLE from CA7 reports
CTMJOBB - Control-M APPTABLE creation utility	Creates the APPTABLE from CONTROL-M reports
ESPRHIST	Creates the APPTABLE from ESP history
JOBTRAC - JOBTRAC APPTABLE creation utility	Creates the APPTABLE from the JOBTRAC schedule PDS
OPCJOB	Creates the APPTABLE from TWS/OPS reports
ZEKEJOB	Creates the APPTABLE from ZEKE reports

The identification component utilities and reports panel is

```
----------- IZBR 1.1.0 Identification Component Utilities and Reports ---------
OPTION  ===>

  Enter one of the following selections next to member name:
     S - to Edit JCL                     H - to View Documentation
     P - to Edit Execution Parameters    X - to Edit User Exit Code
     R - to View Report                  Z - to Submit JCL

                                                                 More:     +
   .  AUDITCHK - Critical Job Auditing Report
   .  CONCATRP - Concatenated File Report
   .  CRITREPT - Critical File List Report
   .  CROSSDEP - Cross Application Dependent File Report
   .  DORMANT  - Report of Dormant Critical File entries
   .  FREQRPT  - Report on number of SMF records found in History Database
   .  HISTRPT  - Adhoc Report of History Database
   .  INVTYRPT - Adhoc Report of Inventory Database
   .  JOBRPT   - Report on Jobs Opening Selected Dataset(s)

  ISV Scheduling Utilities:
   .  CA7JOB   - Utility to create Application Table from CA7 reports
   .  CTMJOBB  - Utility to create Application Table from Control-M
   .  ESPJOB   - Utility to create Application Table from ESP PROC members
   .  OPCJOB   - Utility to create Application Table from OPC/TWS reports
   .  ZEKEJOB  - Utility to create Application Table from ZEKE reports
```

Figure 24. ISV Scheduling Utilities

Check the naming conventions for the job scheduler interface you use. For more information on the APPTABLE conventions for job name format, job position indicator, masking, and wildcards, see *APPTABLE format and field descriptions*.

Exits are included for tailoring the execution of the job scheduler utilities. If you need help coding an exit for any of the job scheduler interfaces, contact Technical Support.

CA7JOB - CA 7 APPTABLE creation utility

Use the IBM Z Batch Resiliency CA7JOB scheduler utility to create APPTABLE entries.

The CA7JOB member creates a separate forecast report for the current day, plus a user-specified additional number of days. This forecast report is then used to determine which jobs comprise an application. The SYSID in CA7 is used to define each application.

For example, if the current date is 06/20/18 and the number of days specified in the FALLCMDS step of CA7JOB is 3 (as best practice, set this to the VFIPARMS RETPD1 setting plus 1), the following FALL commands are generated:

```
FALL,FROM=061818,SPAN=24,TYPE=ALL
FALL,FROM=061918,SPAN=24,TYPE=ALL
FALL,FROM=062018,SPAN=24,TYPE=ALL
```

These FALL commands are input to CA7 to create the forecast report. There is a separate FALL command for each day. This is necessary for marking the start of a particular daily cycle so that IBM Z Batch Resiliency can find the first job for each application.

After IBM Z Batch Resiliency generates the FALL commands, CA7 creates the forecast report. This report is contained in a file referenced in the JCL by the DD name CA7RPT. The forecast report then re-creates the new APPTABLE from the information contained in the old APPTABLE and any newly found jobs from the current forecast report.

CA7JOB prerequisites

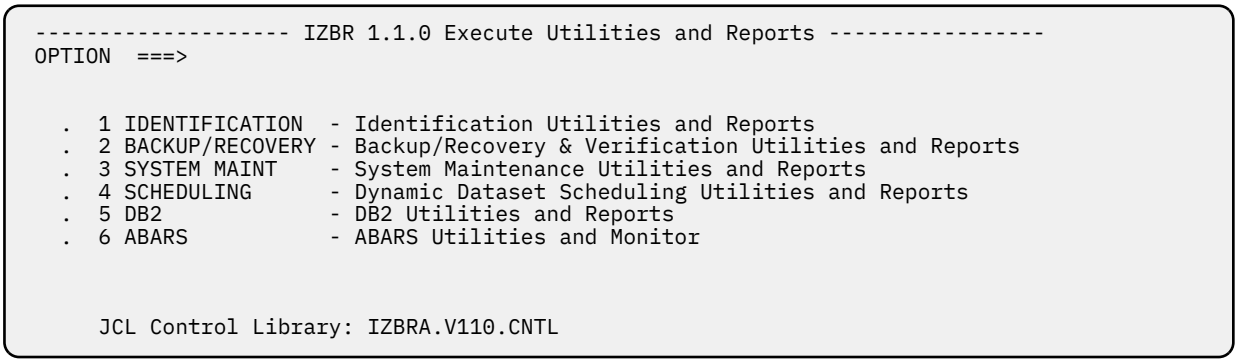
Before executing CA7JOB:

- Ensure you have authority to run CA7 jobs.
- Get a CA7 USERID and the number of a valid batch terminal to run the CA7 portion of this job.
- If you do not have authority, have someone create the forecast report for you using the FALL commands shown above.
- Store the forecast report in a file referenced by the CA7JOB JCL under the DDName of CA7RPT.
- You can then execute only the last two steps of this job to recreate the APPTABLE.
- If you plan to execute the CA7 portion of this job, place a valid CA7 USERID and password in member CNTL(CA7JOB), step FALLCMDS and update CA7 batch terminal information.

Modifying CA7JOB

Procedure

1. To make the FALLCMDS and batch terminal modifications for the CA7JOB JCL, go to the IBM Z Batch Resiliency Primary Panel and select option 3, EXECUTE. The Execute Utilities and Reports panel is displayed.



2. Select option 1, IDENTIFICATION, to view the Identification Component Utilities and Reports panel.
3. Type S next to the CA7JOB entry and then press the Enter key. This invokes edit mode, so you can make the necessary changes to the JCL.

   The CA7JOB JCL is displayed, as shown in the following example:

```
//jobname JOB ...
//*
//IZBRPROC JCLLIB ORDER=...
//*====================================================================
//* JCL TO CREATE APPTABLE FROM CA7 FORCAST REPORT
//*
//* STEP 1 - FALLCMDS - CREATE NEW FALL COMMANDS.
//* STEP 2 - BTERM   - CREATE CA7 FORCAST REPORT FROM FALL COMMANDS.
//*      2B- FREE    - CLOSE IF HIGH RETURN CODE
//* STEP 3 - CREATAB - CREATE APPTABLE FROM FORCAST REPORT.
//* STEP 4 - UPDATAB - UPDATE APPTABLE FROM SCANNED FORECAST
//*
//*--------------------------------------------------------------------
//* FIRST TIME SET-UP
//*--------------------------------------------------------------------
//*
//* 1) CHANGE ALL HHHHHHHH TO THE HIGH-LEVEL NODE(S) FOR CA7 FILES.
//*
//* 2) REVIEW PARAMETERS FOR CA7 BATCH TERMINAL AND MAKE ANY
//*    NECESSARY CHANGES.
//* 3) REVIEW INPUT PARM ARGUMENTS 'FALLCMDS NAME,PSWD #'
//*    NAME IS CA7 LOGON USERID, PSWD IS ASSOCIATED PASSWORD AND
//*    # IS NUMBER OF DAYS TO FORECAST, IE 3
//*
//*====================================================================
//*
//FALLCMDS  EXEC PGM=IRXJCL,PARM='FALLCMDS NAME,PSWD 3'
//OUTFILE  DD DSN=HLQ.CNTL(FALLCMDS),DISP=SHR
//SYSEXEC  DD DSN=HLQ.SBRNSRCE,DISP=SHR
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//*--------------------------------------------------------------------
//*--------------------------------------------------------------------
//*--------------------------------------------------------------------
//BTERM    EXEC PGM=SASSBSTR,PARM=1
//BATCHIN  DD  DISP=SHR,DSN=HHHHHHHH.CA7.BATCHI#1
//BATCHOUT DD  DISP=SHR,DSN=HHHHHHHH.CA7.BATCHO#1
//STEPLIB  DD  DISP=SHR,DSN=HHHHHHHH.CA7.LOADLIB
//SYSPRINT DD  DSN=&FORCAST,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(RECFM=FBA,LRECL=133,BLKSIZE=23408)
//SYSUDUMP DD  SYSOUT=X
//UCC7CMDS DD  DISP=SHR,DSN=HHHHHHHH.CA7.COMMDS
//SYSIN    DD  DSN=HLQ.CNTL(FALLCMDSS),DISP=SHR
... (continues)
```

4. In the CA7JOB JCL do the following.
   - Change all *HHHHHHHH* references to the high-level qualifiers for CA7 files.
   - Review the parameters for CA7 batch terminal and make any necessary changes. This would include:
     - The PARM=1 statement on the //BTERM and //FREE job steps.
     - The #1 characters on the //BATCHIN and //BATCHOUT DD statements.
     - Change NAME,PSWD on FALLCMDS parameter to the CA7 userid and password.
5. When you finish the modifications required for first time setup, submit the job by typing Z next to the CA7JOB entry on the Identification Component Utilities and Reports panel and pressing the Enter key.

   If step BTERM completes with a return code greater than zero, ensure that the following step, FREE, completes with a return code of zero. If not, rerun step FREE as a standalone job prior to rerunning the CA7JOB.

CA7JOB exits
Two exits are available for tailoring the execution of the CA7JOB in the IBM Z Batch Resiliency USERMAC library.

XCA7JOB
Called after the CA7 forecast is processed and new additions for the APPTABLE have been found.

XCA7JOB1

Called after the new additions are merged with the existing APPTABLE. These exits are typically used to delete unwanted jobs or applications, using the field names JOBNAME and APPLID. An example of code to delete test jobs is:

```
IF SUBSTR(JOBNAME,1,4) = 'TEST' THEN DROP = 1
```

Contact Technical Support for more information on using these exits.

CTMJOBB - Control-M APPTABLE creation utility

Use the IBM Z Batch Resiliency Control-M (CTM) utilities to create APPTABLE entries from the CTM scheduling files.

The procedure includes running the CTM report CTMRFLW and using IBM Z Batch Resiliency programs to extract the necessary job names and sequencing data. User exits allow you to further refine the APPTABLE.

To create APPTABLE entries from CTM input, use the following jobs:

- CTMLIST and CTMJOBB – job sequence that creates multiple APPTABLE entries, including one for each member in the CTM scheduling PDS, and the generated CTMBCHnn jobs.
- XCTMLIST – removes scheduling members from being processed by CTMLIST.

Creating multiple APPTABLE entries

About this task

The JCL sequence of: CTMLIST, CTMJOBB, and the generated CTMBCHnn jobs invoke the CTM utility to create an APPTABLE entry for each member in the Control-M scheduling PDS.

Procedure

1. Use the CNTL(CTMLIST) JCL to build USERMAC(CTMLIST), which is a list of all job names and scheduling information contained in the CTM scheduling PDS:

```
//JS10    EXEC  PGM=IKJEFT01,
//        PARM='CTMLIST IOA.CTM.SCHEDULE'  <<< MODIFY SCHEDULE PDS
//STEPLIB  DD  DISP=SHR,DSN=TECH.V110DEMO.LIBRARY
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//SYSEXEC  DD  DISP=SHR,DSN=TECH.V110DEMO.SBRNSRCE
//XCTMLIST DD  DISP=SHR,DSN=TECH.V110DEMO.USERMAC(XCTMLIST)
//OUTCTM   DD  DISP=SHR,DSN=TECH.V110DEMO.USERMAC(CTMLIST)
```

2. The CNTL(CTMJOBB) JCL uses USERMAC(CTMLIST) to build CNTL(CTMBCHnn) JCL for the next job execution in the series. The first member is named CTMBCH00; if additional members are needed, they are named incrementally (01, 02, . . .).

```
//jobname JOB ...
//JS10    EXEC  PGM=IKJEFT01,
//        PARM='CTMJOBB'
//STEPLIB  DD  DISP=SHR,DSN=HLQ.LIBRARY
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//SYSEXEC  DD  DISP=SHR,DSN=HLQ.SBRNSRCE
//XMAKEJCL DD  DISP=SHR,DSN=HLQ.USERMAC(XCTM0001)
//MAKEJCL  DD  DISP=SHR,DSN=HLQ.USERMAC(MAKEJCL1)
//UFIELDS  DD  DISP=SHR,DSN=HLQ.USERMAC(XCTM0001)
//TPLATE   DD  DISP=SHR,DSN=HLQ.USERMAC(ACTM)
//TEMPLATE DD  DISP=SHR,DSN=HLQ.USERMAC
//BKUPMETH DD  *
 CTM HLQ.CNTL. .
/*
```

```
//CTMLIST  DD  DISP=SHR,DSN=HLQ.USERMAC(CTMLIST)
//BKJCL    DD  DSN=&&CTMBJCL,DISP=(NEW,PASS),
//         UNIT=SYSDA,SPACE=(TRK,(1,1)),
//         DCB=(LRECL=133,BLKSIZE=0,RECFM=FB,DSORG=PS)
//*
//SNAPDSN  EXEC SNAPDSN,FILE=CTMBJCL
//*        DSN=HLQ.CTMBJCL
```

3. The CNTL(CTMBCH00) JCL is built from the execution of CNTL(CTMJOBB).
   - First CNTL(CTMBCH00) executes CTMRFLW followed by IRXJCL/CTMREAD for each CTM schedule PDS member listed in USERMAC(CTMLIST).
   - Next CNTL(CTMBCH00) executes IRXJCL/BRNCMJBR to update DD NEWAPPT USERMAC(APPTABLE).

```
//jobname  JOB ...
//*-------------------------------------------------------------
//*   AUTOGENERATED JCL FOR CONTROL-M SCHEDULE READS
//*   THIS JCL PRODUCES A JOB-FLOW REPORT FOR JOBS
//*   WHICH ARE DEFINED IN THE DAJOB DD STATEMENT.
//*   NOTE: *** CTMRFLW IS A NEW DIMENSION PROCEDURE
//*-------------------------------------------------------------
//*-------------------------------------------------------------
//*   ALLOCATE OUTPUT FILE FOR EACH SCAN
//*-------------------------------------------------------------
//ALLOC    EXEC PGM=IEFBR14
//OUTCTM   DD   DSN=&&OUTCTM,DISP=(MOD,PASS),
//              UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=0,RECFM=FB),
//              SPACE=(CYL,(5,5))
//*-------------------------------------------------------------
//*   SCAN OF SYSHLQ.CTM.SCHEDULE.PDS : MONTHJOB
//*-------------------------------------------------------------
//CTM0001 EXEC CTMRFLW
//DAJOB    DD  DISP=SHR,
//             DSN=SYSHLQ.CTM.SCHEDULE.PDS(MONTHJOB)
//DAREPORT DD  DSN=&&CTMRPT,DISP=(NEW,PASS),
//             UNIT=SYSDA,
//             SPACE=(CYL,(5,5),RLSE),
//             DCB=(LRECL=133,RECFM=FBA,BLKSIZE=23408)
//*-------------------------------------------------------------
//JS0001  EXEC PGM=IRXJCL,
//             PARM='CTMREAD MONTHJOB MONTHJOB'
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//SYSEXEC  DD  DSN=HLQ.SBRNSRCE,DISP=SHR
//INCTM    DD  DSN=&&CTMRPT,DISP=(OLD,DELETE)
//OUTCTM   DD  DSN=&&OUTCTM,DISP=(MOD,PASS)

... (continues)

//*-------------------------------------------------------------
//*  MERGE TOGETHER OLD APPTABLE AND NEW ENTRIES AND REMOVE
//*  DUPLICATE ENTRIES
//*-------------------------------------------------------------
//IZBR     EXEC PGM=IRXJCL,PARM=BRNCMJBR
//INCTM    DD  DSN=&&OUTCTM,DISP=(OLD,PASS)
//XCTMJOB  DD  DSN=HLQ.USERMAC(XCTMJOB),DISP=SHR
//XCTMJOB1 DD  DSN=HLQ.USERMAC(XCTMJOB1),DISP=SHR
//*-------------------------------------------------------------
//STEPLIB  DD  DSN=HLQ.LIBRARY,DISP=SHR
//SYSEXEC  DD  DSN=HLQ.SBRNSRCE,DISP=SHR
//WORK     DD  DSN=&&WORK,DISP=(NEW,PASS),
//             UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//             SPACE=(CYL,(5,5),RLSE)
//WORKO    DD  DSN=&&WORK2,DISP=(NEW,PASS),
//             UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//             SPACE=(CYL,(5,5),RLSE)
//DFSPARM  DD  DSN=&&DFSPARM,DISP=(NEW,PASS),
//             UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//             SPACE=(CYL,(5,5),RLSE)
//SYSIN    DD  DSN=&&SYSIN,DISP=(NEW,PASS),
//             UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//             SPACE=(CYL,(5,5),RLSE)
//SYSPRINT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
```

```
//APPTABLE DD  DSN=HLQ.USERMAC(APPTABLE),DISP=SHR
//NEWAPPT  DD  DSN=HLQ.USERMAC(APPTABLE),DISP=SHR
```

ESPJOB - ESP APPTABLE creation utility

Use the IBM Z Batch Resiliency ESP utilities to create APPTABLE entries from the ESP historical scheduling information.

The procedure first uses the appropriate ESP subsystem to obtain the specified ESP job history names relating to the application. First, jobs for each application are identified as jobs not released by an application process within ESP. The second utility, BRNESJBR, combines records extracted by the first step with the existing APPTABLE. User exit points are provided to modify the combined APPTABLE. Error codes and messages can be found in the *IBM Z Batch Resiliency Messages and Codes* manual section on ESP.

Creating multiple ESP APPTABLE entries

About this task

To create multiple application table entries from a specified ESP Scheduling PDS, use the BRNESBAT utility that executes ESPREAD for each ESP list entry.

Qualified APPTABLE names are passed to program BRNESJBR to be added to the APPTABLE.

To create a master list of all members:

Procedure

Run ESPJOB

- JS10 deletes the existing ESPJOB data set.
- JS20 adds ESP history to ESPJOB data set.
- JS30 parses the history to obtain job/application information.
- JS40 merges the jobs into the hlq.USERMAC(APPTABLE).

JS30 BRNESHST uses the file defined by DD RUNHIST to pass qualified entries to the next step. The JS40 step executes BRNVJOBR to add entries to the APPTABLE. Note that this step runs only if the prior step return code is less than 4, which signifies that there are eligible APPTABLE entries to add to the APPTABLE.

```
//*-------------------------------------------------------------------
//*  DELETE HISTORY REPORT FILE
//*-------------------------------------------------------------------
//JS10   EXEC PGM=IEFBR14
//SYSPRINT DD DSN=HLQ.ESPRHIST,
//            DISP=(MOD,DELETE),
//            UNIT=SYSDA,SPACE=(CYL,(15,15),RLSE),
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=0)
//*-------------------------------------------------------------------
//*  THIS JOB PRODUCES THE ESP JOB HISTORY REPORT
//*-------------------------------------------------------------------
//JS20   EXEC PGM=ESP,PARM='SUB(ESPM)'
//SYSPRINT DD DSN=HLQ.ESPRHIST,
//            DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(CYL,(15,15),RLSE),
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=0)
//SYSIN    DD DISP=SHR,DSN=HLQ.USERMAC(ESPPARM)
//*-------------------------------------------------------------------
//*  EXTRACT APPLICATION AND JOB NAMES FROM HISTORY REPORT
//*-------------------------------------------------------------------
//JS30   EXEC PGM=IRXJCL,PARM='BRNESHST',COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSOUT   DD SYSOUT=*
//SYSEXEC  DD DSN=HLQ.SBRNSRCE,DISP=SHR
//*====================================
//RUNHIST  DD DSN=HLQ.ESPRHIST,
//            DISP=(SHR)
//APPTABLE DD DSN=&&APPTABLE,DISP=(NEW,PASS),
//            UNIT=SYSDA,DCB=(LRECL=80,BLKSIZE=0,RECFM=FB),
//            SPACE=(CYL,(5,5))
//*-------------------------------------------------------------------
//*  MERGE TOGETHER OLD APPTABLE AND NEW ENTRIES AND REMOVE
//*  DUPLICATE ENTRIES
//*-------------------------------------------------------------------
//JS40    EXEC  PGM=IRXJCL,PARM=BRNVJOBR,COND=(4,LT)
//INSCH    DD  DSN=&&APPTABLE,DISP=(OLD,PASS)
//USERIN   DD  DSN=HLQ.USERMAC(XESPJOB),DISP=SHR
//USERIN1  DD  DSN=HLQ.USERMAC(XESPJOB1),DISP=SHR
//*-------------------------------------------------------------
//STEPLIB  DD DSN=HLQ.SBRNLOAD,DISP=SHR
//SYSEXEC  DD DSN=HLQ.SBRNSRCE,DISP=SHR
//WORK     DD DSN=&&WORK,DISP=(NEW,PASS),
//            UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//            SPACE=(CYL,(5,5),RLSE)
//WORKO    DD DSN=&&WORK2,DISP=(NEW,PASS),
//            UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//            SPACE=(CYL,(5,5),RLSE)
//DFSPARM  DD DSN=&&DFSPARM,DISP=(NEW,PASS),
//            UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//            SPACE=(CYL,(5,5),RLSE)
//SYSIN    DD DSN=&&SYSIN,DISP=(NEW,PASS),
//            UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PS),
//            SPACE=(CYL,(5,5),RLSE)
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//APPTABLE DD DSN=HLQ.USERMAC(APPTABLE),DISP=SHR
//NEWAPPT  DD DSN=HLQ.USERMAC(APPTABLE),DISP=SHR
//APPTDUP  DD DSN=HLQ.USERMAC(APPTDUP),DISP=SHR
```

Figure 25. Input to the ESPJOB utility

```
ESP VER  5.5 AM02 BATCH INTERFACE               07.02.27 WEDNESDAY APRIL 29TH, 2020

REPORT
HISTFILE HIST1
SETWIDTH 133
FROM 6AM YESTERDAY TO 6AM TODAY
CRITERIA ESPSUB EQ YES;
DISPLAY JOBNAME APPLSYS APPLGEN APPLTAG ACCOUNT INSYS JOBCLASS JOBNO
JOBQUAL EXECSDATE EXECST ENDT
SORT APPLSYS EXECSDATE EXECST
ENDR

ESP VER  5.5 AM02 BATCH INTERFACE               07.02.27 WEDNESDAY APRIL 29TH, 2020

JOBNAME  APPLSYS  APPLGEN APPLTAG          ACCOUNT  INPU JOB         JOB JOBQUAL  START        EXEC     EXEC
                                                    SYST CLASS        NO          DATE         START    END
ACXX160R #ACXXMSG   18207                           HCM  P          1165          TUE28APR20   09.39.12 09.39.18
APXX110D #ALXXMSG   44915                                P          1916          WED29APR20   01.00.39 01.00.44
ALXX399D #ALXXMSG   44916                                P          1923          WED29APR20   01.05.27 01.05.32
CSXX300S CSXX300S   13200                 HCM       HCM  P          1680          TUE28APR20   21.00.57 21.01.00
SGRPWTCH STORGRPT     650                           HCM  O          1095          TUE28APR20   08.00.02 08.00.04
SGRPWTCH STORGRPT     651                           HCM  O          1358          TUE28APR20   14.00.01 14.00.03
SGRPWTCH STORGRPT     652                           HCM  O          1584          TUE28APR20   20.00.11 20.01.56
SGRPWTCH STORGRPT     653                                O          1952          WED29APR20   02.00.08 02.00.22

SUBTOTAL (892)

TOTAL (892)
```

Figure 26. Output from the ESPJOB utility

```
  XESPJOB READ: 0
  XESPJOB1 READ: 0
OLD APPLICATION TABLE READ:13
WRITING OLD APP: 13
INVOKING SORT
SORT RC: 0
WRITING NEW APP: 287
```

Figure 27. Step JS40 contains the APPTABLE update

ESPJOB user exits

The following user exits are available for ESPJOB in the USERMAC library:

XESPJOB
front-end exit and is invoked before processing scheduling records.

XESPJOB1
back-end exit and is executed after the scheduling records have been processed by ESPJOB.

Both of these modules are empty when IBM Z Batch Resiliency is delivered.

Use exit USERMAC(XESPLIST) in job ESPLIST to bypass procedures based on the values of variables MEMBER, APPLID and DESC. For example if you want to bypass any member whose first four characters are BKUP:

```
IF SUBSTR(MEMBER,1,4) = 'BKUP' THEN DROP = 1
```

JOBTRAC - JOBTRAC APPTABLE creation utility

Use the IBM Z Batch Resiliency JOBTRAC utility to create APPTABLE entries from the JOBTRAC scheduling file.

The JOBTRAC utility creates APPLID entries for each member name in the scheduling PDS. Note that only valid JOBTRAC members should be in the JOBTRAC PDS processed by the utility. If there are any comment members, invalid results are produced.

The following JCL in CNTL(JOBTRAC) builds USERMAC(APPTABLE) based on existing APPTABLE entries plus new entries corresponding to the JOBTRAC scheduling PDS members.

To use this sample, specify the JOBTRAC scheduling PDS name in the PARM= line after %JOBTRAC.

```
//jobname JOB ...
//*====================================================================
//* PROGRAM:  JOBTRAC
//* FUNCTION: JCL TO CREATE/UPDATE APPTABLE FROM JOBTRAC SCHEDULE PDS
//*           EACH PDS MEMBER IS PROCESSED USING MEMNAME AS APPLID.
//*====================================================================
//STEP1    EXEC PGM=IKJEFT01,
//   PARM='%JOBTRAC XXXXXXXX.JOBTRAC.PROD.SCLLIB'
//SYSEXEC  DD DSN=HLQ.SBRNSRCE,DISP=SHR
//JOBTRAC  DD DSN=HLQ.USERMAC(JOBTRAC),DISP=SHR
//APPTABLE DD DSN=HLQ.USERMAC(APPTABLE),DISP=SHR
//SYSTSIN  DD DUMMY
//SYSTSPRT DD SYSOUT=*
```

IZWS APPTABLE creation utility

In job OPCJOB, IBM Z Batch Resiliency uses two IZWS reports as input to create the Application Table: The All Jobs report and the Inter-Dependencies report. These two reports are then parsed to identify the jobs, their sequences, and the first job or jobs for each application.

OPCJOB parameters

The job OPCJOB contains a JCL procedure named TWSPROC, which uses two parameters.

Table 8. OPCJOB parameters

Parameter	Description	Default Value
HLQ	The high-level qualifier used by IBM Z Batch Resiliency data sets	The defined high-level qualifier
TWSHLQ	The high-level qualifier used by the IZWS data set	SYS1.TWS

Job OPCJOB includes two SYSIN files that define sorting and selection parameters to the EQQAXR00 and EQQADPRT routines respectively.

Do not modify these parameters, but your production implementation can place them in a control statement PDS. They are located in the following statements:

```
//* Define TWS report sorting parameters: APPLICATION, OP ID, JOB NAME
//ADXREF.SYSIN DD *
010316
/*
```

```
//* Define TWS report selection parameters: dependencies for all applications
//ADPRT.SYSIN DD *
2100000000
/*
```

OPCJOBR exits

Program OPCJOBR calls two exits that allow the modification of the application name or the dropping of the application from inclusion in the APPTABLE.

- **XTWSJOB** – front-end exit and is invoked before processing scheduling records.
- **XTWSJOB1** – back-end exit and is executed after the scheduling records have been processed by OPCJOB. Both of these modules are empty when IBM Z Batch Resiliency is delivered.

The variables available to the exits are:

Table 9. Available XTWSJOB1 exit variables		
Variable	**Description**	**Comment**
JOBNAME	The name of the job in the application table.	
JOBIND	The first job indicator. If this is set to an asterisk (*), the job indicated by JOBNAME is a first job for the application; that is, the presence of that job in the SMF stream indicates that the application is starting.	
APPLID	The name of the application.	Change this variable if you want to have the application have a different name than the application known to IZWS.
DESC	An optional description for the application.	
DROP	An indicator of whether to drop the application from the APPTABLE.	Set this variable to 1 to drop the application or set it to 0 to include the application in the Application Table.

Differences between IZWS APPLIDs and IBM Z Batch Resiliency APPLIDs

Both IZWS and IBM Z Batch Resiliency use the concept of an application to logically group jobs. Each job name is defined as a member of an application when that job is defined to IZWS. However, differences exist in how IZWS and IBM Z Batch Resiliency handle APPLIDs and job assignments to APPLIDs. These differences are summarized as follows:

Table 10. IZWS APPLIDs / APPLID job assignment differences		
	IZWS APPLID structure	**IBM Z Batch Resiliency APPLID structure**
APPLID Length	1-16 characters	1-8 characters
Dependent Jobs	Jobs can belong to multiple Schedules/ APPLIDs	Jobs can belong to one and only one Schedules/APPLID

Processing OPCJOB

The code supplied with IBM Z Batch Resiliency:

- Runs the TWS All Jobs report
- Runs the TWS Inter-Dependencies report
- Parses the All Jobs output to identify the jobs that make up the applications, mapping the 16-character IZWS APPLIDs into their 8-character IBM Z Batch Resiliency equivalents, and deleting all job names belonging to more than one OPC APPLID.
- Parses the Interdependencies report to identify the first jobs for each application.
- Merges the outputs of the two steps above to develop an intermediate Application Table.
- Merges the intermediate Application Table with the existing Application Table.
- Deletes unwanted job names and applications or remaps job names to another more appropriate APPLID via user-developed exits.

For example:

```
/*================================================================*/
/*=  IZWS USER EXIT TO CUSTOMIZE BUILDING OF IZBR APPTABLE       =*/
/*================================================================*/
/******************************************************************/
/** EXCLUDE JOBNAMES WHICH ARE NOT PRODUCTION                    **/
/** SET DROP = 1 WHEN YOU WISH TO EXCLUDE, OTHERWISE DROP = 0    **/
/******************************************************************/
SELECT;
  WHEN (SUBSTR(JOBNAME,1,1) = '$')    THEN DROP = 1;
  WHEN (SUBSTR(JOBNAME,1,1) = '#')    THEN DROP = 1;
  WHEN (SUBSTR(JOBNAME,1,1) = '@')    THEN DROP = 1;
  OTHERWISE  DROP = 0;
END;  /* SELECT */
/******************************************************************/
/** SET APPLICATION NAME TO BE THE FIRST 2 CHARACTERS OF JOBNAME  **/
/******************************************************************/
APPLID = SUBSTR(JOBNAME,1,2);
```

IZWS security

It is likely that your production scheduling system has the necessary security access to the IZWS files and libraries, but during testing check with your security administrator to ensure that you can access these files.

ZEKEJOB - ZEKE APPTABLE creation utility

ZEKEJOB automatically creates a separate schedule report for the current day, along with all dates to be retained in the History database. It then uses this schedule report to determine which jobs comprise an application.

For example, if the current date is 06/06/18 and the retention parameter RETPD1 is set to 2, the following SCHEDULE commands are generated:

```
REPORT INCLUDE JOBS
SCHEDULE DATE 06/04/18

REPORT INCLUDE JOBS
SCHEDULE DATE 06/05/18

REPORT INCLUDE JOBS
SCHEDULE DATE 06/06/18
```

These SCHEDULE commands are input to ZEKE™ to create the schedule report. There is a separate SCHEDULE command for each day. This is necessary for marking the start of a particular daily cycle so that the first job for each application can be found. After IBM Z Batch Resiliency generates the SCHEDULE commands, ZEKE™ creates the schedule report. This report is contained in a file referenced in the last step of the JCL with the DD name ZEKERPT. The schedule report then recreates the APPTABLE from the information contained in the old APPTABLE and any newly found jobs from the current schedule report.

ZEKE prerequisites

- Before executing ZEKEJOB, determine if you have authority to run ZEKE jobs.
  - If you cannot obtain authority, have someone create the schedule report for you using the SCHEDULE commands shown above.
  - Place the report in a file referenced by the ZEKEJOB JCL with the DD name ZEKERPT.
  - You would then execute only the last step of this job to recreate the APPTABLE.
- To execute ZEKEJOB, begin from the IBM Z Batch Resiliency Primary Panel. Open the Execute Utilities and Reports panel by selecting option 3.
- Select option 1, IDENTIFICATION to view the Identification Component Utilities and Reports pane .
- To edit the JCL, type S next to the ZEKEJOB entry and press the Enter key.
- Make any changes necessary to the JCL and submit the job in one of these ways:

  - Enter SUBMIT on the command line to run the job.
  - If you did not edit the job, type Z next to ZEKEJOB on the above panel.
- To edit the exit code for ZEKE, type X next to ZEKEJOB. You are then taken to an edit session for the member XZEKEJOB in the USERMAC library. This member is initially empty, but you can use it to add exit code to delete applications from the schedule report, modify application IDs, remove jobs from the schedule report, or automatically create job masks.
- To submit the ZEKEJOB, type Z next to the ZEKEJOB entry on the Identification Component Utilities and Reports panel and press the Enter key.

Removing or excluding data sets and jobs from the collection process

There are two methods of removing or excluding data sets and jobs from further IBM Z Batch Resiliency processing.

To exclude certain SMF data from the IBM Z Batch Resiliency collection process, use the following members in USERMAC:

- **EXCLDSN** – excludes specific system data sets
- **EXCLJOB** – excludes job names from manually-created wildcard job entries in the APPTABLE

Removing data set names (EXCLDSN)

To remove data set names from critical data set identification, modify the USERMAC(EXCLDSN) member.

About this task

All SMF data related to the data set or series of data sets is excluded from further processing.

CAUTION: IBM Z Batch Resiliency does not monitor the execution of excluded jobs and back-up tracking is not active for data sets that these jobs use as input. IZBR cannot generate recovery JCL for unmonitored data sets.

Procedure

To exclude data sets, add one or more data set names to EXCLDSN using a list of data set names and masks.

You can use wildcards to indicate all data sets that follow a consistent pattern in the naming convention.

- Use a percent sign (%) or a question mark (?) as the wildcard indicator for a single position. For example, use %BC* as the data set mask
- Use an asterisk (*) to indicate any number of positions within a node. For example, the data set mask PROD* specifies all data sets that start with PROD.
- Use a double asterisk (**) to match all subsequent qualifiers. For example, *HLQ*.** matches *HLQ*.INFILE.TEST in addition to *HLQ*.INFILE.

For example, to exclude all GDG files with a GDG base of ABC.XYZ, enter:

```
ABC.XYZ.G%%%%V%%
```

Entries in EXCLDSN do not remove previous entries from the Inventory database or the critical file list. To remove entries from the Inventory database, use the Inventory maintenance function CNTL(INVMAINT) (see IBM Z Batch Resiliency Maintenance). To remove these entries from the critical file list, either edit that file or use the REMOVE utility.

Removing job names (EXCLJOB)

To remove a job or series of jobs from critical data set identification, modify the USERMAC(EXCLJOB) member.

About this task

All SMF data related to a job name or series of jobs is excluded from further processing.

Note: Use this facility with caution because IBM Z Batch Resiliency does not monitor the execution of excluded jobs, and therefore backup tracking is not active for data sets that they use as input. IBM Z Batch Resiliency cannot generate recovery JCL for unmonitored data sets.

Procedure

1. To remove jobs, add one or more job names to USERMAC(EXCLJOB). The format of EXCLJOB is a list of data set names and masks.
2. Use a percent sign (%) or a question mark (?) as the wildcard indicator for a single position and use an asterisk to indicate any number of positions at the end of the job name.

   For example:

   - To specify all jobs that start with ABC, use the job mask ABC*.
   - To specify all jobs that have BC in the second and third positions, use the job mask %BC*.
   - Do not use an asterisk in the first position or all jobs are excluded.

```
/*================================================================*/
/* JOBNAME EXCLUDES                                               */
/*================================================================*/
LLA*
```

Figure 28. Modifying USERMAC(EXCLJOB)

CAUTION: IBM Z Batch Resiliency does not monitor the execution of excluded jobs and back-up tracking is not active for data sets that these jobs use as input. IBM Z Batch Resiliency cannot generate recovery JCL for unmonitored data sets.

Setting up the RT/SMF Extractor

Real-Time SMF data is collected by the SMF Extractor (SMFEXT), and the SMF exits write the required records to extract files (SMF$).

About this task

As the SMF Extractor creates extract files, the Real-Time SMF Dispatcher processes the data. The SMF Extractor typically runs as a started task. You can also run the SMF Extractor as a batch job, as described in Batch SMF data collection.

Procedure

1. Verify that the HLQ.SCKKLOAD data sets are APF authorized.
2. Edit the JCL for your installation.

```
//SMFEXTP  PROC SYSTEM=BRA1                          /* SMF SYSTEM ID */
//SMFMAI   EXEC PGM=CKKXMAI,
//      PARM='TRACE(N),MSGLVL(9),SDUMP(Y),MAXSD(2),MAXQD(3000)'
//*
//STEPLIB  DD DISP=SHR,DSN=HLQ.SCKKLOAD
//SYSPRINT DD SYSOUT=*
//SMFXPARM DD DISP=SHR,DSN=HLQ.CNTL(SMFP&SYSTEM.)
//SYSUDUMP DD SYSOUT=*
//SMFXPRNT DD SYSOUT=*
```

See "SMF Extractor parameters" on page 56 for information about the SMFEXTP parameters.

Setting up the RT/SMF Extractor on multiple systems

About this task

To set up Real-Time SMF processing for multiple systems, begin by assigning a primary system as a collection point for all LPARS in the system complex. Each LPAR requires an SMF Extractor to create the SMF extract files for that system.

For Real-Time SMF to recognize the extracted files, edit SMFPARMS as described below.

For a description of supported SMFPARMS options, see SMF Extractor Parameters.

Procedure

1. Duplicate *hlq*.CNTL(SMFP xxxx) for each LPAR and edit:

   ```
   OUTDSN=HLQ.SMFEXT.SYS xxxx
   ```

   where xxxx is the name of the additional LPAR. Complete this for each additional LPAR. The name is limited to 28 characters or less.
2. Allocate an SMF checkpoint data set, using the existing *hlq*.CHKPOINT.SYS xxxx as a model for each additional LPAR.
3. For each system running the extractor, modify the SYSID= parameter and SMFDSN= parameter to be consistent with each LPAR system in *hlq*.USERMAC(SMFPARMS).

   ```
   SYSID=xxxx,SCRATCH=YES,SMFDSN=HLQ.SMFEXT.SYS xxxx
   ```

Enabling the SMF exits

If z/OS is less than V2R3, you need to define SMF exits IEFU83, IEFU84 and IEFU85 in SMFPRMxx. If z/OS is V2R3 or above, you need to define SMF exit IEFU86 in SMFPRMxx.

About this task

SMF.MANx data sets and the system logger, used with previous releases of IBM Z Batch Resiliency, can be used as an alternative to the SMF exit. See SMF.MANx Data Sets and System Logger for more information.

Procedure

1. Verify that the *hlq*.SCKKLOAD data sets are APF authorized.
2. Check the SYS and SUBSYS parameters in the PARMLIB member SMFPRMxx. Ensure SYS and SUBSYS are defined as follows:

   ```
   SYS(EXITS(IEFU83,IEFU84,IEFU85))
   SUBSYS(STC,EXITS(IEFU83,IEFU84,IEFU85)
   ```

Or:

```
SYS(EXITS(IEFU86))
SUBSYS(STC,EXITS(IEFU86)
```

SYS and SUBSYS define the z/OS system exit, which are required for the SMF Extractor to add the exit. Refer to *IBM z/OS MVS Initialization and Tuning Reference* for additional information.

3. Add SMFEXTP to a system procedure library to initiate the started task.
4. Issue the console command to ensure the SMF exits are active.

   The console command displays similar to the following:

```
D PROG,EXIT,EN=SYS.IEFU83,DIAG
D PROG,EXIT,EN=SYS.IEFU84,DIAG
D PROG,EXIT,EN=SYS.IEFU85,DIAG
```

Or:

```
D PROG,EXIT,EN=SYS.IEFU86,DIAG
```

Defining Real-Time SMF subtasks (VFISTASK) for the Real-Time Dispatcher

About this task

The USERMAC(VFISTASK) member defines the subtasks that the RT/SMF Dispatcher processes for the associated record types. The RT/SMF Dispatcher reads extracted SMF data and calls the appropriate subtask for that record type.

The following figure shows the default subtasks in VFISTASK.

```
*****************************************************************
* REAL-TIME DISPATCHER SUB-TASK STARTUP PARAMETERS               *
*                                                               *
*****************************************************************
START TASK=CMDPROC,PGM=CKKRC004                              < CMD PROCSR
*
START TASK=BRNVANLY1,PGM=BRNVC001,SMF=(14,15,30,61,64,65)  < ANALYSIS
START TASK=SMFCOUNT,PGM=CKKRT999,SMF=(0:127)                < SMF COUNTER
START TASK=IZBR/RTB,PGM=BRNVC007V,SMF=(15,60)               < BKUPTBL
START TASK=RT/HNDLR,PGM=CKKRC033                            < HEART BEAT
*START TASK=IZBR/CKK,PGM=BRNVC008,SMF=(30)                   < RT/SCHED
*START TASK=IZBR/VRS,PGM=BRNVC011                            < RECOVRY SIM
*START TASK=SCANSMF,PGM=BRNVC031,SMF=(14,15,30,64,202)       < SCANSMF
*START TASK=INVMAINT,PGM=BRNVC059                            < RT/INVMAINT
*START TASK=RT/ICP,PGM=CKKRC036,SMF=(30)                     < RT/ICP
*START TASK=UFITASK,PGM=BRNVC065,SMF=(15,30,64,65)           < RT/UFI
```

Table 11. SMF record types used by basic RT/SMF subtasks

SMF type	Description
14	Close for input data sets
15	Close for output data sets
30	Step and job initiation and termination
60	VSAM volume data set updates
61	ICF define activity (ICF catalogs)
64	VSAM component or cluster status

Table 11. SMF record types used by basic RT/SMF subtasks (continued)	
SMF type	**Description**
65	VSAM delete activity
202	IBM Z Batch Resiliency input record for files backed up by DSS

The RT/SMF Dispatcher accesses SMF data collected by the RT/SMF Extractor and distributes it to the subtasks that are defined to receive it according to the rules in the VFISTASK file. For example, using the example VFISTASK file in Defining Real-Time SMF subtasks (VFISTASK) for the Real-Time Dispatcher, when an SMF 15 record comes in, a copy is sent to four different tasks - BRNVANLY1, IZBR/RTB, SMFCOUNT and RT/HNDLR.

Starting the Real-Time SMF Dispatcher (RTSMFP)

About this task

You can start RTSMFP to run the process as a started task, or schedule RTSMF as a batch job. For more information, see Real-Time SMF Dispatcher.

Procedure

1. Before running the SMF dispatcher, ensure that the STEPLIB data sets are APF authorized.

   Note that APF authorization of the IBM Z Batch Resiliency Analysis subtask is not required during a proof of concept, such as when you are using SMF extract data sets.
2. Review the JCL statements in CNTL(RTSMFP) and edit for your installation. If you need to change any of the DD statements, see SMF parameters (SMFPARMS) for more information.
3. Run RTSMFP as a started task or schedule RTSMF as a batch job.

Setting up TimeLiner for ISPF

To set up Interactive System Productivity Facility (ISPF) for TimeLiner, you need to first initialize the Data-In-Virtual (DIV) data space.

About this task

The configuration settings are saved in the user ISPF profile. Therefore, each user who accesses TimeLiner from ISPF must complete these setup steps.

Procedure

1. From the ISPF environment, run the following command for each user's ISPF profile.

   ```
   TSO EX HLQ.CNTL(BRN110)
   ```
2. From the IBM Z Batch Resiliency Primary Panel, select option **8 TLN - TimeLiner Functions**.
3. From the TimeLiner Main Menu, select option 0, Configuration. From the TimeLiner Preferences panel, enter the DIV and the IBM Z Batch Resiliency high-level qualifier.
4. Optional: To set up the ISPF interface to the Dell EMC Data Protector for Z Systems zDP feature, enter:
   - UFI data set name
   - zDP data set name

```
--------------------------- TimeLiner Preferences ---------------------
COMMAND ===>

Debug Trace  =>  N (Y/N)

DIV          =>  IZBRA.V110.TLN.DIV
System        :  SVSCPLEX
Logstream     :  SYSPLEX.TIMELINA.JOURNAL

VFI HLQ      =>
UFI dataset  =>
zDP dataset  =>

                Specify options and press ENTER to continue
```

See Starting TimeLiner for information about the TimeLiner started task.

Configuring the IBM Z Batch Resiliency zDP interface

About this task

To enable the Dell EMC Data Protector for Z Systems zDP product SnapVX feature, modify the Real-Time SMF Dispatcher, SMFP&*systemid*, and VFISTASK members as follows:

Procedure

1. To enable the zDP audit data set, uncomment the following DD statement in RTSMFP:

```
ZDPAUDIT DD DSN=QA.IZBR.V1R1M0.TLN.ZDPAUDT,DISP=SHR
```

2. To enable UFI, uncomment the following DD statements in RTSMFP:

```
UFIFILE  DD DSN=QA.IZBR.V1R1M0.UFIFILE,DISP=SHR
UFICKPT  DD DSN=QA.IZBR.V1R1M0.UFICKPT,DISP=SH
```

3. To extract records for zDP snapsets, add SMF record type 225, or any record number you choose, to CNTL(SMFP&systemid).
4. Update USERMAC(VFISTASK) for the TimeLiner zDP interface:

```
START TASK=TLNZDP,PGM=TLNZDP01,SMF=(225)          < EMC SMF
```

Chapter 4. SMF Data Extraction

The system management facility (SMF) collects and records z/OS system events identified by record types. SMF record types are the primary source of information for IBM Z Batch Resiliency to determine how a data set is referenced within an application. IBM Z Batch Resiliency uses the SMF Extractor and user exit to collect select records for analysis and processing.

The Real-Time SMF Extractor

The IBM Z Batch Resiliency SMF extractor and SMF exits continuously extract specific SMF record types for analysis and processing. SMF records are output to SMF$ extract files which are processed by the Real-Time SMF (RT/SMF) Dispatcher.

The IBM Z Batch Resiliency interface with the z/OS SMF exits IEFU83, IEFU84, and IEFU85, or IEFU86.

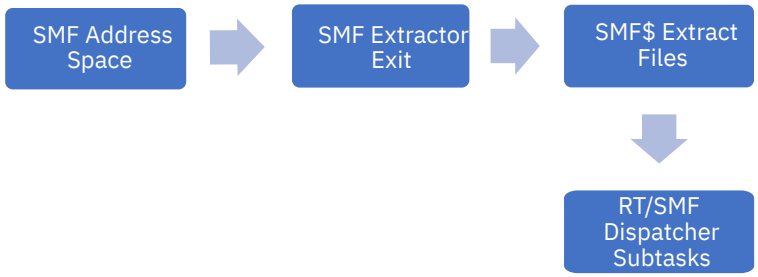


Figure 29. SMF Extractor and SMFXU83 exit job flow

The Real-Time SMF Dispatcher analyzes SMF data and dispatches SMF records to several subtasks for processing. The primary subtasks include collection and analysis of SMF records and updating backed-up critical data sets information in the Backup Table. For a description of all subtasks of the RT/SMF Dispatcher, see Real-Time SMF Dispatcher.

SMF Extractor JCL

The Real-Time SMF Extractor can be used to feed SMF data to multiple products by either Data set or Log stream capabilities. You can specify up to six **OUT*DSN** and **OUT*LGR** parameters within the SMFPsysid member in the CNTL library. The Extractor supports extended SMF records if the z/OS system is V2.3 or later. When z/OS is pre V2.3, the SMF exits IEFU83, IEFU84 and IEFU85 are loaded. z/OS V2.3 or higher only loads SMF exit IEFU86.

You can run the SMF Extractor as a started task or as a batch job.

- To run the SMF Extractor as a started task, place the CNTL(SMFEXTP) member in a system procedure library.
- To run the extractor as a batch job, use the CNTL(SMFEXT) member.

SMF Extractor programs and parameters

The following are the SMF Extractor programs and parameters:

SMFEXTP
The procedure that runs the SMF Extractor as a started task. This JCL resides in *hlq*.CNTL library. You can place this procedure into a system procedure library to be initiated as a started task

SMFPxxxx
Parameters that control the execution of the SMF Extractor. The xxxx represents the system ID under which the extractor was set up. It resides in the HLQ.CNTL library. See SMF Extractor Parameters for details of the parameter values.

© Copyright IBM Corp. 2019, 2020

SYSTEM
System ID that is generated from the Modify option during installation.

SMFEXT
Member SMFEXT in the *hlq*.CNTL library is the JCL required to run the SMF Extractor in batch mode.

```
//*============================================================*/
//*                                                            */
//*  SMF EXTRACTOR STARTED TASK PROC                           */
//*                                                            */
//*  THIS JCL IS THE SMF EXTRACTOR STARTUP PROC. SYMBOLIC USAGE, */
//*  EXEC CARD PARAMETERS,AND JCL DD STATEMENTS ARE EXPLAINED    */
//*  BELOW.                                                    */
//*                                                            */
//*============================================================*/
//SMFEXTP  PROC SYSTEM=BRA1               /* SMF SYSTEM ID */
//SMFMAI   EXEC PGM=CKKXMAI,
//    PARM='TRACE(N),MSGLVL(9),SDUMP(Y),MAXSD(2),MAXQD(3000)'
//*
//STEPLIB  DD DISP=SHR,DSN=HLQ.SCKKLOAD
//SYSPRINT DD SYSOUT=*
//SMFXPARM DD DISP=SHR,DSN=HLQ.CNTL(SMFP&SYSTEM.)
//SYSUDUMP DD SYSOUT=*
//SMFXPRNT DD SYSOUT=*
```

Figure 30. SMF Extractor started task

Parameters

The SMF Extractor started task parameters are as follows:

TRACE
Include the diagnostic trace in the output. Value can be Y or N.

MSGLVL
Defines the message level. Value are 0-9.

SDUMP
SVC Dump. Value is Y or N.

MAXSD
Maximum number of SVC dumps. Default is 2.

MAXQD
Maximum queue depth for SMF records. Range is 1000 – 9999. Default is 3000.

SMF record types and exit support

SMF configuration is performed with PARMLIB member SMFPRMxx. SMF configuration consists of the SMF record types being recorded and the SMF exits defined - as shown in the following table.

Table 12. SMF record types

SMF record	SMF function
Type 14	Close for input data sets
Type 15	Close for output data sets
Type 30	Step and job initiation and termination
Type 60	VSAM volume data set updates
Type 61	ICF define activity (ICF catalogs)
Type 64	VSAM component or cluster status
Type 65	VSAM delete activity

Table 12. SMF record types (continued)	
SMF record	**SMF function**
Type 202	IBM Z Batch Resiliency DSS input record

See Real-Time Dispatcher subtasks for a description of the SMF record types processed by each RT/SMF Dispatcher subtask. Contact the systems programming group responsible for SMF to confirm that the required SMF records are being collected and stored in the file referenced by the //SMFDATA DD card in the CNTL(RTSMF) member.

SMF system recording options

The SMFPRMxx parameters SYS and SUBSYS define SMF recording options:

- SYS – specifies system-wide recording options for all subsystems (JES2, TSO, STC, and so on).
- SUBSYS – specifies subsystem-specific recording options. The SUBSYS parameter options override the equivalent options specified on the SYS parameter. Any SUBSYS parameter option not specified defaults to the SYS parameter option.

The following example illustrates how SMF SYS and SUBSYS parameters work together:

```
SYS(EXITS(IEFU83,IEFU84,IEFU85))
SUBSYS(STC,EXITS(IEFU83,IEFU84,IEFU85))
```

Or:

```
SYS(EXITS(IEFU86))
SUBSYS(STC,EXITS(IEFU86)
```

The SYS parameter EXITS option defines the SMF exits IEFU83, IEFU84, and IEFU85, or IEFU86. The SUBSYS parameter for the STC subsystem also specifies the EXITS option, overriding the SYS EXITS option. The SUBSYS parameter EXIT option defines SMF exits IEFU83, IEFU84, and IEFU85, or IEFU86, for the STC subsystem.

IBM provides dynamic exit facility to add and remove exit routines dynamically. IBM has defined all the SMF exits to the dynamic exit facility.

The exit names used in the dynamic exit facility are formatted as SYS*subsystem_name.exitname*, where *subsystem_name* is the subsystem name, or blank if it is a system wide exit, and *exitname* is the name of the exit, such as IEFU83. The following provides some example dynamic exit facility exit names.

SYS.IEFU83
The default SMF exit IEFU83 for all system workload types.

SYSSTC.IEFU83
The SMF exit IEFU83 for STC subsystem workload.

Batch SMF data collection

You can use the CNTL(COLLECT) member to collect and analyze SMF records in batch mode as a preliminary function for collecting historical data sets from your applications.

This process provides an initial set of historical data when IBM Z Batch Resiliency is first implemented. SMF data is assigned through the value of the DD statement SMFDATA in CNTL(COLLECT). The data that is fed to COLLECT must be sorted in chronological order.

```
//SMFDATA   DD DISP=SHR,DSN=sorted SMF
```

For instructions on preparing SMF data for the COLLECT job, see Sorting SMF data for batch.

The following figure shows sample JCL for the batch collection process.

```
//*====================================================================
//* PROGRAM:  COLLECT
//* FUNCTION: UTILITY WHICH COLLECTS SMF DATA.
//*                        RTSRTD00 => STOPAFT SET
//*====================================================================
//* BUILD SMF DATABASE
//*====================================================================
//GO       EXEC PGM=CKKRC001,
//                       REGION=0M,
//                       PARM=RTSRTD00
//*
//STEPLIB   DD DSN=HLQ.SCKKLOAD,DISP=SHR
//          DD DSN=HLQ.SBRNLOAD,DISP=SHR
//*
//SMFDATA   DD DISP=SHR,DSN=HLQ.SMF
//PARMLIB   DD DISP=SHR,DSN=HLQ.USERMAC
//INVENTRY  DD DISP=SHR,DSN=HLQ.INVENTRY
//INVENTRZ  DD DISP=SHR,DSN=HLQ.INVENTRY
//CMDQUEUE  DD DISP=SHR,DSN=HLQ.RTCMDQ
//APPTABLE  DD DISP=SHR,DSN=HLQ.USERMAC(APPTABLE)
//APPTABLX  DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=(RECFM=FB,LRECL=64)
//BKUPTBL   DD DSN=HLQ.BKUPTBL,DISP=SHR,
//           AMP=('BUFNI=12,BUFND=49')
//*
//AUDTDS    DD DISP=SHR,DSN=HLQ.AUDTDS
//PRODDS    DD DISP=SHR,DSN=HLQ.PRODDS
//*
//CHECKPT   DD DISP=SHR,DSN=HLQ.RTACKPT
//ODTCKPT   DD DISP=SHR,DSN=HLQ.ODTCKPT
```

Figure 31. Sample batch collection process JCL

Sorting SMF data for batch

Before data can be read by COLLECT, it must be sorted in chronological order. Use the CNTL(SMFSORT) member as a sample to arrange SMF data in proper sequence.

In the event that batch SMF data is already sorted by SMF date and time, only the DD SMFDATA is required.

SMF Extractor parameters

The SMFPARMS options are distributed in member SMFPxxxx in the CNTL library. During installation, this member is copied to one or more members as SMFP*systemid*, where *systemid* is the SMF system ID of the systems which runs an SMF Extractor. All options can start in any column; only one option is permitted per statement.

Any characters beyond the value field are treated as comments and are ignored. There are no spaces between the option keyword, the equal sign, and the value of the parameter.

The following table describes the supported SMFPARMS options:

Table 13. SMFPARMS options

Parameter	Default	Range	Description
BLKSIZE=*nnn*	4096	4096 to 32767	The block size for the extractor file data sets. The block size of this file should match the Control Interval Size (CISIZE) of the SYS1.MANx data sets.
BUFND=*nnn*	10	1 to 100	Number of VSAM buffers allocated to the SMF data set.

Table 13. SMFPARMS options (continued)

Parameter	Default	Range	Description
CHECK#=*nnn*	25	1 to 255	Number of extract file records to be written between SMF.MANx checkpoints; The checkpoint facility allows the SMF Extractor to resume.
DYNDD=*ddname*	SMF		DDNAME to be used to dynamically allocate the SMF extract files created by the SMF Extractor. Only the CKKSRDRV program in the Collect job uses this parameter.
DYNMAXRC=*max_rc*	4	0 to 4095	The maximum allowable return code from the program whose name is specified in the DYNPGM parameter. If the program ends with a return code less than or equal to DYNMAXRC, the SMF extract files are deallocated and deleted upon termination of DYNPGM; otherwise, the extract files are deallocated but not deleted. Only the CKKSRDRV program in the Collect job uses this parameter.

DYNPARM=(*program_parms*)	None		Program parameters to be passed to the program whose name is specified in the DYNPGM parameter; If omitted or defaulted, no parameters are passed.
			Enclose the parameter with single quotes if single quotes or spaces are part of the parameter string. If single quotes are part of the string, you must use 2 quotes for each quote to be passed as part of the string. For example, if you want to pass the following parameter:
			`DSN=ABC,MYLIB='DEF'`
			then code DYNPARAMETER as follows:
			`DYNPARM='DSN=ABC,MYLIB=''DEF'''`
			Only the CKKSRDRV program in the Collect job uses this parameter.

DYNPGM=*module_name*	SMFREAD		Name of the load module to be given control once the SMF extract files created by the SMF Extractor have been allocated. Only the CKKSRDRV program in the Collect job uses this parameter.
EOVMSG=*nnn*	0	0 to 32767	The number of logical end-of-volume (EOV) messages that are written to the console. An EOV event occurs at the end of each CKKSRDR interval (specified by the RDRTIME option below). CKKSRDR reads the current SMF.MANx file until no more records are found (that is, logical EOV). A specification of 0 means that no EOV messages are written to the console.

Table 13. SMFPARMS options (continued)

Parameter	Default	Range	Description
IBUFS=*nn*	50	50 to 512	Calculates the number of internal buffers. This value should match the BUFND parameter for the SYS1.MANx data sets.
MSGID=*message_prefix*			Allows for a specific message prefix.
NEWTIME=*hhmm*	0010	0000 to 2359	Interval for creation of a new extract file, in *hhmm* (hours/minutes) format. Applies for all OUTDSN and OUTLGR parameters.
OUTDREC=*n, x, y, z* OUT1DREC=*n, x, y, z* OUT2DREC=*n, x, y, z* OUT3DREC=*n, x, y, z* OUT4DREC=*n, x, y, z* OUT5DREC=*n, x, y, z* OUTLREC=*n, x, y, z* OUT1LREC=*n, x, y, z* OUT2LREC=*n, x, y, z* OUT3LREC=*n, x, y, z* OUT4LREC=*n, x, y, z* OUT5LREC=*n, x, y, z*	14, 15, 30, 60, 61, 64, 65	0 to 255 or 0 to 2048	SMF Record types to be collected by each OUT*REC or OUT*LGR. If not specified, then the values in the SMFREC or SMFXREC parameter below are used.
OUTDSN=*Base_DSName* OUT1DSN=*Base_DSName1* OUT2DSN=*Base_DSName2* OUT3DSN=*Base_DSName3* OUT4DSN=*Base_DSName4* OUT5DSN=*Base_DSName5* OUTLGR=*Base_LSName* OUT1LGR=*Base_LSname1* OUT2LGR=*Base_LSName2* OUT3LGR=*Base_LSName3* OUT4LGR=*Base_LSName4* OUT5LGR=*Base_LSName5*	None		Base name used to allocate the SMF extract files. The Extract files can be an Output Data set or a log stream. This name must be a valid MVS data set name from 1 to 28 characters in length and must not be a GDG base name. Each OUTDSN and OUTLGR can use its own SMF record filter, specified by the OUTDREC and OUTLREC parameters. The SMF Extractor allocates extract files with the following data set name format: *outdsn*.SMF$*yy*.T*dddhhmm* or *outlgr*.SMF$*yy*.T*dddhhmm*, where: • *outdsn* is the value of the OUTDSN parameter • *outlgr* is the value of the OUTLGR parameter • *yy* is replaced with the current year • the Julian day of the year replaces *ddd* • *hh* is the current hour • *mm* is the current minute At least one OUTDSN or OUTLGR parameter must be specified.

Table 13. SMFPARMS options (continued)

Parameter	Default	Range	Description
PRIMARY=*nnn*	10	1 to 32767	Primary cylinder space allocation for extract data sets. Applies to all OUTDSN parameters.
SECONDARY=*nnn*	10	0 to 32767	Secondary cylinder space allocation for extract data sets. Applies to all OUTDSN parameters.
SMF$MAX=*nn*	144	0-512	The maximum number of SMF$ data sets that the extractor creates when the RTA environment is not active. A zero (0) indicates no limit is used.
SMFREAD=YES/NO	NO		Indicates whether the Collector (that is, the file represented by the //READJCL DD) is submitted by this SMF Extractor task. If you are using the RTSMF started task, specify SMFREAD=NO for all systems.
SMFREC=*n,x,y...,z*	14, 15, 30, 60, 61, 63, 64	0 to 255	SMF record types to be collected by the SMF Extractor. Each value must be a valid numeric from 0 to 255, separated by a comma with no intervening blanks. Alternatively, a range of values can be specified as *nnn*:*ppp*, where *nnn* denotes the start of the range, and *ppp* denotes the end of the range (for example, 60:64 specifies SMF record types 60 through 64, inclusive). An SMF record type of asterisk character (*) resets all prior specified types (for example, SMFREC=24,28,*,32,80:84 would result in SMF record types 32, 80, 81, 82, 83, and 84 collection). If you want to specify more SMF types than can fit on a single statement, specify multiple SMFREC option statements (as many as necessary).
SMFXREC=*n,x,y...,z*	14, 15, 30, 60, 61, 64, 65	0 to 2048	Same as SMFREC above. Requires IEFU86 and z/OS V2.3 or above.
SUBCLASS=@	1		The internal reader sysout class to be used to submit the collector job. An internal reader is dynamically allocated by CKKSFSUB with this sysout class in order to submit the Collector batch job.
SUBTIME=*hhmm*	None	Cannot be specified	The default is none and cannot be specified. Remove this option if you do not want to use the extractor submit feature.
UNIT=*dasd_unitname*	SYSDA		Extract data set DASD unit name (generic or esoteric). Must be 8 characters in length. Applies to all OUTDSN parameters.
VOLSER=*volume_serial*	None		Extract data set volume serial. Must be 6 characters. Applies to all OUTDSN parameters.

SMF Extractor console commands

You can use these console commands to manage the SMF Extractor started task.

Change *smfext* to match the name of your SMF Extractor:

S *smfext*,SYSTEM=*sysname*
Start the SMF Extractor on system *sysname*.

P *smfext*
Terminate the SMF Extractor.

F *smfext*,C option=*newvalu*
Change a startup option to a new value while the SMF Extractor is running. Any SMF Extractor startup option can be modified.

F *smfext*,STAT COUNTS / PARMS / DEBUG / ALL
Request statistics:

- COUNTS displays number of SMF records read from the SMF VSAM files and the number written to the extract files. The counts are cumulative.
- PARMS displays current values of startup options.
- DEBUG displays additional status debug information.
- ALL lists the COUNTS, PARMS and DEBUG information.

F *smfext*,NEWFILE
Create a new extract file immediately.

F *smfext*,RESET
Reset the logical EOV counter (related to SMFPARMS EOV startup option).

F *smfext*,SUBMIT
Submit the Collect job immediately.

F *smfext*,VER
Display SMF Extractor Version/Release/Modification level.

F *smfext*,STATUS|STAT PROCESS
The PROCESS parameter displays differs depending on the z/OS system version. Pre z/OS V2.3 only displays SMF record types 0 - 255. z/OS V2.3 or higher displays SMF record types 0 - 2047.

F *smfext*,STATUS|STAT PROCESS(ZERO)
The PROCESS(ZERO) parameter only displays SMF record types that have a zero count.

F *smfext*,STATUS|STAT PROCESS(NZERO)
The PROCESS(NZERO) parameter only displays SMF record types that have a non-zero count.

Chapter 5. Real-Time SMF Dispatcher

The IBM Z Batch Resiliency real-time system management facility (RT/SMF) processes SMF data continuously and automates the collection and analysis of SMF data in real time.

The Real-Time SMF Dispatcher compares each SMF record type with a subtask table and calls the appropriate subtask for that record type. The following figure is an example of the Real-Time SMF Dispatcher RTSMFP JCL:

```
//RTSMF    PROC RTSPARM=VFIRTD00
//*
//*            RT/SMF - REAL-TIME DISPATCHER
//*
//GO               EXEC PGM=CKKRC001,
//                 REGION=0M,
//                 PARM=&RTSPARM
//*
//STEPLIB      DD DISP=SHR,DSN=HLQ.SCKKLOAD
//                DD DISP=SHR,DSN=HLQ.SBRNLOAD
//*
//PARMLIB      DD DISP=SHR,DSN=HLQ.USERMAC
//INVENTRY     DD DISP=SHR,DSN=HLQ.INVENTRY
//INVENTRZ     DD DISP=SHR,DSN=HLQ.INVENTRY
//CMDQUEUE     DD DISP=SHR,DSN=HLQ.RTCMDQ
//APPTABLE     DD DISP=SHR,DSN=HLQ.USERMAC(APPTABLE)
//APPTABLX     DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=(RECFM=FB,LRECL=64)
//BKUPTBL      DD DSN=HLQ.BKUPTBL,DISP=SHR,
//                        AMP=('BUFNI=12,BUFND=49')
//BKUPAUD      DD DSN=HLQ.BKUPAUD,DISP=SHR
//*
//AUDTDS       DD DISP=SHR,DSN=HLQ.AUDTDS
//PRODDS       DD DISP=SHR,DSN=HLQ.PRODDS
//*
//CHECKPT      DD DISP=SHR,DSN=HLQ.RTACKPT
//ODTCKPT      DD DISP=SHR,DSN=HLQ.ODTCKPT
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENT IF:
//******    RT/SCANSMF IS ENABLED
//SSPCKPT      DD DISP=SHR,DSN=HLQ.SSPCKPT
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENT IF:
//******    DYNAMIC DATASET SCHEDULING IS ENABLED.
//SCHDJDS      DD DISP=SHR,DSN=HLQ.SCHDJDS
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENTS IF:
//******    VIRTUAL RECOVERY SIMULATOR IS ENABLED.
//SIMTSNAP     DD DSN=HLQ.SIMTSNAP,DISP=SHR
//VRSCKPT      DD DSN=HLQ.VRSCKPT,DISP=SHR
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENTS IF:
//******    KEYED HISTORY COLECTION IS UPDATED DYNAMICALLY.
//*KEYDHIST     DD DSN=HLQ.KEYDHIST,DISP=SHR
//*KEYJHIST     DD DSN=HLQ.KEYJHIST,DISP=SHR
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENTS IF:
//******    USING UPDATED FILE IDENTIFIER (UFI) FEATURE.
//*UFIFILE      DD DSN=HLQ.UFIFILE,DISP=SHR
//*UFICKPT      DD DSN=HLQ.UFICKPT,DISP=SHR
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENTS IF:
//******    USING EMC SMF RECORDS.
//*ZDPAUDIT     DD DSN=HLQ.TLN.ZDPAUDT,DISP=SHR
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENT IF:
//******    RT/HSMTASK IS ENABLED
//*HSMCKPT      DD DSN=HLQ.HSMCKPT,DISP=SHR
//*
//******    UNCOMMENT THE FOLLOWING DD STATEMENT IF:
//******    TRACKING TCP/IP FILES
//*TTFVFILE     DD DSN=HLQ.TTFVFILE,DISP=SHR
//*
//RTDPRINT     DD SYSOUT=*
//SNAP         DD SYSOUT=*
//SYSPRINT     DD SYSOUT=*
//SYSABEND     DD SYSOUT=*
//VFIDUMP      DD SYSOUT=*
//SYSOUT       DD SYSOUT=*
//*
```

Figure 32. Example of the Real-Time SMF Dispatcher RTSMFP JCL

SMF data is input to the RT/SMF Dispatcher. The address space continues to run and looks for extract files (SMF$) created by SMF Extractors. RT/SMF Dispatcher reads the SMF extract files and passes to appropriate subtasks such as IBM Z Batch Resiliency analysis and Backup Table processing.

See SMF parameters (SMFPARMS) for a description of the DD statements in RTSMFP.

© Copyright IBM Corp. 2019, 2020

SMF data not available – CKKR03014A message

When SMF records are not being produced for a defined system that is running the SMF Extractor, or an identified system is not available, the message CKKR03014A is issued at the system console. See “CKKR03014A” on page 298 for more information. Respond by entering CONTINUE or REMOVE:

CONTINUE
Instructs RT/SMF Dispatcher to keep looking for an SMF$ data set that matches the mask for that system. RT/SMF continues to look for a match if no response is issued

REMOVE
Eliminates the identified system from processing until the RT/SMF RESTORE command is issued from the Command Processor. After an SMF time is processed, the system cannot go back to process SMF data prior to that time. When RT/SMF is restarted, the removed system must be deleted or commented out from the USERMAC(SMFPARMS) data set for RT/SMF to ignore that system.

Real-Time Dispatcher subtasks

The RT/SMF Dispatcher reads extracted SMF data and passes select records to appropriate subtasks for the SMF record types defined in the started task member USERMAC(VFISTASK).

For efficiency, the RT/SMF Dispatcher reads each record once and passes it to the appropriate subtask.

```
*
* THESE SHOULD NOT BE MODIFIED EXCEPT BY TERACLOUD S.A. AND IBM CORP.
*
* IZBR REAL-TIME DISPATCHER SUB-TASK STARTUP PARAMETERS
*
START TASK=CMDPROC,PGM=CKKRC004                                 < ALWAYS USED
*
START TASK=BRNVANLY1,PGM=BRNVC001,SMF=(14,15,30,61,64,65)   < ANALYSIS
START TASK=SMFCOUNT,PGM=CKKRT999,SMF=(0:127)                < SMF COUNTER
START TASK=IZBR/RTB,PGM=BRNVC007V,SMF=(15,60)               < BKUPTBL
START TASK=RT/HNDLR,PGM=CKKRC033                            < HEART BEAT
*START TASK=IZBR/CKK,PGM=BRNVC008,SMF=(30)                   < RT/SCHED
*START TASK=IZBR/VRS,PGM=BRNVC011                            < RECOVRY SIM
*START TASK=SCANSMF,PGM=BRNVC031,SMF=(14,15,30,64,202)       < SCANSMF
*START TASK=INVMAINT,PGM=BRNVC059                            < RT/INVMAINT
*START TASK=RT/ICP,PGM=CKKRC036,SMF=(30)                     < RT/ICP
```

Figure 33. VFISTASK member

Note: The subtasks defined in VFISTASK vary based on your installation requirements. An asterisk (*) in column 1 indicates a comment; the active entry must begin in column 1.

The following are the commonly used R/T SMF Dispatcher subtasks:

Table 14. R/T SMF Dispatcher subtasks

Name	Description	Program	Subtask name
Collection and analysis	Updates the Inventory database History/Audit database records.	BRNVC001	IZBR/RTA
BKUPTBL	Updates the date and time entry in the Backup Table if a previously backed-up data set in the Backup Table has changed.	BRNVC007	IZBR/RTB
Exception handler	Terminates the RT/SMF Dispatcher if a subtask is non-responsive for an extended period of time.	CKKRC033	RT/HNDLR
Scheduler	Updates eligibility information in the Inventory database for Backup Dynamic Data Set Scheduling.	BRNVC008	IZBR/CKK

Table 14. R/T SMF Dispatcher subtasks (continued)			
Name	**Description**	**Program**	**Subtask name**
Recovery simulator	Simulates a disaster recovery (DR) test to verify that all required data sets are in the Backup Table.	BRNVC011	IZBR/VRS
SCANSMF	Scans non-IBM Z Batch Resiliency backups and updates the Backup Table.	BRNVC031	SCANSMF
SMF counter	Shows the SMF record types and their counts processed at each checkpoint.	CKKRT999	SMFCOUNT

Analysis subtask

The collection and analysis subtask IZBR/RTA performs the following functions:

- Updates Base and Component records with information gathered from SMF records.
- Adds SMF reformatted SMF records to the History and Audit databases at SMF interval times.
- Invokes the analysis process (program BRNVC006) when a selected application job is identified.

Note: Under normal conditions, a return code of 0 is expected.

The collection and analysis subtask processes the following SMF record types:

Type 14
Close for input data sets

Type 15
Close for output data sets

Type 30
Step and job initiation and termination

Type 61
ICF define activity (ICF catalogs)

Type 64
VSAM component or cluster status

Type 65
VSAM delete activity

Backup Table subtask

The Backup Table subtask IZBR/RTB updates dates and times in the Backup Table. If a previously backed-up data set in the Backup Table has changed, the BKUPTBL date and time entry is updated.

The Backup Table subtask processes the following SMF record types:

Type 15
Close for output data sets

Type 60
VSAM volume data set updates

Exception Handler subtask

The R/T Exception Handler subtask monitors the health of the real-time analysis (RTA) environment by evaluating the execution of the other subtasks. If a subtask is unresponsive and remains in a running state for an excessive amount of time, the RTA environment is terminated.

RT Exception Handler subtask 'wakes up' every 10 minutes, indicated by the following message:

```
CKKRC03303I - MM/DD/YY HH:MM:SS - CKKRC033 - RT/HRT: WAKE UP
```

The R/T Exception Handler subtask evaluates the status of the defined subtasks. If a task is identified as still running, it is identified in a report to RTDPRINT. If, after two additional cycles, the same task is still 'RUNNING,' an error code is set and RTA is terminated.

To confirm the feature is active, ensure the subtask entry is uncommented in USERMAC(VFISTASK):

```
START TASK=RT/HRT,PGM=CKKRC033        < RTA MONITOR
```

Backup Dynamic Data Set Scheduling subtask

The Backup Dynamic Data Set Scheduling subtask IZBR/CKK works dynamically with the Inventory database for backups.

To activate this subtask, uncomment the statement in USERMAC(VFISTASK):

```
*START TASK=IZBR/CKK,PGM=BRNVC008,SMF=(30)                < RT/SCHED
```

This subtask processes the following SMF record type:

Type 30
Step and job initiation and termination

Virtual Recovery Simulator subtask

The Virtual Recovery Simulator subtask IZBR/VRS simulates the recovery as the applications are running in production.

To activate this subtask, uncomment this statement in (USERMAC)VFISTASK:

```
*START TASK=IZBR/VRS,PGM=BRNVC011                          < RECOVRY SIM
```

There are no SMF record types associated with this subtask.

SCANSMF subtask

The SCANSMF subtask scans for non-IBM Z Batch Resiliency backups such as IBM IDCAMS, IEBGENER, and IEBCOPY and updates the Backup Table.

To activate this subtask, uncomment this statement in USERMAC(VFISTASK):

```
*START TASK=SCANSMF,PGM=BRNVC031,SMF=(14,15,30,64,202)     < SCANSMF
```

The SCANSMF subtask processes the following SMF record types:

Type 14
Close for input data sets

Type 15
Close for output data sets

Type 30
Step and job initiation and termination

Type 64
VSAM component or cluster status

Type 20
IZBR DSS input record

For additional information about SMF record types and collection of SMF data, refer to IBM *System Management Facilities (SMF)*, GC28-1457 (for MVS/ESA) or SA22-7630 (for z/OS).

Command Processor subtask

You can use the Command Processor to communicate with the RT/SMF process. The Command Processor is activated in the VFISTASK member by this specification:

```
START TASK=CMDPROC,PGM=CKKRC004
```

A sample JCL stream that interfaces with the Command Processor is in the CNTL(RTCMDB) member:

```
//RTCMD         EXEC PGM=CKKRC008,
//                   PARM='DISPLAY SUBTASKS'
//*                  PARM='INVMSTRT INVMAINT'
//*                  PARM='INVMSTRT INVREMOV'
//*                  PARM='REFRESH APPTABLE'
//*                  PARM='REFRESH EXCLDSN'
//*
//STEPLIB       DD DISP=SHR,DSN=HLQ.SCKKLOAD
//*
//CMDQUEUE      DD DISP=SHR,DSN=HLQ.RTCMDQ
//*HOSTID       DD DISP=SHR,DSN=HLQ.CNTL(HOSTID)
//RTDPRINT      DD SYSOUT=*
//SNAP          DD SYSOUT=*
//ERRLOG        DD SYSOUT=*
//SYSPRINT      DD SYSOUT=*
//SYSABEND      DD SYSOUT=*
```

A sample TSO EXEC that interfaces with the Command Processor is in CNTL(RTCMD):

```
"ALLOC     F(CMDQUEUE) DA('HLQ.RTCMDQ') SHR REU"
"CALL      'HLQ.SCKKLOAD(CKKRC006)'"
"FREE      F(CMDQUEUE)"
```

Submit the CNTL(RTCMDB) member to issue commands to the Command Processor.

The following commands can be supplied in the PARM field of the RTCMDB job.

Table 15. RTCMDB job parameters

Command	Description
REFRESH xxxxxxx	Refreshes the table specified where xxxxxxxx is the table name. For example, REFRESH APPTABLE refreshes the Application Table.
STOP	Initiates the RT/SMF termination process. The facility can also be stopped via console 'P' command issued for the RT/SMF processor.
STARTVRS	Starts the Virtual Recovery Simulator Process. The process begins at the next RT/SMF checkpoint and continues for 24 hours of processing or until a STOPVRS request is issued. See Chapter 8, “Virtual Recovery Simulator,” on page 131 for more information.
STOPVRS	Halts the Virtual Recovery Simulator process.
DISPLAY SUBTASK	Displays RT/SMF subtasks.
DISPLAY SUBTASK=IZBR/RTB	Displays subtask information for the specified subtask. You can specify either the program (BRNVC007) or the subtask name (IZBR/RTB). Both come from the VFISTASK table.
DISPLAY OATS	Displays the Open Application Table entries and status.
DISPLAY CPU	Displays CPU usage for RT/SMF.
DISPLAY STORAGE	Displays the storage in use by RT/SMF. You can request information for the entire address space or for a particular subtask/program.
DISPLAY STORAGE=BRNVC007	
DISPLAY TABLES	Displays information regarding the system tables; e. g. APPTABLE. The display lists the table, program loader, file type, status, load time, PDS member (if applicable), and file number.
DISPLAY SMF	Displays the status of processing of each SMF data set for each system defined including SMF$ data sets, if applicable. It also displays the last record date/time.
DIAGNOSE xxxxxxxx	Activates diagnostics for subtask xxxxxxxx.
DIAGNOSE xxxxxxxx.RSC=yyy	Activates diagnostics for resource yyy within subtask xxxxxxxx.
DIAGNOSE xxxxxxxx,OFF	Deactivates diagnostics for subtask xxxxxxxx.

A sample DISPLAY SUBTASKS report follows. The report is directed to the RTDPRINT DD statement in the COLLECT or RTSMF JCL and is prefixed by text identifying the issuing program (BRNVC014 in these cases) and a date/time stamp in the form: BRNVC01400I - *mm/dd/yyhh:mm:ss*.

```
 UID: xxxx    CMD: DISPLAY                                      COMMAND: DISPLAY
PROGRAM: BRNVC014  ATTACHED, TCB: 008BD858  IZBR RT/SMF:SUBTASK REPORT
EYECATCHER: RTSRTDSB                                          SUB-POOL/LENGTH: 020003A0
ADDRESS: 00017C60
MAX SUBTASKS:           9   ACT SUBTASKS:   5

TASKNAME  PROGRAM   TCB ADDR
---------  --------  --------
SMFALLOC  CKKRC030  008BDB40
CMDPROC   CKKRC004  008CE960
BRNVANLY1 BRNVC001  008BD1E0
SMFCOUNT  CKKRT999  008CE438
IZBR/RTB  BRNVC007  008CE1A0

IZBR RT/SMF:SUBTASK REPORT COMPLETE
COMMAND: DISPLAY  TERMINATED - COMPLETION CODE: 00000000
```

Figure 34. Sample DISPLAY SUBTASKS report

Refreshing IBM Z Batch Resiliency tables

The CNTL(RTCMDB) member is used to refresh IBM Z Batch Resiliency tables such as the APPTABLE and LEGTABLE.

About this task

Run RTCMDB in these cases:

- Inventory maintenance is running in real time and RT/SMF remains active for a period of time. Details on Inventory maintenance are contained in IBM Z Batch Resiliency Maintenance.
- APPTABLE is updated manually, and you want to refresh the APPTABLE without waiting for Real-Time processing to be restarted.

Procedure

1. Uncomment the PARM='REFRESH APPTABLE' statement in RTCMDB, as shown in the following example:

```
     HLQ.CNTL(RTCMDB)
//RTCMD  EXEC PGM=CKKRC008,
//            PARM='REFRESH APPTABLE'
```

2. Specify the system where the Real-Time dispatcher is running. If the dispatcher is on a different system, specify the system ID in the file pointed to by DDNAME HOSTID.

Definition of test or alternate configurations for parameter sets (VFIRTD00)

The parameter member USERMAC(VFIRTD00), which is used by the RTSMF job, allows you to define test configurations or alternate configurations for the various parameter sets without modifying the original parameter sets.

VFIRTD00 identifies the SMFPARMS, VFIPARMS, VFISTASK, and VFITABLS in the USERMAC library, which are required to process the SMF data.

```
*  IZBR REAL-TIME SMF STARTUP PARAMETER LIST
*
* NOTE THAT 2 (OR MORE) SPACES DELIMIT KEYWORD/OPERAND FROM COMMENTS
*
*
SMFPARMS = SMFPARMS
VFIPARMS = VFIPARMS
CKKRTASK = VFISTASK
CKKRTABL = VFITABLS
```

Figure 35. VFIRTD00 Real-Time Startup parameters

SMF parameters (SMFPARMS)

The SMFPARMS options are distributed in member USERMAC(SMFPARMS). The parameters define the patterns of the extract files that are read by the RT/SMF Dispatcher.

One control statement is required for each system being processed.

```
*
* SMF PARAMETERS
*
SYSID=xxxx,SCRATCH=YES,SMFDSN=smf$.extract.dataset.hlq{,ALTSUFX=YES|NO}
*
```

The following table describes the SMFPARMS parameters:

Table 16. SMFPARMS parameters

Keyword	Description
SYSID	Identifies the system for the SMF extraction data sets. The system ID must match the SMF Extractor SYSID in CNTL(SFMPxxxx).
SCRATCH	Specifies if SMF$ data sets should be scratched. • YES – scratches the SMF$ data sets after processing. • NO – does not scratch the data sets after processing.
SMFDSN	Defines the high-level qualifier for the SMF$ data sets. The high-level qualifier must match the OUTDSN/OUTLGR parameter supplied to the SMF Extractor in CNTL(SMFPxxxx). The maximum length is 28 characters.

ALTSUFX	This is an OPTIONAL feature.
	If ALTSUFX=YES is specified, an alternate suffix is expected for the user-generated SMF extraction files. That is, the user is not employing our automatic SMF Extractor process. This is a possible option for very large installations to generate their own SMF extraction files that require granularity among the files down to the second to allow the possibility of two LPARs switching SMF recording files within the same minute.
	The suffix has the format:
	`$MF%%%%%%.T%%%%%%`
	where the **$MF** and the **T.** must be present in their indicated positions and the percent signs may represent any character.
	Using the parameters:
	`SYSID=LPR1, SMFDSN=SYS2.LPR1.SMFEXT, SCRATCH=YES,ALTSUFX=YES`
	an acceptable SMF extraction file name would be:
	`SYS2.LPR.SMFEXT.$MF18121.T092033`
	where 18122 represents the 121st day of 2018 and 092033 represents the time 9:20:33 AM
	If this parameter is used, the maximum length of the SMFDSN parameter is 27 rather than 28.
	The standard IBM Z Batch Resiliency naming format remains as the preferred SMF extraction file naming format.

RT/SMF Dispatcher DD statements

The following table describes the DD statements in RTSMFP:

Table 17. RTSMFP DD statements

DD name	Description
PARMLIB	Defines the PARM library for this execution (default USERMAC).
INVENTRY	Defines the Inventory database.
INVENTRZ	Secondary DDNAME for the Inventory database, required by the Backup Dynamic Data Set Scheduler.
CMDQUEUE	Defines the Command Queue database. Commands issued to RT/SMF are stored here pending processing.
APPTABLE	IBM Z Batch Resiliency application table.
APPTABLX	Temporary work data set for the APPTABLE.
BKUPTBL	Backup Table, required for IZBR/RTB.
AUDTDS	SMF interval Audit data set.
PRODDS	SMF interval History data set.
CHECKPT	RT/SMF Checkpoint data set. This data set contains information regarding which SMF$ data sets were processed and the latest SMF record time.
ODTCKPT	Open Application/data set checkpoint file. This file is updated at the end of an SMF$ data set and at 10-minute SMF intervals. It contains the Open Application Table and Open Data Set Table Entries at the time of the checkpoint. It is used on an RT/SMF warm start.
SCHDJDS	Backup Dynamic Data Set Scheduling journal work file.
RTDPRINT	RT/SMF messages file.
SNAP	Diagnostic information.

Real-Time critical file analysis

The Real-Time collection and analysis subtask creates and updates the Inventory database and identifies which data sets within the inventory are critical to recovery.

IBM Z Batch Resiliency defines a critical file as a data set that is required to restart an application from a given synchronization point; a data set that is absolutely vital in the context of performing an application recovery.

Critical file status

The critical file status, defined in the Inventory database for each data set, has one of the following values:

- The file is critical.
- The file is an exception file.
- The file is deleted on its first access and is not critical.
- The file is created, opened for output or defined on its first access and is not critical.

Critical file analysis based on recovery strategies

The Critical File List varies depending on what type of recovery you use and from what point you are recovering. IBM Z Batch Resiliency provides application recovery and global recovery strategy types. Both types are addressed by the IBM Z Batch Resiliency RECOVERY parameter contained in member USERMAC(VFIPARMS).

Application recovery

With application recovery, you can view applications to determine critical file requirements and recover them on an individual application basis. This can be useful in disaster recovery testing, as there is often not enough time to execute all applications at the test site, or when component testing of applications is being performed.

Furthermore, if the disaster recovery plan requires restoring applications in a sequence that is different from the normal flow of execution, a priority-based application recovery is created. With priority-based recovery, the order in which applications are recovered is based on the importance of an application to the survival of the business, rather than on the order in which applications are actually executed on a normal day.

If you choose application recovery, each application is analyzed separately and the files that must exist to begin each application process are considered critical to that application. Each application has its own set of critical files, its own set of backups for those files, and its own backup verification procedures for those backups. Priority-based application recovery is possible if:

1. The application does not contain cross-application files.
2. Cross-application files are backed up at the last point of update or at the end of the application that last updated the file which allows the files to then be recovered before any of the applications that use them.

Global recovery

With the global recovery strategy, applications cannot be recovered on an individual basis, as the prerequisite processing that creates files critical to subsequent applications has not yet taken place.

For example, if applications one and two are being viewed as one large application, a global recovery is being performed. In this type of recovery, the transaction file is not considered critical, since it is created when application one executes. The total number of critical files is then reduced to:

- The online database from application one
- The updated output file from application two

In example two, the prerequisite processing for application two would be the creation of the transaction file.

If you choose global recovery, only those files that are needed at the beginning of all the applications are considered critical.

Using global recovery, the Tape Management System Vault Job cannot be run more than once per day. To gain optimal recovery, synchronize the TMS Vault job, catalog backup, and the BKUPTBL to run at the same time.

Newly found critical data sets

The analysis process creates and updates the Inventory database and also identifies which data sets within the inventory are critical to recovery.

IBM Z Batch Resiliency reporting by application produces information that allows identification of cross-application interface files. This information is relevant in planning and testing the recovery of a specific application, as the cross-application files are needed at the recovery site for successful recovery of the downstream dependent application.

When reporting on a global basis, IBM Z Batch Resiliency reports on files that must be present for the recovery of all defined applications. The global perspective assumes that all applications are restored,

and that recovery begins from the first application that runs in the daily cycle. Cross-application files are not reported, as recovery is viewed as one large application.

Known critical data sets identified as not critical

All newly identified critical files, with the exception of those entered in the exclusion list, are marked in the Inventory database as critical by the analysis process.

If a known critical file is not identified as critical, possible reasons include:

- The file is already identified as critical and is included in the Critical File List or the exclusion table.
- The file was not opened during the currently available SMF data time period.
- The file was not opened by a job listed in the application table.
- An SMF record is not being captured for that file.

To verify whether an SMF record exists for the file, use the CNTL(JOBRPT) report program to extract SMF records from the History file.

Exception file reporting

Exception file reporting is provided after IBM Z Batch Resiliency analysis processing. Exception files are defined as files opened as output with a disposition of OLD or MOD. Examples include on-line reports and log files. The use of these files determines whether they need to be backed up. For example, if the file has a disposition of OLD and is over-written each day during the daily cycle from another input source, the data does not need to be backed-up (the input file that creates this file is critical). In this case, the information contained in the file is noncritical; only the fact that the file must exist makes it critical.

During recovery, as during normal processing, you will encounter JCL errors if an application attempts to open a file as DISP=OLD when that file does not exist.

If you choose not to back up the file, you must make provisions in your disaster recovery plan so that a dummy file exists before the application runs. An alternative approach would be to include a prerequisite step to the job that allocates all files of this type if they do not exist. In order to accomplish this, you may change the JCL during the first open from DISP=OLD to DISP=MOD and include SPACE and DCB parameters for the file (the file is created if it does not exist).

If the data set uses non-sequential file access (direct access, SAS file, and so on), and is not created from scratch with each daily run, then the file may be critical. Any files in the exception report that research indicates to be critical should be added to the Critical File List via the IBM Z Batch Resiliency ISPF interface.

Exception files that have been verified as being non-critical should be added to the data set exclusion list by entering them in member EXCLDSN in the USERMAC library. Exception files can be flagged as critical files on an automatic basis by setting the exception file switch to CRITSW=1. CRITSW is found in member VFIPARMS of the USERMAC library.

The advantage of automatically marking exception files as critical is that the application is protected while research is underway to determine actual file status. CRITSW should be set to a value of zero (0) if automatic flagging of exception files is not desired.

Chapter 6. Generating IBM Z Batch Resiliency Backup JCL

The BKUPJCL process generates the daily JCL or control cards for backing up critical data sets. After the critical data sets are successfully backed up, the backup table is updated.

Backup dynamic data set scheduling is an optional feature that schedules backups of critical data sets to be run at intervals throughout the day. You can define which applications and data sets are eligible for scheduling and how to schedule them as described in “Backup Dynamic Data set Scheduling” on page 97.

Backup cycle

The following image shows a typical IBM Z Batch Resiliency backup cycle that generates JCL to back up critical data sets.

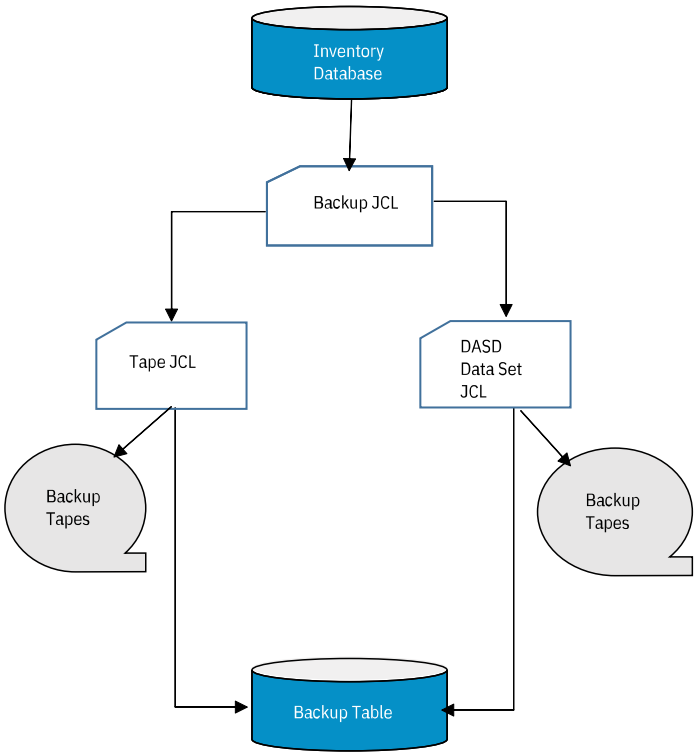


Figure 36. Backup cycle flowchart

Backing up critical application data sets

BKUPJCL generates the JCL to back up critical data sets and control cards defined in the Inventory database and data set Include list (INCLDSN). BKUPJCL optionally generates JCL for all backup methodologies grouped by application and data set name.

You can define and track multiple backup methodologies for different backup utilities, as described in Defining backup methodologies. For example, you can use the DF/DSS utility to back up critical sequential data and use FAVER to back up VSAM data to tape.

The following figure shows a sample CNTL(BKUPJCL) program which generates the BKUPJCL member:

© Copyright IBM Corp. 2019, 2020

```
//*====================================================================
//* PROGRAM:   BACKJCL
//* FUNCTION:  CREATE CONTROL CARDS & JCL TO BACKUP
//*             CRITICAL FILES FOR MODE=CFL
//*====================================================================
//STEP1        EXEC PROCVFI,SYSIN=BRNBKJCL
//BACKJCL      DD DISP=SHR,DSN=HLQ.USERMAC(BACKJCL)
//BKUPMETH     DD DISP=SHR,DSN=HLQ.BKUPMETH
//USEREXIT     DD DISP=SHR,DSN=HLQ.USERMAC(XBKUP001)
//USEREXT2     DD DISP=SHR,DSN=HLQ.USERMAC(XBKUPJCL)
//UFIELDS      DD DISP=SHR,DSN=HLQ.USERMAC(UFBKPJCL)
//*--------------------------------------------------------------------
//* COPY REPORT(S) TO ONLINE LIBRARY
//*--------------------------------------------------------------------
//COPYRPT      EXEC COPYPRNT,COND=EVEN,MEMBER=BKUPJCL
**********************************************************************
```

Figure 37. BACKJCL program

BACKJCL parameters

The following table describes the USERMAC(BACKJCL) parameters. Some backup methodologies can differ in their interpretation of the table below and are described in Defining backup methodologies. The default values are underlined.

Table 18. BACKJCL parameters

Option	Parameter	Description
GDG	LISTALL	Back up all generations of GDG files, appends a period and asterisk (.*) to the base name when processed.
	LISTCUR	Like CURRENT, back up only the most current critical file but Catalog Services is used to determine the actual generation number (for example, G0901V00) for the file.
	LISTTAB	Like TABLE2, but Catalog Services is invoked to determine the generation numbers of all critical files as listed in the GDGCOUNT field of the Critical File List.
INCLUDES	YES	Generate backups for files defined in the USERMAC(INCLDSN) member.
	NO	Do not generate backups for files specified in the USERMAC(INCLDSN) member.
BKUPTYPE	ALL	Generate backups for all critical files.
	UPDATES	Generate backups only for critical files that either do not already have a backup or have been modified since its last backup.
CROSSAPP	YES	Generate a pseudo-application named CROSSAPP for files belonging to more than one application.
	NO	Back up files that belong to more than one application in each application.

BKUPJCL Exits

BKUPJCL uses two exit points: XBKUP001, which forces an early backup, and XBKUPJCL, which redirects backup methodologies.

Forcing an early backup

The USERMAC(XBKUP001) exit is the first exit point invoked by BKUPJCL after the Backup Table is read. This exit point can be used to force early backups of records that are about to expire.

Redirecting or changing backup methodologies

The USERMAC(XBKUPJCL) exit is the second exit point invoked after the complete list of critical files have been evaluated and prior to the removal of any files when BKUPTYPE=UPDATES is specified.

You can use the XBKUPJCL exit to:

- Redirect backup methodologies based on migration status
- Change backup methodologies based on the volume on which they reside
- Remove or combine applications, to generate JCL for specific applications

For example, a file may be listed in the Critical File list as a DASD file however, the most current version may be on tape, possibly because of a shortage of production DASD space. If your backup methodology is DF/DSS, this file would not be backed up. In addition, the typical incremental backup process that is run would not have a copy of this file. A simple line of code would redirect this file to be backed up with the IBM Z Batch Resiliency tape copy process and therefore a copy of this critical file would be made.

Defining backup methodologies

The BKUPMETH data set contains a table of backup methodologies and the name of the files that JCL is written to for both backup and restore operations. When IBM Z Batch Resiliency is installed, BRNVCPY and DASD (for DSS, FDR, or DMS) backup methodologies are defined.

About this task

You can add a backup methodology in two ways:

Procedure

- Edit BKUPMETH. Remove the asterisk (*) from the associated backup method entry and update the BKUPJCL and RESTJCL.
- Use the ISPF Backup Methodology Panel (via Option 5 MAINTENANCE then Option 9 BKUPMETH) to remove the asterisk (*) from the associated backup method entry.

Results

The backup methodology name cannot be longer than 7 characters. The files listed are the PDS files of the JCL generated by these methodologies.

Adding data sets to the Include list

You can use the Include list USERMAC(INCLDSN) member to manually direct data sets that are not critical in the IBM Z Batch Resiliency backup process. By using the BKUPJCL option INCLUDES=YES, the BKUPJCL process combines this input with the established critical files list. The advantage of using the Include list is its masking capability. The Include list can be a masked input to allow similar files with the same recovery requirements to be backed up and restored automatically.

The masking of data sets supports DF/DSS standard conventions using the percent sign (%) as a place holding character and an asterisk (*) to represent a series of characters within a node. The asterisk string (**) can be used to replace one or more nodes.

You can also exclude data sets from the INCLDSN process by masking the DSNAME parameter. To mask the DSNAME, add a colon (:) at the end of the INCLDSN control card followed by the masked DSNAME to be excluded.

To add multiple exclude masks, follow each exclude mask with a comma. Place a hyphen (-) at least one space after the last comma to continue the exclude masks on the next line.

Note: All control cards must start in column one. The high-level qualifier in the inclusion list cannot be masked. Each line must follow the Include list record syntax with no spaces between operands.

Include list record syntax

```
APPLID,DSNAME,[BKPMETH],[DAYSFROMCREATION],[DATATYPES]
```

Table 19. Include list record syntax

Parameter	Description
APPLID	Specifies the APPLID to be assigned to the data set.
DSNAME	The data set or masked data set to be backed up.
BKPMETH	The optional backup methodology to be used for this data set or group of data sets. Any valid backup method defined in the BKUPMETH file. If not specified, the default is Global DMS specified in USERMAC(VFIPARMS).
DAYSFROMCREATION	This optional parameter refers to including only those data sets that have been created within the past X number of days. The default value is for all data sets matching DSNAME mask. A value of 'N' allows the data set to be passed to the backup methodology without checking to see if the data set is cataloged; however, masking is only permitted for the DF/DSS methodology.

DATATYPES	This optional parameter specifies cataloged data set data types to process. Types include: VSAM, NONVSAM, Alternate Index, Alias, and Generation Data Groups with or without specific number of relative generations for the DSNAME. Options are separated by semicolons, or else ignored.
	`DATATYPE(S): G###;C;N;A;X`
	`G### = GDG - ### OCCURRENCES, MAX=255`
	`G    = GDG - ALL OCCURRENCES, MAX=255`
	`###  = GDG - ### OCCURRENCES, MAX=255`
	`C    = VSAM`
	`N    = NONVSAM`
	`A    = ALIAS`
	`X    = ALTERNATE INDEX`
	Note:
	1. Can choose any combination of the above. This limits DSN processed to datatype selected.
	2. Specifying none of the above defaults to all: G###;C;N;A;X
	3. Can choose numeric value only to get all G### only.
	4. Alias datatypes are not used for backup.
	5. Alternate Index datatypes are not used for backup.

Testing INCLDSN masking

Because masking rules can be complex, and it is not always clear exactly which data sets are selected for a given mask; you can use the CNTL(BRNVCAT8) utility to test masking results before using an INCLDSN member in production. See BRNVCAT8 – INCLDSN testing utility for more information.

INCLDSN examples

In this example, all files that match the pattern ABC.*.D* (for example, ABC.MASTER.D10230) is assigned to application PRODAP01 as long as the creation date of the file is within 5 days. The backup methodology used is DF/DSS.

```
PRODAP01,ABC.*.D*,DSS,5
```

In these equivalent examples, most files that match the pattern ABC.** is assigned to application PRODAP02. Excluded files are GDG bases, and VSAM data and index components. Backup methodology used is DF/DSS.

```
PRODAP02,ABC.**,DSS

PRODAP02,ABC.**,DSS,,G;N;C;A;X
```

In these equivalent examples, up to two GDG data sets for each GDG base matching the pattern ABC.** is assigned to application PRODAP03. GDG groups are listed from newest to oldest. Backup methodology used is DF/DSS.

```
PRODAP03,ABC.**,DSS,,2

PRODAP03,ABC.**,DSS,,G2
```

Note: Requested GDG dsname mask, using GDG datatype, should NOT use G%%%%V%% as last node.

Example of assigned data set names for these examples:

```
PRODAP03   ABC.MASTER.BKUPAUD.FLATFILE.G0015V00
PRODAP03   ABC.MASTER.BKUPAUD.FLATFILE.G0014V00
PRODAP03   ABC.MASTER.BKUPTBL.FLATFILE.G0015V00
PRODAP03   ABC.MASTER.BKUPTBL.FLATFILE.G0014V00
```

In this example, all alternate indexes matching the pattern ABC.*.INVENTRY.** is assigned to application PRODAP04. Backup methodology used is DF/DSS.

```
PRODAP04,ABC.INVENTRY.**,DSS,,X
```

Note: Any of the datatypes can be requested in the same way. Alternate Index data types cannot be used for backup.

Example of assigned data set names for this example:

```
                                                    Datatype
PRODAP04   ABC.PROD0001.INVENTRY.AIX                   X
PRODAP04   ABC.PROD0002.INVENTRY.AIX                   X
```

In this example, all files that match the pattern ABC.** of datatypes (N)onvsam, (C)Vsam, (A)lias and (X)Alternate index, is assigned to application PRODAP05. This excludes all GDGs. Backup methodology used is DF/DSS.

```
PRODAP05,ABC.**,DSS,,N;C;A;X
```

In these equivalent examples, all GDG data sets that match the pattern above is assigned to application PRODAP06. GDG groups are listed alphabetically from oldest to newest. Backup methodology used is DF/DSS.

```
PRODAP06,ABC.**,DSS,,G

PRODAP06,ABC.**,DSS,,G255

PRODAP06,ABC.**.G%%%%V%%,DSS
```

Note: The last example works because the GDG datatype(,,G) is not used.

Example of assigned data set names:

```
PRODAP06   ABC.MASTER.BKUPAUD.FLATFILE.G0012V00
PRODAP06   ABC.MASTER.BKUPAUD.FLATFILE.G0013V00
PRODAP06   ABC.MASTER.BKUPAUD.FLATFILE.G0014V00
PRODAP06   ABC.MASTER.BKUPAUD.FLATFILE.G0015V00
PRODAP06   ABC.MASTER.BKUPTBL.FLATFILE.G0012V00
PRODAP06   ABC.MASTER.BKUPTBL.FLATFILE.G0013V00
PRODAP06   ABC.MASTER.BKUPTBL.FLATFILE.G0014V00
PRODAP06   ABC.MASTER.BKUPTBL.FLATFILE.G0015V00
```

This first example includes all files that match the pattern ABC.ZZZZ.** and assign to application PRODAP07. However, files ABC.ZZZZ.TEST.** is excluded:

```
PRODAP07,ABC.ZZZZ.**,DSS:ABC.ZZZZ.TEST.**
```

The second example includes all files that match the pattern ABC.XXXX.** and assign to application PRODAP07. However, files ABC.XXXX.TEST.** and ABC.XXXX.SUMMARY.**and ABC.XXXX.HISTORY.** is excluded:

```
PRODAP07,ABC.XXXX.**,FDR:ABC.XXXX.TEST.**,ABC.XXXX.SUMMARY.**, -
ABC.XXXX.HISTORY.**
```

Deleting PDS members (VFIMEMDL)

The VFIMEMDL JCL is an IBM Z Batch Resiliency utility that deletes control card members from a PDS library. Add the VFIMEMDL JCL to CNTL(BKUPJCL) as a first step to delete all existing members. The JCL resides in CNTL(VFIMEMDL).

Examples of VFIMEMDL JCL in CNTL(VFIMEMDL) using various options follow:

```
//*============================================================
//* VFIMEMDL - DELETE ALL CONTROL CARD MEMBERS IN A PDS
//* LIBRARY.
//*============================================================
//VFIMEMDL EXEC PGM=IKJEFT01,REGION=0M,PARM=(VFIMEMDL)
//SYSEXEC  DD  DISP=SHR,DSN=HLQ.SBRNSRCE
//BKUPCNTL DD  DISP=SHR,DSN=HLQ.RESTJCL.DSS
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY

//*------------------------------------------------------------
//* JS010: DELETE MEMBER DELX304 IN LIBRARY POINTED BY BKUPCNTL
//*         HIGHEST EXPECTED RETURN CODE:  04
//*------------------------------------------------------------
//JS010    EXEC PGM=IKJEFT01,PARM=(VFIMEMDL,'BKUPCNTL DELX304')
//STEPLIB  DD  DISP=SHR,DSN=HLQ.LIBRARY
//SYSEXEC  DD  DISP=SHR,DSN=HLQ.SBRNSRCE
//BKUPCNTL DD  DISP=SHR,DSN=HLQ.RESTJCL.DSS
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY

//*------------------------------------------------------------
//* JS010: DELETE $DELNC* MEMBERS IN LIBRARY POINTED BY BKUPCNTL
//*         HIGHEST EXPECTED RETURN CODE:  04
//*------------------------------------------------------------
//JS010    EXEC PGM=IKJEFT01,PARM=(VFIMEMDL,'BKUPCNTL $DELNC*')
//STEPLIB  DD  DISP=SHR,DSN=HLQ.LIBRARY
//SYSEXEC  DD  DISP=SHR,DSN=HLQ.SBRNSRCE
//BKUPCNTL DD  DISP=SHR,DSN=HLQ.RESTJCL.DSS
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
```

Figure 38. VFIMEMDL JCL

If DDNAME is not passed as a parameter, the default is DDNAME BKUPCNTL. If a member is passed, then only that member is deleted. Limited masking is supported for the member parameter. An asterisk (*) is allowed as the last character for member parameter, so TEST* matches any member that starts with TEST.

Checking for existence of PDS members (BRNVFMMC)

BRNVFMMC is an IBM Z Batch Resiliency utility that checks for the existence of a member before a backup job starts. A common use of BRNVFMMC is to check that the BKUPJCL BRNVCPY member exists and then use condition code checking before starting the backup. For this use, place BRNVFMMC in the XTAPECPY process prior to the TAPEPARS step. The JCL resides in CNTL(VFIMEMCK)

If DDNAME is not passed as a parameter, the default is DDNAME BKUPCNTL.

A sample of the BRNVFMMC JCL follows:

```
//*============================================================
//* BRNVFMMCK - CHECKS FOR THE EXISTENCE OF BACKUP CONTROL CARD
//* MEMBER FOR APPLID.  IF NOT FOUND RETURNS CC 3
//* FOR OTHER ERRORS, RETURNS CC 4
//*============================================================
//PS05    EXEC PGM=IKJEFT01,REGION=0M,PARM=(BRNVFMMC",'&APPLID')
//SYSEXEC  DD DISP=SHR,DSN=HLQ.SBRNSRCE
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//BKUPCNTL DD DSN=HLQ.BKUPJCL.DSS,DISP=SHR
```

Figure 39. BRNVFMMC JCL

Load balancing backups using application splitting

You can use the USERMAC(APPLDEF) member if backup processing exceeds the available batch window.

APPLDEF implements load balancing to split a specific application into a subset of more manageable-sized jobs that may potentially run in parallel. This results in an overall reduction in the wall clock time required to complete the backup of the aggregate within the allotted time period.

Sample APPLDEF member:

```
*-------------------------------------------------------------
*
* USED BY BACKJCL TO CREAT JCLS FOR APPLICATION BACKUPS
* RECORD FORMAT: (COL 1: '*' COMMENT)
* BKUPMETH,APPLID,JOBAPP,SPLITS,MAXFILES,MAXMEG
*
*-------------------------------------------------------------
DSS,APPLE,,2,5,500
DSS,CROZZAPP,XAPP,2
DSS,FISH,SHARK,,,2500
```

Figure 40. APPLDEF JCL

Backup methodology utilities

This section describes the backup methodologies that you can use with IBM Z Batch Resiliency.

Backing up Online DASD to tape (DF/DSS)

You can use DF/DSS to back up online DASD data sets to tape. The generated JCL executes a procedure that resides in the CNTL library. The members of the BKUPJCL.DSS library correspond to the APPLIDs defined to this backup methodology. The name of the output data set is generated by IBM Z Batch Resiliency and the Backup Table is updated upon successful completion of the process.

Table 20. DF/DSS information

DF/DSS information	
NAME IN BKUPMETH	DSS
Procedure in the CNTL Library (BACKUP)	XDSSBKUP
DD Statement in BKUPJCL	Dynamic allocation
BKUPJCL template in USERMAC	ADSS

The following is a sample of the JCL BKUPJCL created for DF/DSS using the ADSS template:

```
//BRNVBU01  JOB (0000,0000),'XXXX',CLASS=D,MSGCLASS=X,NOTIFY=&SYSUID
//*-------------------------------------------------------------------
//* IZBR BACKUP JOB
//* (JOBNAME ABOVE MUST BE 6 CHARACTERS OR LESS)
//* DO NOT EXCEED 20 LINES IN THIS MEMBER
//*-------------------------------------------------------------------
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL.)
//*-------------------------------------------------------------------
//JS10     EXEC XDSSBKUP,
//         APPLID=GL,
//         BDSN='IZBR.V4X.GL.D070131.T0446.DSS001'
//PS10.BRNVFILT DD *,LRECL=80
             INCLUDE(GL.HCD.MSGLOG -
                     GL.HCD.TERM -
                     GL.HCD.TRACE  )
/*
```

Figure 41. BKUPJCL for DF/DSS

The following procedure is in CNTL(XDSSBKUP):

```
//XDSSBKUP PROC APPLID=,  APPLID (REQUIRED)   LEN < OR = 8
//         SUBAPPL=,      SUBAPPLID(OPTIONAL) LEN < OR = 8
//         USERINFO=,     USERINFO(OPTIONAL)  LEN < OR = 24
//         BDSN=,
//         TUNIT=3490
//*================================================================
//* THIS STEP WILL BACK UP ALL OF THE SPECIFIED DATA SETS TO
//* SCRATCH TAPES.
//*================================================================
//PS10     EXEC PGM=ADRDSSU,REGION=0M
//*S10     EXEC PGM=ADRDSSIM,REGION=0M
//STEPLIB  DD DSN=HLQ.LIBRARY,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   DSN=&&SYSOUT,DISP=(NEW,PASS),
//         UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
//         DCB=(RECFM=VBA,DSORG=PS,LRECL=121,BLKSIZE=1214)
//TAPE1    DD   DSN=&BDSN,
//         UNIT=&TUNIT,VOL=(,RETAIN,,50),
//         DISP=(,CATLG),LABEL=(,SL,RETPD=10)
//SYSIN    DD   DSN=HLQ.USERMAC(DSSCARD),DISP=SHR
//INPARM   DD  DUMMY
//*----------------------------------------------------------------
//PS20     EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1   DD DSN=&&SYSOUT,DISP=SHR
//SYSUT2   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*----------------------------------------------------------------
//PS30     EXEC PGM=BRNVYSIN,PARM=('&BDSN')
//STEPLIB  DD DSN=HLQ.LIBRARY,DISP=SHR
//BRNVOUT  DD DSN=&&SYSIN,DISP=(,PASS),DSORG=PS,
//         UNIT=SYSDA,SPACE=(CYL,1),
//         DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//ERRLOG   DD SYSOUT=*
//*----------------------------------------------------------------
//PS40     EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=&&SYSIN,DISP=(OLD,DELETE,DELETE)
//*----------------------------------------------------------------
//PS50     EXEC PGM=BRNVBDSS,
//         PARM=('&APPLID,&SUBAPPL,&USERINFO'),
//         COND=(0,NE,PS40)
//STEPLIB  DD DSN=HLQ.LIBRARY,DISP=SHR
//TAPEHOLD DD DSN=*.PS10.TAPE1,DISP=OLD,UNIT=&TUNIT
//DSSOUT   DD DSN=&&SYSOUT,DISP=(SHR,DELETE)
//DDIN     DD DSN=&&BRNVTEMP,DISP=(NEW,DELETE),
//         UNIT=SYSDA,SPACE=(CYL,(5,5)),
//         DCB=(LRECL=1741,BLKSIZE=31338,RECFM=FB,DSORG=PS)
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ERRORS   DD SYSOUT=*
//ERRVSAM  DD SYSOUT=*
//BKUPTBL  DD DSN=HLQ.BKUPTBL,DISP=SHR,
//            AMP=('BUFNI=12,BUFND=49')
//BKUPAUD  DD DSN=HLQ.BKUPAUD,DISP=(SHR)
```

Figure 42. CNTL(XDSSBKUP)

Backing up online or migrated DASD and tape data sets to tape (ABARS, ABARACC, ABARALL)

You can use DFSMShsm ABARS to back up online DASD, migrated DASD and tape data sets to tape.

ABARS is a started task that is initiated by the HSM command. The BKUPJCL members created are ABARS control statements that are invoked by the aggregate previously defined through ISMF. See Defining aggregate groups for more information.

Aggregate backup and recovery support (ABARS) provides the following functions:

- Validates the ABARS aggregate lists prior to DFSMShsm submission so that ineligible files are not processed
- Scans ABARS activity logs to update the Backup Table which allows for a simpler restore from an ABARS aggregate backup using the ISPF interface.

Three backup methodologies correspond to ABARS:

- **ABARS** – for data sets to be backed up
- **ABARACC** – for tape files to accompany an aggregate backup
- **ABARALL** – for disk files to be allocated but not restored by the aggregate recovery

The members of the BKUPJCL.ABARS library correspond to the APPLIDs designated for the aggregate.

Table 21. ABARS information

ABARS information	
NAME IN BKUPMETH	ABARS
Procedure in the CNTL Library (BACKUP)	XABARBKP
DD Statement in BKUPJCL	Dynamic allocation
BKUPJCL template in USERMAC	AABARS

Defining aggregate groups

Before you can begin using the JCL created by BKUPJCL with ABARS, define an aggregate group for each application defined in the APPTABLE.

About this task

For more information on defining aggregate groups and using ABARS for disaster recovery, see "Aggregate Backup and Recovery Support (ABARS)" in IBM® publication SH21-1076 Storage Administration Guide for DFSMS.

Procedure

1. To define aggregate groups, select option 9, AGGREGATE GROUP, on the ISPF Primary Option menunfor z/OS DFSMS.

```
                ISPF PRIMARY OPTION MENU - z/OS DFSMS
Enter Selection or Command ===>

Select one of the following options and press Enter:

0  ISPF Profile            - Change ISPF User Profile
1  Data Set                - Perform Functions Against Data Sets
2  Volume                  - Perform Functions Against Volumes
3  Management Class        - Specify Data Set Backup and Migration Criteria
4  Data Class              - Specify Data Set Allocation Parameters
5  Storage Class           - Specify Data Set Performance and Availability
9  Aggregate Group         - Specify Data Set Recovery Parameters
L  List                    - Perform Functions Against Saved ISMF Lists
R  Removable Media Manager - Perform Functions Against Removable Media
X  Exit                    - Terminate ISMF



Use HELP Command for Help; Use END Command to Exit.
```

Figure 43. ISPF Primary Option Menu for z/OS DFSMS

The Aggregate Group Application Selection panel, shown in the following figure, is displayed.

```
Panel  Utilities  Help
 ------------------------------------------------------------------------------
                      AGGREGATE GROUP APPLICATION SELECTION
 Command ===>

 To Perform Aggregate Group Operations, Specify:
   CDS Name . . . . . . . . 'SMS.ZOS1.SCDS'
                                (1 to 44 Character Data Set Name or 'Active')
   Aggregate Group Name . . GL        (for Aggregate Group List, fully or
                                      Partially Specified or * for All)
 Select one of the following Options:
   3  1. List     - Generate a list of Aggregate Groups
      2. Display  - Display an Aggregate Group
      3. Define   - Define an Aggregate Group
      4. Alter    - Alter an Aggregate Group
      5. Abackup  - Backup an Aggregate Group
      6. Arecover - Recover an Aggregate Group

 If List Option is Chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 44. Aggregate Group Application Selection panel

2. On the Aggregate Group Application Selection panel, complete the following:
   a) Enter the name of the SCDS (source control data set) in the CDS NAME field.
   b) Enter the aggregate group name in the AGGREGATE GROUP NAME field. The name of the aggregate group should correspond to the APPLID field used in the IBM Z Batch Resiliency application table.
   c) Enter '3' on the SELECT ONE OF THE FOLLOWING OPTIONS line to continue defining this aggregate group.
   d) Press the Enter key.

   The Aggregate Group Define panel shown in the following figure is displayed.

```
                           AGGREGATE GROUP DEFINE      ENTER OUTPUT DSN PREFIX
Command ===>

SCDS Name  . . . . . : SMS.ZOS15.SCDS
Aggregate Group Name : gl

To DEFINE Aggregate Group, Specify:
 Description ==> general ledger
             ==>

 Backup Attributes
   Number of Copies  . . . . . . . 1               (1 to 15)
   Management Class Name . . . . . dr              (1 to 8 characters, to be
                                                   defined in current SCDS)

 Output Data Set Name Prefix . . . IZBR.gl
                                                   (1 to 33 Characters)
 Account . . . . . . . . . . . . . dracct
                                                   (1 to 32 Characters)

Use ENTER to Perform Verification; Use DOWN Command to View Next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 45. Aggregate Group Define panel

3. On the Aggregate Group Define panel, enter the description, the number of backup copies to make, the output data set name prefix and the JCL account for the backup jobs for this aggregate group. Then, press the Enter key. The panel shown in the following figure displays.

   Note: Although not required by IBM Z Batch Resiliency, it is best practice for this prefix to contain a consistent high-level node for the first qualifier, and the APPLID as the second qualifier, so output data sets can be easily identified for each application.

```
                          AGGREGATE GROUP DEFINE              Page 1 of 2
Command ===>

SCDS Name  . . . . . : SMS.ZOS15.SCDS
Aggregate Group Name : gl

To DEFINE Aggregate Group, Specify:
 Description ==> general ledger
             ==>

 Backup Attributes
   Number of Copies  . . . . . . . 1               (1 to 15)
   Management Class Name . . . . .                 (1 to 8 characters, to be
                                                   defined in current SCDS)

 Output Data Set Name Prefix . . . IZBR.gl
                                                   (1 to 33 Characters)
 Account . . . . . . . . . . . . . 1234
                                                   (1 to 32 Characters)

Use ENTER to Perform Verification; Use DOWN Command to View Next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 46. Aggregate Group Define panel (Page 1 of 2)

4. On the next panel, enter the name of the IBM Z Batch Resiliency BKUPJCL library. The member name must correspond to the APPLID for the aggregate group. If BKUPJCL was previously executed, you can, from this panel, edit the selection data set generated by IBM Z Batch Resiliency. The selection data set contains INCLUDE cards in the following format:

```
ALLOCATE(PROD.GL.TREX543,
         PROD.GL.HAXC456)
INCLUDE (PROD.GL.TRDS01,
         PROD.GL.TRDS37,
         PROD.GL.TRX.CNTL.,
         PROD.GL.XTR.GDG.FILE.*,
         PROD.GL.ZBASE.SBRNSRCE.)
```

   After the aggregate group is defined to ABARS with the member from the IBM Z Batch Resiliency BKUPJCL library, you can use BKUPJCL to automatically add or remove entries from the selection data set.

5. At this point, activate the aggregate to use the aggregate.

```
                          AGGREGATE GROUP DEFINE              Page 2 of 2
Command ===>

SCDS Name  . . . . . : SMS.ZOS15.SCDS
Aggregate Group Name : GL
To Edit a Data Set, Specify Number  . .     (1, 2, 3, 4, 5, or 6)
  Selection Data Sets:            (1 to 44 characters)
   1  . . 'IZBR.v4x0.bkupjcl.abars'
     Member Name  . . gl          (1 to 8 characters)
   2  . .
     Member Name  . .             (1 to 8 characters)
   3  . .
     Member Name  . .             (1 to 8 characters)
   4  . .
     Member Name  . .             (1 to 8 characters)
   5  . .
     Member Name  . .             (1 to 8 characters)

  Instruction Data Set:           (1 to 44 characters)
   6  . .
Use ENTER to Perform Verification; Use UP Command to View Previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 47. Aggregate Group Define panel (Page 2 of 2)

Evaluating control cards (ABARVFY)

The IBM Z Batch Resiliency ABARVFY utility evaluates the control cards that are generated by BKUPJCL and comments out statements referring to data sets that are not cataloged at invocation time. It then optionally issues the command to initiate the ABARS started task. ABARVFY is supplied in the CNTL library as a standalone job to initiate ABARS processes and is invoked in the procedures generated when JCL=YES is specified. The following is a sample of ABARVFY JCL.

```
//jobname JOB ...
//*
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL.)
//*====================================================================
//* PROGRAM:  ABARVFY
//* FUNCTION: COMMENT OUT UNCATALOGED FILES FOR ABARS PROCESS
//*====================================================================
//*
//ST1 EXEC PGM=IKJEFT01
//SYSEXEC  DD DSN=HLQ.SBRNSRCE,DISP=SHR
//SYSTSPRT DD DUMMY
//REPORT   DD DSN=HLQ.REPORTS(ABARVFY),DISP=SHR
//USERIN   DD DSN=HLQ.USERMAC(ABARVFY),DISP=SHR
//SYSPRINT DD DCB=(RECFM=VBA,LRECL=125,BLKSIZE=629),
//            SPACE=(CYL,(1,1)),UNIT=VIO,DISP=(,DELETE,DELETE)
//SYSIN    DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//            SPACE=(CYL,(1,1)),UNIT=SYSDA,DISP=(,DELETE,DELETE)
//*====================================================================
//*  ABARVFY IS DESIGNED TO VERFIY AGGREGATE DEFINITIONS FROM A PDS.
//*
//*  IF NO MEMBER SPECIFIED, ALL MEMBERS WILL BE PROCESSED.
//*  IF A MEMBER NAME IS SPECIFIED AS THE THIRD PARAMETER IN
//*  DD SYSTSIN THEN ONLY ONE MEMBER IS PROCESSED.
//*  EG. '%ABARVFY DRVRI.BKUPJCL.ABARS MEMBER1'
//*
//*  PLEASE REFER TO THE IZBR MANUAL FOR CODING OF
//*  DD USERIN AND MORE SPECIFICS.
//*====================================================================
//SYSTSIN  DD *
%ABARVFY HLQ.BKUPJCL.ABARS
//
```

Figure 48. ABARVFY JCL sample

You can modify this job to evaluate only members listed in the optional PARAMETERS DD statement. In addition, you can use the utility to issue HSEND commands by tailoring the execution parameters as defined in the USERIN DD statement USERMAC(ABARVFY). The following table describes the ABARVFY options.

Table 22. ABARVFY options

Option	Parameter	Description
SEND=	YES	After evaluating the control cards, issue the DF/HSM command ABACKUP to initiate the backup for this aggregate.
	WAIT	Same as YES, but wait for ABARS backup processing to complete.
	NO	Do not issue the ABACKUP command after the member has been evaluated.
TYPE=	3480	The device type to issue with the ABACKUP command.
PARM=	VERIFY	Command issued to DFHSM when SEND=Y. HSEND ABACKUP applid UNIT(3480) VERIFY. May be a combination of any optional ABACKUP sub-parameters.

ABARVFY JCL is built for each aggregate when JCL=YES is specified in the USERMAC(XABARS) driver. This eliminates the requirement for running this JCL from CNTL library.

Generating ABARVFY JCL for each aggregate

When you set the JCL parameter to YES in the XABARS driver, IBM Z Batch Resiliency generates the ABARVFY JCL required for each aggregate defined in the BKUPJCL.ABARS control deck.

In this case, it is necessary to code a valid library to contain the JCL required to run the ABARVFY procedure. The library specified must be of PDS organization suitable for JCL, that is, LRECL=80 and DSORG=PO. The JCL created invokes the PROC XABARBKP in the CNTL library. The following is a sample of the JCL created.

```
//jobname JOB ...
//JS10    EXEC ABARBKP
//PS10.SYSTSIN DD *
 %ABARVFY HLQ.BKUPJCL.ABARS APPL1
//PS10.USERIN DD *
  SEND=YES
  TYPE=3490
  PARM=EXECUTE
```

Figure 49. ABARVFY JCL

In this example, the ABARS control statements in member APPL1 in IBM Z Batch Resiliency. V1R1M0. BKUPJCL.ABARS is verified before the procedure issues the command HSEND ABACKUP APPL1 UNIT(3490) EXECUTE. This JCL is written to the library defined by ABARJCL= in USERMAC(XABARS).

Capturing ABARS logs (BKUPBARS)

ABARS backups enable ABARVFY to capture ABARS logs immediately upon completion of backups. This can be useful when resolving problems in ABARS backup processing.

BKUPBARS details

As distributed by IBM, all commands are sent to DFSMShsm using the HSEND primary TSO command. Partial command results are sent to the terminal of the TSO user using TPUT, and it is not currently possible to capture this information. In addition, ABARS commands are written to the log data set created by the ABARS secondary address space and the name of this data set is also sent to the TSO terminal by TPUT.

Such practice is quite inconvenient; one accidental press of an ENTER key removes the results from the TSO screen.

BKUPBARS requirements:

- BKUPBARS and ARCEDEXT require APF authorization (when the WAIT parameter is specified).
- The command SETSYS EXITON(ED) must be issued to DFSMShsm for the exit to be called. To ensure that the exit is always enabled, include the command in the ARCCMDxx startup HSM command set. Only when WAIT parameter is specified.
- The Batch Requestor ensures that the exit is enabled and reports the status of all exits. If the exit is not enabled, the Batch Requestor terminates with an appropriate message and a non-zero return code (4 if exit is off, 8 if it has never been activated). Only done when WAIT parameter is specified.
- The user or job that issued the ABACKUP command must have DFSMShsm administrator privileges or appropriate security permits.
- In the ABARS procedure, a STEPLIB must be added that points to the IBM Z Batch Resiliency library where the exit resided. The ABARS procedure that needs to be modified is specified in ABARSPROCNAME(xxxxxxxx) in member ARCCMDxx in the //HSMPARM data set specified in the DFSMShsm started task. Alternatively, the exit module can be placed in a library in the LINKLIST concatenation. Only necessary when WAIT parameter is used.

IBM Z Batch Resiliency utility BKUPBARS is written to overcome the problem of capturing logs from the ABACKUP command and to allow further real-time updates of the backup table. This utility requires an IBM Z Batch Resiliency - provided ABARS exit ARCEDEXT – Expiration Date exit. This exit as provided by

IBM Z Batch Resiliency does not in any way change the expiration date processing. There are two modules in the process – BKUPBARS and ARCEDEXT.

The following commands are expected to be present in the DD:CMDIN.

- HSEND WAIT ABACKUP <parameters>
  - The Backup requestor sends the ABACKUP command to DFSMShsm, waits for the task completion, and returns ABARS log to the Batch Requestor. Batch Requestor has to be APF-authorized and ARCEDEXT is required.
- HSEND ABACKUP <parameters>
  - The Backup requestor sends the ABACKUP command to DFSMShsm, and does not wait for the request completion. The ABARS log is not returned to the Batch Requestor. In this case, APF-authorization for the Batch Requestor is not needed and ARCEDEXT is not used.
- ABACKUP <parameters>
  - Same logic as for HSEND ABACKUP command above.

```
//*
//* Run Batch pgm to send a command to HSM and set up
//*  global tables for ARCEDEXT
//*
//RUNSAB2  EXEC PGM=BKUPBARS,
//     PARM='MSGLVL(3),SVCDUMP(N),TRACE(N),TYPERUN(RUN)'
//*  STEPLIB  DD DISP=SHR,DSN=&VAPLIB       <- in linklist
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CMDIN    DD *
 HSEND WAIT ABACKUP AXB VERIFY
/*
//*
//* ABARS log file name is written to DD:ABARSLGN,
//* if allocated.
//*
//ABARSLGN  DD  DSN=&&LGN,
//          UNIT=SYSALLDA,
//          DISP=(NEW,PASS),
//          LRECL=44,RECFM=F,DSORG=PS,
//          SPACE=(TRK,1)
//*
//* Just to test, copy the ABARS log name to sysout
//*
//GENERLGN  EXEC PGM=IEBGENER,COND=(0,LT)
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   DUMMY
//SYSUT1    DD   DSN=&&LGN,DISP=(OLD,DELETE)
//SYSUT2    DD   SYSOUT=*
```

Figure 50. BKUPBARS JCL

BKUPBARS utility together with ARCEDEXT exit returns the entire ABARS log data set to the sysout (DD:ABARSLOG) and in addition writes a one-line data set (DD:ABARSLGN) that contains the data set name of the ABARS log data set, if further processing of this data set is required. See WAIT parameter for HSEND command above.

DD:CMDIN contains the ABARS ABACKUP command to be sent to DFSMShsm.

The ABACKUP command provided in DD:CMDIN input data set is sent unchanged to DFSMShsm and the Batch Requestor waits for the command to complete. When the command completes, the return code from the ABACKUP command is used as the step return code. The name of the log data set generated by ABARS is reported in the DD:SYSPRINT of the BKUPBARS job step. The name of the ABARS log data set is also written to DD:ABARSLGN. Again, see WAIT parameter for HSEND command above.

If ARCEDEXT exit is active and a standard (HSEND ABACKUP) command is issued, processing flow is the same as if the exit does not exist.

If the BKUPBARS program is used to issue an ABACKUP command and an exit is not available to the ABARS secondary address space (but is enabled), the processing is also the same as if HSEND command is issued.

In both cases above, the name of ABARS log data set is not available.

Backing up online DASD data sets to tape (FDR/ABR)

You can use FDR/ABR to back up online DASD data sets to tape. The generated JCL executes a procedure that resides in the CNTL library. The members of both BKUPJCL.FDR and RESTJCL.FDR correspond to the APPLIDs defined to this backup methodology.

Note: Run only in a non-production environment. The member $DELETE$ is created by RESTJCL to optionally delete all of the critical data sets associated with this backup methodology.

The name of the output data set is generated by IBM Z Batch Resiliency and the backup table is automatically updated upon successful completion of the process.

Table 23. FDR/ABR information

FDR/ABR information	
NAME IN BKUPMETH	FDR
Procedure in the CNTL Library (BACKUP)	XFDRBKUP
DD Statement in BKUPJCL	Dynamic allocation
BKUPJCL template in USERMAC	AFDR

The following figure is a sample of the JCL BKUPJCL created for FDR/ABR:

```
//GL    JOB (),CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*IZBRPROC JCLLIB ORDER=(HLQ.CNTL.)
//*-------------------------------------------------------------------
//*-------------------------------------------------------------------
//* APPLID = GL
//*-------------------------------------------------------------------
//JS0010   EXEC XFDRBKUP,
//         APPLID=GL,
//         BDSN='IZBR.ARCHIVE.GL.D060828.T070219.T0010'
//PS20.SYSIN DD *
DUMP TYPE=ARC,
   ARCBACKUP=DSF,BUFNO=MAX,ONLVOL,SCRATCH=NO,
   MAXCARDS=1000,COMPRESS=ALL,DSNENQ=USE,RETPD=2
 SELECT CATDSN=HLQ.BKUPTBL
 SELECT CATDSN=HLQ.SBRNSRCE
/*
```

Figure 51. BKUPJCL for FDR/ABR

The following procedure is in CNTL(XFDRBKUP):

```
//*====================================================================
//* SAMPLE FDR-ABR PROCEDURE TO BACK UP CRITICAL FILES
//*
//* NOTE: REVIEW THIS JCL PROCEDURE AND ALTER THOSE PARAMETERS THAT
//*       ARE SPECIFIC TO YOUR INSTALLATION.
//*       THEN COPY THE UPDATED PROCEDURE TO A SUITABLE PROCEDURE
//*       LIBRARY ON YOUR SYSTEM.
//*====================================================================
//XFDRBKUP     PROC RGN=6M,
//             SOUT='*',
//             WUNIT=SYSDA,            <-- SET MANUALLY AT INSTALL
//             TUNIT=3490,             <-- SET MANUALLY AT INSTALL
//             APPLID=, APPLID   (REQUIRED)  LEN= 8 OR <
//             SID=,    SUBAPPLID(OPTIONAL)  LEN= 8 OR <
//             UIN=,    USERINFO (OPTIONAL)  LEN=12 OR <
//             CPY=1    CPY      (DEFAULT=1) 1 OR 2
//PS10        EXEC PGM=FDRARCH,REGION=&RGN
//SYSPRINT    DD SYSOUT=&SOUT
//ARCHIVE     DD DSN=&BDSN,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=&WUNIT,SPACE=(CYL,(1,1),RLSE)
//SYSIN       DD DSN=HLQ.USERMAC(FDRCARD),DISP=SHR
//*--------------------------------------------------------------------
//PS20        EXEC PGM=FDRABR,REGION=&RGN
//SYSPRINA    DD SYSOUT=&SOUT
//SYSPRINT    DD DSN=&&FDROUT,DISP=(NEW,PASS),
//            UNIT=&WUNIT,SPACE=(CYL,(1,1)),
//            DCB=(BLKSIZE=12100,DSORG=PS,LRECL=121,RECFM=FBA)
//ARCHIVE     DD DSN=&BDSN,
//            DISP=(OLD,DELETE)
//TAPEA       DD DSN=TAPEA,DISP=(NEW,KEEP),
//            UNIT=&TUNIT
//*--------------------------------------------------------------
//PS30     EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1   DD DSN=&&FDROUT,DISP=SHR
//SYSUT2   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*--------------------------------------------------------------
//PS40     EXEC PGM=BRNVBFDR,
//         PARM=('AID=&APPLID,SID=&SID,UIN=&UIN,CPY=&CPY')
//STEPLIB  DD DSN=HLQ.LIBRARY,DISP=SHR
//FDROUT   DD DSN=&&FDROUT,DISP=SHR
//DDIN     DD DSN=&BRNVTEMP,DISP=(NEW,DELETE),
//         UNIT=SYSDA,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=1741,BLKSIZE=31338,RECFM=FB,DSORG=PS)
//BKUPTBL  DD DSN=HLQ.BKUPTBL,DISP=SHR,
//            AMP=('BUFNI=12,BUFND=49')
//BKUPAUD  DD DSN=HLQ.BKUPAUD,DISP=(SHR)
//SYSUDUMP DD SYSOUT=&SOUT
//ERRORS   DD SYSOUT=&SOUT
//ERRVSAM  DD SYSOUT=&SOUT
```

Figure 52. FDR/ABR procedure

Backing up VSAM data sets to tape (FAVER)

You can use FAVER to back up VSAM data sets to tape. The generated JCL executes a procedure that resides in the CNTL library. The members of BKUPJCL.FAVER correspond to the APPLIDs defined to this backup methodology. The name of the output data set is generated by IBM Z Batch Resiliency and the Backup Table is automatically updated upon successful completion of the process.

Table 24. Backing up VSAM to tape

FAVER information	
NAME IN BKUPMETH	FAVER
Procedure in the CNTL Library (BACKUP)	XFVRBKUP
DD Statement in BKUPJCL	Dynamic allocation
BKUPJCL template in USERMAC	AFAVER

The following is a sample of the JCL BKUPJCL created for FAVER:

```
//jobname JOB ...
//*-------------------------------------------------------------------
//* IZBR BACKUP JOB
//* (JOBNAME ABOVE MUST BE 6 CHARACTERS OR LESS)
//* DO NOT EXCEED 20 LINES IN THIS MEMBER
//*-------------------------------------------------------------------
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL.)
//*-------------------------------------------------------------------
//JS10    EXEC XFVRBKUP,
//             APPLID=GL,
//             BDSN='IZBR.BK.GL.D150216.T1122.FVR001'
//PS10.SYSIN DD *
  EXPORT EXO
  CLUSTER
  CL=HLQ.BKUPTBL
  CL=HLQ.INVENTRY
```

Figure 53. BKUPJCL for FAVER

The following procedure is in CNTL(XFVRBKUP):

```
//XFVRBKUP PROC APPLID=,      APPLID ( REQUIRED)  LEN= 8OR<
//             SUBAPPL=,      SUBAPPLID(OPTIONAL) LEN= 8OR<
//             USERINFO=,     USERINFO(OPTIONAL)  LEN=12OR<
//             BDSN=,
//             TUNIT=3490
//*====================================================================
//* THIS STEP WILL BACK UP ALL OF THE SPECIFIED DATA SETS TO
//* SCRATCH TAPES.
//*====================================================================
//PS10    EXEC PGM=GVEXPORT
//SYSPRINT DD  DSN=&SYSOUT,DISP=(NEW,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
//             DCB=(RECFM=VBA,DSORG=PS,LRECL=121,BLKSIZE=1214)
//FVROUT0  DD  DSN=&BDSN,
//             UNIT=&TUNIT,VOL=(,RETAIN,,50),
//             DISP=(,CATLG),LABEL=(,SL,EXPDT=99000)
//SYSIN    DD  DUMMY
//*--------------------------------------------------------------------
//PS20    EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1   DD  DSN=&SYSOUT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DUMMY
//*--------------------------------------------------------------------
//PS30    EXEC PGM=BRNVYSIN,PARM=('&BDSN')
//STEPLIB  DD  DSN=HLQ.LIBRARY,DISP=SHR
//BRNVOUT  DD  DSN=&&SYSIN,DISP=(,PASS),DSORG=PS,
//             UNIT=SYSDA,SPACE=(CYL,1),
//             DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//ERRLOG   DD  SYSOUT=*
//*--------------------------------------------------------------------
//PS40    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DSN=&&SYSIN,DISP=(OLD,DELETE,DELETE)
//*--------------------------------------------------------------------
//PS50    EXEC PGM=BRNVBFVR,
//             PARM=('&APPLID,&SUBAPPL,&USERINFO'),
//             COND=(0,NE,PS40)
//STEPLIB  DD  DSN=HLQ.LIBRARY,DISP=SHR
//TAPEHOLD DD  DSN=*.PS10.FVROUT0,
//             DISP=OLD,UNIT=&TUNIT
//FVROUT   DD  DSN=&SYSOUT,DISP=SHR
//DDIN     DD  DSN=&BRNVTEMP,DISP=(NEW,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=1741,BLKSIZE=31338,RECFM=F,DSORG=PS)
//BKUPTBL  DD  DSN=HLQ.BKUPTBL,DISP=SHR,
//             AMP=('BUFNI=12,BUFND=49')
//BKUPAUD  DD  DSN=HLQ.BKUPAUD,DISP=SHR
//ERRORS   DD  SYSOUT=&SOUT
//ERRVSAM  DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
```

Figure 54. APPLID XFVRBKUP procedure

Backing up online DASD data sets to tape (CA-DISK/DMS/OS)

You can use CA-DISK DMS/OS to back up online DASD data sets to tape. The generated JCL executes a procedure that resides in the CNTL library. The members of BKUPJCL.DMS correspond to the APPLIDs defined to this backup methodology. The name of the output data set is generated by IZBR and the backup table is automatically updated upon successful completion of the process.

Table 25. Backing up online DASD

DMS information	
NAME IN BKUPMETH	DMS
Procedure in the CNTL Library (BACKUP)	XDMSBKUP
DD Statement in BKUPJCL	Dynamic allocation
BKUPJCL template in USERMAC	ADMS

The following figure is a sample of the JCL BKUPJCL created for DMS OS:

```
//jobname JOB ...
//*-------------------------------------------------------------------
//* IZBR BACKUP JOB
//* (JOBNAME ABOVE MUST BE 6 CHARACTERS OR LESS)
//* DO NOT EXCEED 20 LINES IN THIS MEMBER
//*-------------------------------------------------------------------
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL.)
//*-------------------------------------------------------------------
//JS10     EXEC DMS
//SYSPRINT DD DSN=&&SYSPRINT,DISP=(NEW,PASS),UNIT=SYSDA,
//            SPACE=(CYL,(5,5),RLSE)
//MSGPRINT DD DSN=&&MSGPRINT,DISP=(NEW,PASS),UNIT=SYSDA,
//            SPACE=(CYL,(5,5),RLSE)
//SYSIN    DD  *
 ARCHIVE DSNAME=HLQ.CNTL.,
    NOSCRATCH,NOCATALOG,NORESET,RETPD=10
 ARCHIVE DSNAME=HLQ.LIBRARY,
    NOSCRATCH,NOCATALOG,NORESET,RETPD=10
 ARCHIVE DSNAME=HLQ.SBRNSRCE,
    NOSCRATCH,NOCATALOG,NORESET,RETPD=10
 ARCHIVE DSNAME=HLQ.USERMAC,
    NOSCRATCH,NOCATALOG,NORESET,RETPD=10
 ARCHIVE DSNAME=IZBR.STG372.PSOURCE,
    NOSCRATCH,NOCATALOG,NORESET,RETPD=10
 ARCHIVE DSNAME=IZBR.TOOLKIT,
    NOSCRATCH,NOCATALOG,NORESET,RETPD=10
//JS10     EXEC XDMSBKUP,
//         APPLID=GL,COPY=PRIMARY
```

Figure 55. BKUPJCL for CA-DISK DMS OS

The following procedure is in CNTL(XDMSBKUP):

```
//XDMSBKUP PROC APPLID=,  APPLID ( REQUIRED)  LEN= 8OR<
//         SUBAPPL=,      SUBAPPLID(OPTIONAL) LEN= 8OR<
//         USERINFO=,     USERINFO(OPTIONAL)  LEN=12OR<
//         BDSN=,
//         RGN=6M,
//         HLQDMS='DMS.DASD',      <-- SET MANUALLY AT INSTALL TIME
//         WUNIT=SYSDA,            <-- SET MANUALLY AT INSTALL TIME
//         SOUT='*',
//         TUNIT=3490              <-- SET MANUALLY AT INSTALL TIME
//*====================================================================
//* THIS STEP WILL BACK UP ALL OF THE SPECIFIED DATA SETS TO
//* SCRATCH TAPES.
//* XDMSBKUP: SAMPLE DMS PROCEDURE TO BACK UP CRITICAL FILES
//*====================================================================
//*
//* NOTE: REVIEW THIS JCL PROCEDURE AND ALTER THOSE PARAMETERS THAT
//*       ARE SPECIFIC TO YOUR INSTALLATION
//*       THEN COPY THE UPDATED PROCEDURE TO A SUITABLE PROCEDURE
//*       LIBRARY
//*
//PS10        EXEC PGM=ADSMI002,
//             PARM=ADSDM120,
//             REGION=&RGN
//*
//STEPLIB     DD   DISP=SHR,
//            DSN=&HLQDMS..LOADLIB
//*
//ABNLDUMP    DD   DUMMY
//AIX         DD   DYNAM
//ARCHIVEC    DD   DUMMY
//ARCHIVE0    DD   DSN=&BDSN,
//            DCB=(BLKSIZE=23476),
//            UNIT=&TUNIT,
//            VOL=(,RETAIN,,50),
//            DISP=(NEW,PASS),
//            LABEL=(,SL,EXPDT=99000)
//A1          DD   DUMMY,DISP=SHR
//A2          DD   DUMMY,DISP=SHR
//CMDPRINT    DD   SYSOUT=&SOUT
//DSNEXPDT    DD   UNIT=&WUNIT,
//            SPACE=(1024,(1000,1000),,,ROUND)
//*
//FILES       DD   DISP=SHR,DSN=&HLQDMS..FILES
//MSGPRINT    DD   SYSOUT=&SOUT,DCB=BUFNO=0
//PARMLIB     DD   DISP=SHR,DSN=&HLQDMS..PARMLIB
//REPORTS     DD   DISP=(NEW,PASS),
//            DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6140,RECFM=VB),
//            SPACE=(6144,(200,200),,,ROUND),UNIT=&WUNIT
//SYSPRINT    DD   DSN=&&SYSOUT,DISP=(MOD,PASS),
//            DCB=(BLKSIZE=23476,DSORG=PS,LRECL=141,RECFM=VB)
//SYSUDUMP    DD   SYSOUT=&SOUT
//VSAM        DD   DYNAM
//*-------------------------------------------------------------------
//PS20     EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1   DD DSN=&&SYSOUT,DISP=SHR
//SYSUT2   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*-------------------------------------------------------------------
//PS30     EXEC PGM=BRNVBDMS,
//         PARM=('&APPLID,&SUBAPPL,&USERINFO')
//STEPLIB  DD DSN=HLQ.LIBRARY,DISP=SHR
//TAPEHOLD DD DSN=*.PS10.ARCHIVE0,DISP=OLD
//DMSOUT   DD DSN=&&SYSOUT,DISP=SHR
//DDIN     DD DSN=&BRNVTEMP,DISP=(NEW,DELETE),
//         UNIT=SYSDA,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=1741,BLKSIZE=31338,RECFM=F,DSORG=PS)
//BKUPTBL  DD DSN=HLQ.BKUPTBL,DISP=SHR,
//         AMP=('BUFNI=12,BUFND=49')
//BKUPAUD  DD DSN=HLQ.BKUPAUD,DISP=(SHR)
//ERRORS   DD SYSOUT=&SOUT
//ERRVSAM  DD SYSOUT=&SOUT
//SYSUDUMP DD SYSOUT=&SOUT
```

Figure 56. CA-DISK DMS OS XDMSBKUP procedure

Copying critical tapes to backup volumes for offsite storage

You can use BRNVCPY to copy critical tapes to backup volumes that is rotated offsite. BRNVCPY simplifies the backup of critical tapes by using standard copy utilities such as IEBGENER.

Tapes are called for dynamically, as directed by the INTAPE control stream. Successful tape copies are updated in the IBM Z Batch Resiliency backup table and are used by the RESTJCL program at recovery time. The tape management system used at your location affects the means by which data is recovered. The ability to restart at a step is automatically provided in the event that a copy is not completed successfully.

As in the DASD utility backups, all members defined in BKUPJCL correspond to the APPLID associated with the data set. The figure Figure 58 on page 94 contains sample JCL for backing up critical tapes. Each file successfully backed up is immediately marked in the backup table (BKUPTBL) as complete. The main emphasis of the process is backing up the critical files so that the last 17 characters of the original data sets are maintained on the tape labels. By following this standard, IBM Z Batch Resiliency is able to uncatalog the original data sets and catalog the backup data sets so they appear to the operating system as the original files. This expedites recovery time significantly, since no tape IO is used to recover these files.

Procedure XTAPECPY

XTAPECPY has three steps that control the process of copying tapes.

The steps are summarized as follows:

PS10
The TAPEPARS program produces the control statements for the BRNVCPY program for an application.

PS20
The program BRNVCPY which performs the actual backup.

PS30
IEFBR14 step to delete the OUTTAPE data set created in PS05. This data set is kept if step PS10 ends with a return code greater than 0.

The OUTTAPE data set created in PS10 step is a unique permanent data set which directs the program BRNVCPY to back up the appropriate data sets.

The following JCL is created in BKUPJCL BRNVCPY:

```
//AG00003B JOB (XXXXXX,INWA),CLASS=C,MSGCLASS=X,
//         MSGLEVEL=(1,1),NOTIFY=&SYSUID
//* NOTE: UNCOMMENT IZBRPROC BELOW TO RUN PROCS FROM CNTL LIBRARY
//IZBRPROC JCLLIB ORDER=(AG.IZBR
                     V1R1M0
                     .CNTL.)
//*--------------------------------------------------------------------
//* BACKUP OF: MJE
//* JCL MADE:: D021103.T1006
//*--------------------------------------------------------------------
//JS001   EXEC XTAPECPY,
//        TIMESTMP=D021103.T1006,
//        APPLID=BRNV001
//INTAPE  DD *
BKUPHLQ(AG.DR)
INCLUDE(AG.PJE.TEST.TAPE1)
INCLUDE(AG.PJE.TAPE.F3,
        AG.PJE.TAPE.F4)
```

Figure 57. JCL Created in BKUPJCL BRNVCPY

The procedure XTAPECPY, in the following figure, resides in the IBM Z Batch Resiliency CNTL library and has several symbolic values that direct the operation of the BRNVCPY run.

```
//*====================================================================
//*  PROC FOR DATAMOVER BRNVCPY
//*====================================================================
//XTAPECPY PROC OUTCLASS='*',
//         UNIT=CART,
//         APPLID=APPLID,
//         RESTHLQ=IZBRHLQ,
//         TIMESTMP=TIMESTMP,
//         RETPD=5,
//*        RETPD=X2001365, OR EXPIRATION DATE LONGFORM
//*        RETPD=X99365,   OR EXPIRATION DATE SHORTFORM
//         BUFNO=10
//*====================================================================
//PS10    EXEC PGM=TAPEPARS
//STEPLIB  DD  DSN=HLQ.LIBRARY,DISP=SHR
//SYSTSPRT DD  SYSOUT=&OUTCLASS
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//SYSEXEC  DD  DSN=HLQ.SBRNSRCE,DISP=SHR
//TCEXIT   DD  DUMMY
//*====================================
//INTAPE   DD  DUMMY
//*====================================
//OUTTAPE  DD  DSN=&RESTHLQ..&TIMESTMP..&APPLID.,
//             DISP=(NEW,CATLG),
//             UNIT=SYSDA,DCB=(LRECL=2048,BLKSIZE=0,RECFM=VB),
//             SPACE=(CYL,(5,5))
//*====================================================================
//PS20    EXEC PGM=BRNVCPY,
//             COND=(0,NE,PS10),
//             PARM=('RETPD=&RETPD,UNIT=&UNIT,BUFNO=&BUFNO')
//STEPLIB  DD  DSN=HLQ.LIBRARY,DISP=SHR
//SYSPRINT DD  SYSOUT=&OUTCLASS,BLKSIZE=16093
//SYSIN    DD  DSN=&&SYSIN,DISP=(NEW,DELETE,DELETE),
//             UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,BUFNO=1)
//DDIN     DD  DSN=&BRNVTEMP,DISP=(NEW,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=1741,BLKSIZE=31338,RECFM=FB,DSORG=PS)
//INPUT    DD  DISP=OLD,DSN=&RESTHLQ..&TIMESTMP..&APPLID.
//TCDSSIN  DD  DSN=HLQ.USERMAC(TCDSSIN),DISP=SHR
//SYSUDUMP DD  SYSOUT=&OUTCLASS
//TRACE    DD  SYSOUT=&OUTCLASS
//MESSAGES DD  SYSOUT=&OUTCLASS
//ERRORS   DD  SYSOUT=&OUTCLASS
//ERRVSAM  DD  SYSOUT=&OUTCLASS
//BKUPTBL  DD  DSN=HLQ.BKUPTBL,DISP=SHR,
//             AMP=('BUFNI=12,BUFND=49')
//BKUPAUD  DD  DSN=HLQ.BKUPAUD,DISP=SHR
//*====================================================================
//PS30    EXEC PGM=IEFBR14,COND=(0,NE,PS20)
//RESTART  DD  DISP=(OLD,DELETE),
//             DSN=&RESTHLQ..&TIMESTMP..&APPLID.
```

Figure 58. Sample JCL for backing up critical tapes

Table 26. XTAPECPY defaults

Symbolic	Value	Description
UNIT=	CART	Unit used for output tape volumes.
MODE=	3.5	Reserved. Do not modify.
RETPD=	5	Retention period of output volumes in days.
RESTHLQ=	IZBRHLQ	High-level qualifier assigned to BRNVCPY restart data set.
APPLID=	(no value)	APPLID. Reserved for use of invoking JCL only.
TIMESTMP=	(no value)	Time Stamp. Reserved for use of invoking JCL only.
BUFNO=	10	Output buffers assigned for output volumes.

Creating OUTTAPE (TAPEPARS) data set

Step PS10 of procedure BRNVCPY (program TAPEPARS) creates the OUTTAPE data set that is used in step PS20 (program BRNVCPY) to perform the backups and pass information into the Backup Table.

You can specify several parameters in the input file defined by the INTAPE DD statement. Each parameter has the form: *name(value)*. These parameters are passed to the BRNVCPY program and affect the way that the file is copied, and the information stored for it in the backup table.

Table 27. Creating OUTTAPE (TAPEPARS)

Parameter	Values	Description
APPLID	8-character string	The application associated with this data set. A file can reside multiple times in the backup table for multiple applications and sub-applications.
SUBAPPL	8-character string	A sub-application for the data set. A file can reside multiple times in the backup table for multiple applications and sub-applications.
RETPD	5	Retention period of output volumes. • If specified as a number (for example, 5), this is the number of days to hold the tapes. • If specified as Xyyyyddd or Xyyddd (for example, 2007365 or 99365), this is the Julian date when the tapes expire.
EXPDT	5	Expiration date of output volumes. If specified as a number (for example, 5), this is the number of days to hold the tapes. If specified as Xyyyyddd or Xyyddd (for example, 2007365 or 99365), this is the Julian date when the tapes expire.
CATBACK	YES NO OVERLAY	Action if the output data set name already exists in the system catalog. • Yes: The file is copied but not cataloged. The step gets a NOTCAT2 condition code. • No: This file is not backed up. • Overlay: The entry in the catalog is replaced with the new entry.
STORCLAS	8-character string	SMS Storage Class.
DATACLAS	8-character string	SMS Data Class.
MGMTCLAS	8-character string	SMS Management Class.
UNIT	Legitimate tape device	Tape device type where backup files are written.
USERIN or UIN	12 characters	User information for this file.
BLKSIZE	Numeric	Block size for the copied file. This parameter overrides the individual file attributes.

Table 27. Creating OUTTAPE (TAPEPARS) (continued)		
Parameter	**Values**	**Description**
BKUPHLQ		High Level Qualifier used to name the backup data sets. The trailing 17 characters of the backup data set name are the same as original data set.
PGM	8-character string IEBGENER BRNVGENR ADRDSSU FDRTCOPY ICEGENER SYNCGENR	Program to use to copy the file. This parameter overrides the individual file attributes.
SYSINDD	8-character string	DDNAME for the file that provides parameters to the data mover program. This parameter overrides the individual file attributes.
DDIN	8-character string DDIN	DD Name for BRNVCPY input file. This parameter overrides the individual file attributes.
DDOUT	8-character string DDOUT	DD Name for BRNVCPY output file. This parameter overrides the individual file attributes.

Tape stacking critical tape volumes

IBM Z Batch Resiliency provides a basic tape stacking facility with the IBM Z Batch Resiliency Tape Copy process to allow multi-file or non-multi-file tape volumes to be stacked together within an application group. This facility provides the capability of reducing the number of tape volumes that are rotated to off-site storage.

To enable this facility, set the BRNSTCVLS parameter to YES in the USERMAC(XTAPECPY). In this case, tapes are stacked limited by the parameters set for MAXMEG and MAXSTAC. Note that, by default, Tape Copy does not stack multi-volume or multi-file data sets with other data sets; multi-file inputs are stacked together without other files.

Dense stacking

By setting the USERMAC(XTAPECPY) option STACVOL to DENSE (STACVOL=DENSE), multi-volume and multi-file data sets become eligible for stacking. This option is useful when using high-density output media for tape backups. Be advised though that dense stacking increases the potential for device contention at the recovery site and should be used with selective unstacking there to minimize the time to recover files.

Tape stacking exclude list

Use the member USERMAC(EXCLSTAC) to exclude specific or patterned data sets from the process of being stacked. The syntax is like that of EXCLDSN where the data set name or pattern is coded in lines 1 through 44 and a question mark (?) or number sign (#) is used as a placeholder and an asterisk (*) is used to mask an entire node. To include anything past a certain point, use double asterisks (**). The conventions are modeled after IBM's DF/DSS.

Backup Dynamic Data set Scheduling

Backup Dynamic Data set Scheduling is an optional feature that schedules backups of critical data sets to be run at intervals throughout the day, providing the ability to reduce the backup window.

The Backup Dynamic Data set Scheduler tracks application data sets and builds backup windows based on event criteria defined by the installation. Event criteria are established on a data set level and are used by Backup JCL generating program when submitting segmented backups over a broad range time.

You can define which applications and data sets are to be eligible for scheduling and how to schedule them. At scheduled intervals, backups are run for the applications that are eligible to be backed up.

The Backup Dynamic Data set Scheduler consists of the several components.

Components of the Backup Dynamic Data set Scheduler

Backup Dynamic Data set Scheduler is a subtask of the RT/SMF Dispatcher and defined in USERMAC(VFISTASK).

- **SCHDPARM** – contains the IZBR/CKK start-up parameters found in USERMAC. Four start up parameters can be assigned, as described in Activating Backup Dynamic Data Set Scheduling.
- **SCHDREQ** – defines what is to be scheduled and the scheduling criteria in the parameter table found in USERMAC.
- **SCHEDULE EVENT ARRAYS (SEA)** – contains scheduling information for all scheduled data sets and the keys that point to the scheduled data set's records in the Inventory database.
- **Inventory scheduling fields** – Inventory records that define the scheduling criteria.
- **BKUPJCLS** – interprets the schedule from the Inventory database and creates the JCL to back up eligible files.
- **VFISUB** – updates the backup information in the Inventory database for the records being backed up, and optionally submits the backup job.
- **VFISUBR** – reports on the contents of the subfile2 data set.
- **Backup Dynamic Data set Scheduler reports** – creates the reports: SCHELIGR and SCHPENDR using batch jobs in the CNTL library.
- **Backup Dynamic Data set Scheduler commands** – are executed by the Command Processor. See Backup Dynamic Scheduler execution commands for a description of these commands.

Activating Backup Dynamic Data set Scheduling

Backup Dynamic Data set Scheduling is a real-time subtask that runs under the control of the RT/SMF Dispatcher.

About this task

To activate the Scheduler subtask IZBR/CKK, complete these steps:

Procedure

1. Activate the subtask in USERMAC(VFISTASK) by removing the leading asterisk (*) for this statement:

   ```
   START TASK=IZBR/CKK,PGM=BRNVC008,SMF=(30)
   ```

   Note: VFISTASK is distributed with the above entry commented out.
2. Submit member in CNTL(SCHDALLC) to allocate the submit file (SUBFILE) and journaling data sets.
3. Uncomment the SCHDJDS scheduling journal data set statement in the startup procedure in CNTL(RTSMFP). The statement looks similar to the following:

   ```
   //SCHDJDS   DD DISP=SHR,DSN=IZBRHLQ.SCHDJDS
   ```

4. Check the IZBR/CKK startup parameters defined in USERMAC(SCHDPARM). The distributed values of the following parameters might need to be increased depending on your requirements.

   See Defining IZBR/CKK startup parameters for a description of the startup parameters and distributed values.

   - SEAMAXJB = 5000
   - SEAMAXDS = 50000
   - SEAJMAX = 100000
   - SEATMAX = 100000

   Note: Two (or more) spaces are required to delimit keywords/operands from comments.
5. The IZBR/CKK subtask requires that the scheduler request table (SCHDREQ) be loaded into storage for processing. To load SCHDREQ, uncomment the following statement in USERMAC(VFITABLS):

   ```
   SCHDREQ,PARMLIB,SCHDREQ,BRNVL011
   ```

   Note: VFITABLS is distributed with the above entry commented out.

Defining IZBR/CKK startup parameters

IZBR/CKK startup parameters are defined in USERMAC(SCHDPARM) member. Default values are underlined.

Table 28. IZBR/CKK Startup parameters

Parameter	Description
START = WARM | COLD	Type of IZBR/CKK start • COLD – rebuilds the schedule by applying SCHDREQ criteria to all the records in the Inventory database and build the Schedule Event Arrays. Use a cold start when you initially implement Backup Dynamic Dataset Scheduling. • WARM – does not rebuild the schedule but rather loads the schedule from the Inventory records and builds the Schedule Event Arrays. This is the default.
CRITLEV = 2 | 9	Level of criticality for scheduling files • 2 – dynamically schedules only critical files (status less than or equal to 2). • 9 – schedules all files in the Inventory database regardless of the status. Default is CRITLEV = 2.
GDGLEV= GCOUNT | ALL	Number of generations of GDGs that is scheduled. • GCOUNT – dynamically schedules generations of GDG based on the GCOUNT field in the Inventory database. • ALL – schedules all generations of GDGs in the Inventory database regardless of GCOUNT. Default is GDGLEV = GCOUNT.
SEAMAXDS = 50000	Defines the maximum number of data sets to be scheduled. Maximum value is 33,554,431.
SEAJMAX = 100000	Defines the maximum number of entries in Job Schedule Event Array (SEA/J). Maximum value is 107,374,182.
SEAMAXJB = 5000	Defines the maximum number of unique jobs in the schedule. Maximum value is 999,999.

Table 28. IZBR/CKK Startup parameters (continued)	
Parameter	**Description**
SEATMAX = 100000	Defines the maximum number of entries in the Time Schedule Event Array (SEA/T). Maximum value is 107,374,182.
SEAHWMP = 80	Defines the high-water mark percentage for the Schedule Event Arrays (SEA) tables before a warning message is displayed in RTDPRINT. Maximum value is 99.
JOURNAL = NO	Defines whether to use journal scheduling and rescheduling of data sets.
JRNLHLQ = *HLQ*	Defines the high-level qualifier of the journal data sets. Maximum length of *HLQ* is 24. Default is the installation high-level qualifier.
JRNLUNIT = SYSALLDA	Defines the unit name for the journal data sets (for example SYSDA).
JRNLVOL = TSO001	Defines DASD volume serial number for the journal data sets.
JRNLPRIM = 1	Defines primary space allocation in cylinders (CYL) for the journal data sets.
JRNLSEC = 3	This parameter defines secondary space allocation in cylinders (CYL) for the journal data sets.
⚠ **CAUTION:** DO NOT CHANGE ANY OF THE FOLLOWING PARAMETERS WITHOUT PRIOR INSTRUCTION FROM TECHNICAL SUPPORT.	
SCHLOADR = BRNVSCHD	Schedule load/reload routine.
SEALOADR = BRNVLSEA	Schedule Event Array (SEA) load/reload routine.
SEAINSR = BRNVEAIR	Schedule Event Array (SEA) insert routine.
SEADELR = BRNVSEAD	Schedule Event Array (SEA) delete routine.
SEAGETR = BRNVSEAG	Schedule Event Array (SEA) retrieve routine.
SRAACCR = BRNVRAAR	Schedule Request Array (SRA) access routine.
SCHJRNLR = BRNVC002	Scheduler Journal Processor routine.

Building Backup Dynamic Data set Scheduler criteria

You can use the USERMAC(SCHDREQ) to define the applications to be scheduled for backup and the scheduling criteria. The application's critical data sets are backed up periodically throughout the day.

About this task

To add an application to the Backup Dynamic Data set Scheduler complete the following steps.

Procedure

1. Edit the USERMAC(SCHDREQ) member.

   The following figure shows a sample SCHDREQ member:

```
/* SCHDREQ FOR SCHDTBL PROGRAM
/* 1   2    3     4      5      6      7       8
/*OPT,APPL,DSN,BKPMETH,EVENT1,EVENT2,CUTOFF,RESCHEDULE
/*INCL,,,,,C:02:00
/*INCL,DAILY
INCL,AP,,,%UPJOB,T:03:00
INCL,GG,,,%UPJOB,T:03:00
INCL,GL,,,%UPJOB,T:03:00
INCL,MU,,,%UPJOB,T:03:00
INCL,UR,,,%UPJOB,T:03:00
INCL,,,DSS,,T:20:00,,N:24:00
```

Figure 59. SCHDREQ member sample

2. For each application, add SCHDREQ parameters following the syntax described in Defining SCHDREQ parameters.
   For example, to add the application CM to the Backup Dynamic Data set Scheduling, add the following line to SCHDREQ:

```
INCL,CM,,,%UPJOB,T:03:00
```

Defining SCHDREQ parameters

Each line in SCHDREQ consists of eight positional parameters. A slash asterisk (/*") in column one indicates a comment line and its characters are not processed. Blank lines are not processed; a comma at the end of a line indicates continuation. Embedded blanks are not allowed.

From left to right, the parameters are:

`Option [,Application] [,Data Set] [,BackupMethod] [,Event1] [,Event2] [,Cutoff] [,Reschedule]`

Table 29. SCHDREQ parameters

Parameter	Description
Option	Defines the data sets to be backed up. Values are: • INCL – include CRITICAL data sets. • INCI – include from INCLDSN. Defines the data sets to exclude from a previous backup. • EXCL – exclude critical data sets. • EXCI – exclude from INCLDSN.
Application	(Optional) consists of one to eight characters, with no embedded blanks or commas. If the Option parameter is: • INCL or INCI – the default is all applications. • EXCL or EXCI – the default is null, that is, no exclusion. Enter a comma (,) to indicate any application.
Data Sets	(Optional) Specifies a data set name or a data set mask. If the Option parameter is: • INCL or INCI – the default is all data sets. • EXCL or EXCI – the default is null, that is, no exclusion. **Note:** The all data sets mask, double asterisks (**), is not allowed with the EXCL or EXCI option. Enter a comma (,) to indicate all data sets. To code this parameter, see DSN Coding in the table below.

Table 29. SCHDREQ parameters (continued)	
Parameter	**Description**
Backup Method	Specifies the name of the method used to back up the files. The method must be specified in the BKUPMETH file. If the Option parameter is: • INCL – All methods. (Optional). • INCI – All methods. (Required). • EXCL or EXCI, the default is null, that is, no exclusion. Enter a comma (,) to use the default.
Event1	(Optional) Applies to the INCL and INCI Option parameters only. • INCL – is the last referenced job (%LRJOB: see the Field Coding parameter below) for the data set or mask in question. • INCI – is N:23:00 (see New Time Coding below). **Note:** For INCI, either Event1 or Event2 or both must be specified. Enter a comma (,) to use the default. For more information, see Field Coding in the following table. Choose one type of coding only. **Note**: If the Option parameter is INCI, Field Coding cannot be used. Event1 or Event2, whichever occurs first, initiates backup processing for the data set in question.
Event2	(Optional) Applies to the INCL and INCI Option parameters only. No default is available for Event2. **Note:** For INCI, either Event1 or Event2 or both must be specified. Enter a comma (,) to skip this parameter and continue with the Cutoff parameter. To code this parameter, use the Job Name Coding, Time Coding, Relative Time Coding, New Time Coding, or Field Coding type. Choose one type of coding only. (For a description of the coding types, see Table 30 on page 102.) **Note:** If the Option is INCI, Field Coding cannot be used. Event1 or Event2, whichever comes first, initiates backup processing for the DATA SET in question.
Cutoff	(Optional) Applies to the INCL and INCI Option parameters only. It defines the end-incident for data set backup, after which a backup is not taken. To code this parameter, use the Job Name Coding, Time Coding, Relative Time Coding, New Time Coding or Field Coding type. Choose one type of coding only. For a description of the coding types, see Table 30 on page 102. **Note:** If the Option is INCI, Field Coding cannot be used. The default is the date and time 28 hours after the END parameter, which is N:28:00 (see New Time Coding below). If you choose a time that is before either Event1 or Event2, an error occurs.
Reschedule	(Optional) Resets the Backup Dynamic Dataset Scheduler based on the Inventory database. This parameter marks the end of the cycle for a data set and the beginning of the next cycle. The default value for this parameter is 24 hours after the last END event. When the end event has been encountered, all critical data sets are rescheduled for the next cycle. The eligible and pending flags are reset, and time-sensitive events are recalculated.

The following table describes how to code SCHDREQ parameters.

Table 30. Coding SCHDREQ parameters

Type	How to code
Placement	For each kind of inclusion (that is, CRITICAL and INCLDSN), place the more specific ahead of the less specific; that is, the more general after the less general.
DSN Coding	Code a full valid data set name, or code a DSN mask using these rules: • A single asterisk by itself indicates that either a qualifier or zero or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters. • Double asterisks (**) indicate that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a period or a blank. • A single percent sign, (%), by itself indicates that exactly one alphanumeric or national character can occupy that position. • One to eight percent signs, (%%)..., can be specified in each qualifier.
Job Name Coding	Code J: followed by a 1 to 8-character job name.
Time Coding	Code T: followed by a definite (military) time in the form HH:MM for hour and minute, with leading zeros (the range is 00:01 to 23:59). If a time is chosen which is after the END event (current day), the current date and that time is used; if a time is chosen before the END event, tomorrow's date and that time is used.
Relative Time Coding	Code C: followed by an additive time in the form HH:MM for hours and minutes, with leading zeros. C: stands for the creation time of the file in question, with HH:MM indicating the time to add to the file creation time before implementation of the requested function. For Event1 and Event2, HH:MM cannot exceed 16:00; for Cutoff, HH:MM cannot exceed 20:00. **Note:** Thus far, relative time coding is relative to creation time only, but other types may be added in future enhancements.
New Time Coding	Code N: followed by an additive time in the form HH:MM for hours and minutes, with leading zeros. N: stands for a new time after the runtime of the END event, with HH:MM indicating the time to add. For example, N:05:30 means 5 hours and 30 minutes from the END event. HH:MM cannot exceed 36:00.

Table 30. Coding SCHDREQ parameters (continued)	
Type	**How to code**
Field Coding	The following three field names, which specify jobs, are allowed: %LRJOB - the last reference job, %CRJOB - the creation job, and %UPJOB - the update job. • Example 1: from CRITICAL, include all GL.PROD.** data sets for the GL Application, default the Backup Method, Event, and Cutoff: INCL,GL,GL.PROD.** • Example 2: exclude all previously included SYS1.** data sets from CRITICAL: EXCL,,SYS1.** • Example 3: include all data sets from INCLDSN for the PAY Application, choose a Backup Method, relative creation time for Event1and a Cutoff time: INCI,PAY,,DSS,C:02:00,,T:20:15. • Example 4: include all data sets for the PAY Application, choose a Backup Method and a relative creation time for Event1, and default the Cutoff: – INCL,PAY,,DSS,C:02:00 – INCI,PAY,,DSS,C:02:00 • Example 5: from CRITICAL, include all PACCT.%TRNS.** data sets for the ACCOUNT Application, Backup-Method DSS, Event1 and Cutoff are jobs: – INCL,ACCOUNT,PACCT.%TRNS.**,DSS,J:JOB011,,J:JOB099 • Example 6: include everything, but exclude all DSS Backup-Method selections: – INCL – EXCL,,,DSS • Example 7: from CRITICAL, include everything with Event2 as the update job: – INCL,,,,,%UPJOB

SCHDREQ statement example

This section provides an example SCHDREQ statement.

```
INCL,,,DSS,,T:20:00,T:600,N:24:00
```

The scheduling criteria is defined as follows:

Table 31. Example SCHDREW statement scheduling criteria		
Position	**Value**	**Description**
1	INCL	Include critical data sets
2	,	For any application
3	,	Apply to all data sets
4	DSS	Use backup method DSS (DASD or Tape)
5	,	Default to the last referenced job that was run
6	T:20:00	08:00 p.m
7	T:600	Do not back up after 06:00 a.m
8	N:24:00	Reschedule 24 hours in the future

Rejected SCHDREQ statements

Any SCHDREQ statements that are rejected, are written to the RTPPRINT file of the RT/SMF subtask.

Backup Dynamic Data set Scheduler execution commands

This section provides a list of Backup Dynamic Data set Scheduler commands executed by the Command Processor via ISPF Menu Option 7 - Miscellaneous, Option 1 - RT/SMF Commands.

Table 32. Backup dynamic Data set scheduler commands

Command	Description
VFISCHED DISPLAY STATUS	Displays status information about the Dynamic Data set Scheduler subtask.
VFISCHED DISPLAY ARRAYS	Displays information regarding the schedule event arrays (SEA) in storage. The display lists the addresses, length, percent used, and number of entries.
VFISCHED DISPLAY SEA	
VFISCHED REBUILD	Rebuilds the schedule by reading Inventory records and applying the scheduling criteria defined in the SCHEDREQ table. This has the same effect as a COLD start for the Dynamic Data set Scheduler subtask.
VFISCHED REFRESH SCHDREQ	Refreshes SCHDREQ table in storage. This may be done after modifying USERMAC(SCHDREQ).
VFISCHED DUMP SEA	Dumps the storage contents of all the SEAs. The dump is written to the file allocated to DDname VFIDUMP.
VFISCHED DUMP IRKA	Dumps the storage contents of the IRKA schedule event array.
VFISCHED DUMP SEA/J	Dumps the storage contents of the SEA/J schedule event array.
VFISCHED DUMP SEA/T	Dumps the storage contents of the SEA/T schedule event array.
VFISCHED DUMP SCHDREQ	Dumps the storage contents of the SCHDREQ table.

```
UID: SPJE    CMD: VFISCHED DISPLAY STATUS
COMMAND: VFISCHED PROGRAM: BRNVSCMD ATTACHED, TCB: 008CEE88
VFISCHED COMMAND PROCESSING SUB-COMMAND DISPLAY
 VFISCHED DISPLAY STATUS

   IZBR SCHEDULER SUBTASK (IZBR/CKK) WARM START ON 2017.227 AT 11:14:33
   SCHEDULE EVENT ARRAYS (SEA) WERE LAST LOADED ON 2017.227 AT 11:14:33
         75 DATA SETS ARE CURRENTLY BEING SCHEDULED BY IZBR/CKK

VFISCHED REQUEST SUCCESSFUL: DISPLAY STATUS
```

Figure 60. DISPLAY STATUS command example

```
UID: SPJE    CMD: VFISCHED DISPLAY ARRAYS
COMMAND: VFISCHED  PROGRAM: BRNVSCMD  ATTACHED, TCB: 008CEE88
VFISCHED COMMAND PROCESSING SUB-COMMAND DISPLAY
 VFISCHED DISPLAY ARRAYS

    ARRAY   START ADDR    SIZE    # OF ENTRIES  % USED  NEXT FREE
   -------- ----------  --------  ------------  ------  ---------
   IRKA      0CD76E00    0004E200           75    1.50%  0CD780C0
   SEA/J     0CDD42C0    00030D40          132    1.32%  0CDD4D10
   SEA/JIX   0CDC55A0    0000EA60           13    0.86%  0CDCCAD0
   SEA/T     0CE052C0    00030D40          150    1.50%  0CE05E78
   SEA/TIX   0CDC54E0    000000C0            2    8.33%  ********

VFISCHED REQUEST SUCCESSFUL: DISPLAY ARRAYS
```

Figure 61. DISPLAY ARRAYS command example

```
UID: SPJE    CMD: VFISCHED DUMP IRKA
IZBR/CKK DUMP OF IRKA:          NEXT FREE=0CD780C0, LAST AVAILABLE=0CDC4FC0
  ADDRESS  RECID DSNAME                                        APPLID
  -------- ----- ---------------------------------------------- ------
  0CD76E00   C   CICSTS13.CICS.SDFHAUTH                        APPLI1
  0CD76E40   C   CICSTS13.CICS.SDFHLOAD                        APPLI1
  0CD76E80   C   GDDM.SADMMOD                                  APPLI1
  0CD76EC0   C   ISF.HASPINDX                                  APPLI1
  0CD76F00   C   ISP.SISPEXEC                                  APPLI1
  0CD76F40   C   ISP.SISPMENU                                  APPLI1
  0CD76F80   C   ISP.SISPSLIB                                  APPLI1
  0CD76FC0   C   ISP.SISPTENU                                  APPLI1
```

Figure 62. DUMP IRKA command example

Updating backup job information

VFISUB updates the Inventory record with backup job information, using the parameters described here. The JCL resides in CNTL(VFISUB).

You can use the following parameters when invoking VFISUB:

Table 33. Parameters for invoking VFISUB

Option	Parameter	Description
SMODE	I	Internal reader job submission
	T	TSO job submission
	N	No job submission (default)
JOBDSN	JOBDSN(MEMBER)	This parameter is used when SMODE=N to update only a specific jobname member, for example, BKUPJCL.DSS(APP1)
UMODE	G	Global update type
	A	Application update type

Files used by job VFISUB

The files listed here are used by the job VFISUB.

Table 34. Files used by VFISUB

File name	Description
SUBFILE	The submit file (DD DDIN) which is output by BKUPJCL with the application, the DSN to backup, the DSN of the backup job JCL and the size of the DSN to be backed up in megabytes (1024x1024 bytes).
SUBFILE2	A copy of the SUBFILE plus the submit date/time and the job name. It is used as input to the VFISUBR program.
READER	The internal reader file. If parameter SMODE=I was chosen, this requires the following DD card in JCL: //READER DD SYSOUT=(,INTRDR).
TSOSUB	The temporary TSO file if the parameter SMODE=T is selected.

```
//*-------------------------------------------------------------------
//* EXECUTE BRNVISUB TO SUBMIT JOB(S) AND/OR UPDATE SCHEDULING
//* INFORMATION IN THE INVENTRY DATABASE.
//*
//* NOTE: FOR SMODE=N, THIS STEP SHOULD BE THE FIRST STEP OF THE
//*             BACKUP JOB.
//*             FOR SMODE=N, JOBDSN PARAMETER SHOULD POINT TO WHERE
//*             THE BACKUP CONTROL CARDS RESIDE.
//*
//* NOTE: IF THIS STEP IS COMBINED W/BKUPJCLS, MAKE SURE THAT THE
//*             RELATIVE GENERATION FOR //SUBFILE REFLECTS (+1)
//*-------------------------------------------------------------------
//SUBMIT       EXEC PGM=IKJEFT01,REGION=0M,
//                  PARM=('BRNVISUB SMODE=N,UMODE=G',
//                   'JOBDSN=HLQ.BKUPJCL.DSS(APPL01)')
//*            PARM=('BRNVISUB SMODE=T,UMODE=G')
//STEPLIB      DD DSN=HLQ.SCKKLOAD,DISP=SHR
//             DD DSN=HLQ.SBRNLOAD,DISP=SHR
//CMDQUEUE     DD DISP=SHR,DSN=HLQ.RTCMDQ
//RTDPRINT     DD SYSOUT=*
//SYSTSPRT     DD SYSOUT=*
//SYSTSIN      DD DUMMY
//SUBFILE      DD DISP=SHR,
//             DSN=HLQ.SUBFILE(0)
//SUBFILE2     DD DSN=HLQ.SUBFILE2(+1),
//                DCB=(LRECL=160,BLKSIZE=0,RECFM=FB),FREE=CLOSE,
//                SPACE=(CYL,(1,1)),UNIT=SYSDA,VOL=SER=VVVVVV,
//                DISP=(NEW,CATLG,DELETE)
//*  UNCOMMENT FOLLOWING DD IF SUBMITTING VIA INTERNAL READER (SMODE=I)
//*READER      DD SYSOUT=(A,INTRDR)
//*  UNCOMMENT FOLLOWING DD IF SUBMITTING VIA TSO SUBMIT (SMODE=T)
//*TSOSUB      DD DSN=&&TSOSUB,
//*               UNIT=SYSDA,SPACE=(CYL,(1,1)),VOL=SER=VVVVVV,
//*               DISP=(NEW,DELETE,DELETE),
//*               DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
```

Figure 63. VFISUB JCL sample

Backup Dynamic Data set Scheduler reports

There are a number of reports available for the backup dynamic data set scheduler.

VFISUBR

VFISUBR reports on the SUBFILE2 data set from job VFISUB. It has a parameter that indicates the number of the most recent date/time entries desired. A parameter value of 0 means that all entries are desired; 0 is the default.

The following files are used by VFISUB:

- SUBFILE2 - The Submit file 2
- REPORT - The report

```
//*--------------------------------------------------------------------
//* EXECUTE BRNVSUBR TO REPORT ON SUBFILE2
//*--------------------------------------------------------------------
//REPORT       EXEC PGM=IKJEFT01,REGION=0M,
//             PARM=(BRNVSUBR)
//*            PARM=(BRNVSUBR,6)   /* WILL REPORT ON ONLY LAST 6 ENTRIES */
//STEPLIB      DD DSN=HLQ.SCKKLOAD,DISP=SHR
//                DD DSN=HLQ.SBRNLOAD,DISP=SHR
//USERMAC      DD DSN=HLQ.USERMAC,DISP=SHR
//SUBFILE2     DD DISP=SHR,
//                   DSN=HLQ.SUBFILE2(0)
//REPORT       DD SYSOUT=*
//SYSTSPRT     DD SYSOUT=*
//SYSOUT       DD SYSOUT=*
//SYSTSIN      DD DUMMY
//SYSUDUMP     DD SYSOUT=*
//SYSIN        DD DSN=&&SYSIN,DISP=(NEW,DELETE,DELETE),
//                UNIT=SYSDA,SPACE=(TRK,1),
//                DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=0
//SORTIN       DD DSN=&&SORTIN,DISP=(NEW,DELETE,DELETE),
//                UNIT=SYSDA,SPACE=(TRK,(5,5)),
//                DSORG=PS,RECFM=FB,LRECL=50,BLKSIZE=0
//SORTOUT      DD DSN=&&SORTOUT,DISP=(NEW,DELETE,DELETE),
//                UNIT=SYSDA,SPACE=(TRK,(5,5)),
//                DSORG=PS,RECFM=FB,LRECL=50,BLKSIZE=0
```

Figure 64. VFISUBR JCL sample

Backup Dynamic Data set Scheduling batch process

There are several steps required to enable the batch process for Backup Dynamic Data set Scheduling. This involves the submission of CNTL(BKUPJCLS) together with User-created Application Data set Backup JCL.

About this task

BKUPJCLS uses the control cards within the corresponding USERMAC member (eg USERMAC(ADSS)) to build the list of data sets requiring backup. This list is propagated to HLQ.BKUPJCL.backup method (for example, HLQ.BKUPJCL.DSS). Your application data set backup jobs access this list and submit application-related backups for each data set.

To set up the daily batch process for application backups using the Backup Dynamic Data set Scheduler:

Procedure

1. Update USERMAC member for site backup method (for example, ADSS) to contain only the following code:

```
//#SELECT
//:F INCLUDE (%DSNAME% -
//:O         (%DSNAME5 -
//:L         (%DSNAME% )
//:S INCLUDE (%DSNAME%)
//#ENDJCL
```

   Figure 65. Backup Dynamic Data set Scheduling Batch

2. Add CNTL(VFIMEMDL) JCL as the first step of CNTL(BKUPJCLS).

```
//*===================================================================
//*        DELETES ALL BACKUP CONTROL CARD MEMBERS IN THE
//*        BACKUPJCL.BKUPMETH PDS LIBRARY
//*
//*        YOU CAN CHANGE THE PARM ON THE EXEC STATEMENT TO
//*        DELETE ONE MEMBER NAME OR A MEMBER NAME MASK AS FOLLOWS:
//*
//*        PARM=(VFIMEMDL, 'BKUPCNTL GLOBAL01')  OR
//*        PARM=(VFIMEMDL, 'BKUPCNTL GLOBAL*')
//*
//*        IT IS RECOMMENDED THAT THE BKUPJCL.BKUPMETH PDS LIBRARY
//*        BE COPIED TO A GDG BACKUP IN A PRIOR STEP.
//*===================================================================
//PS05     EXEC PGM=INKJET01,REGION=0M,PARM=(BRNVFMMD)
//STEPLIB   DD  DSN=HLQ.SBRNLOAD,DISP=SHR
//SYSEXEC   DD  DSN=HLQ.SBRNSRCE,DISP=SHR
//BKUPCNTL  DD  DSN=HLQ.BKUPJCL.XXXXXXX,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSTSPRT  DD  SYSOUT=*
//SYSTIN    DD  DUMMY
```

Figure 66. VFIMEMDL JCL Sample

Note: Update the BKUPCNTL data set to reflect backup method for the site; for example, HLQ.BKUPJCL.DSS.

3. Create Application Data set Backup jobs for each application, containing the following sample code:

```
//*
//*=====================================================================
//*
//    SET APPLID=APPLICATION
//    SET BKPDSN=HLQ.DR.&APPLID..D&LYYMMDD..T&LHHMMSS..DSS001
//*
//*=====================================================================
//*   FUNCTION: DFDSS BACKUP OF APPLICATION CRITICAL FILES ON DASD.          =
//*=====================================================================
//*-- JS010: DFDSS BACKUP OF APPLICATION CRITICAL FILES ON DASD             --
//*--         HLQ.BKUPJCL.DSS(&APPLID)                                 --
//*           RC = 00 - MEMBER FOUND                                       --
//*           RC = 03 - MEMBER NOT FOUND                                   --
//*           RC = 04 - OTHER REASONS.                                     --
//*=====================================================================
//JS005 EXEC PGM=INKJET01,REGION=0M,PARM=(BRNVFMMC, '&APPLID')
//SYSEXEC    DD  DISP=SHR,DSN=HLQ.SBRNSRCE
//SYSTSPRT   DD  SYSOUT=*
//SYSTSIN    DD  DUMMY
//BKUPCNTL   DD  DSN=HLQ.BKUPJCL.DSS,DISP=SHR
//*
//IFJS005    IF (JS005.RC = 0  |  -JS005.RUN )  THEN
//JS010   EXEC XDSSBKUP,
//        APPLID=&APPLID,
//        BDSN=&BKPDSN
//PS10.VFIFILT  DD  DISP=SHR,DSN=HLQ.BKUPJCL.DSS(&APPLID)
//EIFJS005  ENDIF
```

Figure 67. Create Application Data set Backup jobs

Note: Ensure the APPLID is set to the relevant Application Name, as per your APPTABLE entry.

4. Change APPL01 in CNTL(VFISUB) to reflect the relevant Application Name:

```
//SUBMIT  EXEC PGM=IKJEFT01,REGION=0M
//        PARM=('BRNVISUB SMODE=N,UMODE=G',
//          'JOBDSN=HLQ.BKUPJCL.DSS(APPL01)')
//*       PARM=('BRNVISUB SMODE=T ,UMODE=G')
```

Figure 68. Change APPL01 in CNTL(VFISUB) to reflect the relevant Application Name

5. Add CNTL(VFISUB) JCL as first step of your Application Data set Backup jobs.

```
//SUBMIT   EXEC PGM=IKJEFT01,REGION=0M
//         PARM=('BRNVISUB SMODE=N,UMODE=G',
//            'JOBDSN=HLQ.BKUPJCL.DSS(APPL01)')
//*        PARM=('BRNVISUB SMODE=N,UMODE=A')
//*        PARM=('VFISUB SMODE=I,UMODE=G')
//STEPLIB  DD DSN=HLQ.SCKKLOAD,DISP=SHR
//         DD DSN=HLQ.SBRNLOAD,DISP=SHR
//CMDQUEUE DD DISP=SHR,DSN=HLQ.RTCMDQ
//RTDPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//SUBFILE  DD DISP=SHR,DSN=HLQ.SUBFILE(0)
//SUBFILE2 DD DSN=HLQ.SUBFILE2(=1),
//            DCB=(LRECL=160,BLKSIZE=0,RECFM=FB),FREE=CLOSE,
//            SPACE=(CYL,(1,1)),UNIT=SYSDA,VOL=SER=SW0006,
//            DISP=(NEW,CATLG,DELETE)
//* UNCOMMENT FOLLOWING DD IF SUBMITTING VIA INTERNAL READER (SMODE=I)
//*READER  DD SYSOUT=(A,INTRDR)
//* UNCOMMENT FOLLOWING DD IF SUBMITTING VIA TSO SUBMIT (SMODE=T)
//*TSOSUB  DD DSN=&&TSOSUB,
//*           UNIT=SYSDA,SPACE=(CYL,(1,1)),VOL=SER=SW0006,
//*           DISP=(NEW,DELETE,DELETE),
//*           DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//*
//*=============================================================================
//*
//   SET APPLID=APPLICATION
//   SET BKPDSN=HLQ.DR.&APPLID..D&LYYMMDD..T&lHHMMSS..DSS0001
//*
//*=============================================================================
//*      FUNCTION: DFDSS BACKUP OF APPLICATION CRITICAL FILES ON DASD.          =
//*=============================================================================
//*--     JS010: CHECKS FOR EXISTENCE OF BACKUP CONTROL CARD MEMBER             =
//*--            HLQ.BKUPJCL.DSS(&APPLID)                                  =
//*              RC = 00 - MEMBER FOUND                                         =
//*              RC = 03 - MEMBER NOT FOUND                                     =
//*              RC = 04 - OTHER REASONS.                                       =
//*=============================================================================
//JS005  EXEC    PGM=IKJEFT01,REGION=0M,PARM=(BRNVFMMC, '&APPLID')
//SYSEXEC  DD    DISP=SHR,DSN=HLQ.SBRNSRCE
//SYSTSPRT DD    SYSOUT=*
//SYSTSIN  DD    DUMMY
//BKUPCNTL DD    DSN=HLQ.BKUPJCL.DSS,DISP=SHR
//*
//IFJS005  IF (JS005.RC = 0  |  -,JS005.RUN )  THEN
//JS010    EXEC XDSSBKUP,
//         APPLID=&APPLID,
//         BDSN=&BKPDSN
//PS10.VFIFILT DD DISP=SHR,DSN=HLQ.BKUPJCL.DSS(&APLID)
//EIFJS005 ENDIF
```

Figure 69. Add CNTL(VFISUB) JCL

6. Ensure the Application Data set Backup JOBCLASS allows batch jobs to use System Symbols ($T JOBCLASS(x),SYSSYM=ALLOW)
7. Submit .CNTL(BKUPJCLS) at required intervals
8. Submit Application Data set Backup jobs for each Application following every run of BKUPJCLS.

Chapter 7. Monitoring Non-IBM Z Batch Resiliency Generated Backups

You can configure IBM Z Batch Resiliency to monitor backups generated by utilities such as IBM IDCAMS, IEBGENER, and IEBCOPY using the SCANSMF utility.

SCANSMF identifies critical files produced by these backup utilities and:

- Tracks current operational backups.
- Adds non-IBM Z Batch Resiliency backups to the BKUPTBL to enable recovery of all critical data sets
- Continually evaluates BKUPTBL entries
- Records the date and time of data sets that have changed since the last backup

Configuring IBM Z Batch Resiliency to monitor non-IBM Z Batch Resiliency generated backups

IBM Z Batch Resiliency can monitor non-IBM Z Batch Resiliency-generated backups.

About this task

To configure IBM Z Batch Resiliency to monitor the backups:

Procedure

Run the following jobs (located in CNTL) as part of a daily batch process:

LEGACY
Builds the LEGTABLE which defines the backup jobs to monitor.

SCANSMF
Scans the jobs listed in the LEGTABLE and adds entries in the backup table. By adding non-IBM Z Batch Resiliency backups, the backup table contains complete information for all backups and can be used to recover the data set, if needed.

(Optional) SCANALC and BKUPALC
For non-standard backup utilities (such as IBM IDCAMS, SORT, and IEBCOPY), adds allocation data to the backup table.

Building the LEGTABLE

The LEGTABLE resides in the USERMAC library. To update the LEGTABLE, run the CNTL(LEGACY) job. LEGACY adds entries to the LEGTABLE based on job records in the IZBR History file. The backup methodology programs are defined in USERMAC(XLEGACY).

About this task

The following figure shows the parameter statement within the LEGACY JCL.

```
       HLQ.CNTL(LEGACY)
//PS020    EXEC PGM=BRNVB051
//         PARM='NOCOMMENT,APPLID'
//*
```

Procedure

1. Uncomment the PARM= statement line in LEGACY by removing the leading asterisk (*).

© Copyright IBM Corp. 2019, 2020

2. Change the PARM= parameter in LEGACY as follows:
   a) For standard backup utilities, such as DSS or FDR, specify:
      `PARM='NOCOMMENT,APPLID'`
   b) For non-standard backup utilities, such as IBGENER or SORT, specify:
      `PARM='COMMENT,APPLID'`
3. Run the LEGACY job to build the LEGTABLE.

What to do next

After updating the LEGTABLE, use the RTA Command Processor IZBR ISPF Primary Panel - Option 7 to reload a new copy (REFRESH LEGTABLE) for the new contents to take effect.

LEGTABLE entries

This section describes the fields contained in USERMAC(LEGTABLE).

Table 35. LEGTABLE entries

Field	Columns	Description
Comment	1	When an asterisk (*) is in this field, the record is ignored.
Job name	2-9	The job to be scanned.
Step name	11-18	The name of the step to be scanned.
Relative Step #	20-22	The relative step number. Not used in SCANSMF analysis.
Program	24-31	The program to be scanned.
Condition Code	33-34	Highest possible condition code of the step to be considered successful.
APPLID	36-43	The application ID to assign to the scanned backup.
Input DD	45-52	Optional. Select only this DD if multiple inputs exist.
Output DD	54-61	Optional. Select only this DD if multiple outputs exist.

New Program	65-72	Replace program name with this name. Supported Program/New Program combinations are:
		`Program New Program`
		`ADRDSSU SMFDSSU`
		`FDR ZDR`
		`FDRDSF ZDRDSF`
		See Enabling the IBM Z Batch Resiliency SMF job validation exit (BRNVIUJV.) for more information.

The following is a sample LEGTABLE entry in USERMAC for job DV4008P:

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
 DV4008P  X5BKUP   10  IDCAMS   0  OMNI
```

The following is a sample entry for job DV5000P using program name replacement:

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
 DV5000P  STEP20   001 ADRDSSU  04 PAYROLL                    SMFDSSU
```

LEGACY utility

The LEGACY utility searches production SMF record type 30 for occurrences of programs that you want SCANSMF to monitor. Schedule LEGACY to run as a batch process to update LEGTABLE daily.

LEGACY can be controlled by execution parameter options and the USERMAC(XLEGACY) member. To limit which data set names are selected, you can specify filters using XLEGACY parameters.

The output of LEGACY consists of an updated LEGTABLE and LEGACY report in the REPORTS data set. After the new LEGACY report is generated, review the output to ensure that the SCANSMF process is monitoring the correct jobs and programs.

LEGACY utility parameter options

The output report order, and whether the LEGTABLE entry contains an asterisk (*) in column 1, is determined by the parameter value on the EXEC card.

Use the following parameters to specify reports by application (APPLID) or program. The options are COMMENT or NOCOMMENT:

- PARM='COMMENT,APPLID' - non-standard backup utilities
- PARM='NOCOMMENT,APPLID' - standard backup utilities
- PARM='COMMENT,PROGRAM' - non-standard backup utilities
- PARM='NOCOMMENT,PROGRAM' - standard backup utilities

LEGACY exit (XLEGACY)

The coding of the USERMAC(XLEGACY) determines what programs LEGACY incorporates into the LEGTABLE for subsequent SCANSMF runs.

XLEGACY parameter options

Simple XLEGACY example:

```
SMF30PGM = 'IEBGENER'
SMF30PGM = 'IEBCOPY'
SMF30PGM = 'IDCAMS
```

You can expand this exit with additional program names as required.

Use additional XLEGACY parameters to further refine the action of the LEGACY utility:

- An asterisk (*) in column 1 indicates a comment.
- Start SMF30PGM assignment starts in column 2.
- Complete SMF30PGM, OUTPUT, or FILTER assignment can span, at most, two lines.
- SMF30PGM, OUTPUT, or FILTER must appear in single quotes. For example,

  SMF30PGM = 'IEBGENER'
- To continue a definition over two lines (maximum), add a character in column 72; column 71 must be blank; continue the definition in column 5.
- OUTPUT – defines the backup program or mask used in the application jobs:
  - TAPE – only consider tape output records
  - DASD – only consider DASD output records
  - BOTH – applies to tape or DASD (default spaces=both)
- FILTER – use a DD statement within the CNTL(LEGACY) job that contains DSN masks to match (a match implies, keep the entry)
- NEWPGM – defines a new program that replaces the SMF30PGM.

- JOB – defines a job or job mask.
- STEP – defines a step or step mask.
- JOBINFO – option to capture backup job name in the BKUPTBL USERINFO field.

  Specify JOBINFO = 'Y' to place a backup job name in BKUPTBL.
- Entries are processed in order of appearance.
- You can assign up to 10 unique filters.

XLEGACY example with additional options:

```
SMF30PGM = 'ACPMAIN ' OUTPUT = 'BOTH' FILTER = 'DSNFILTR'
SMF30PGM = 'ADRDSSU ' OUTPUT = 'BOTH FILTER = 'DSNFILTR'
SMF30PGM = 'IEB* ' OUTPUT = 'BOTH'  NEWPGM = ‘REPLPROG'
```

Filter example JCL to be added to CNTL(LEGACY) job:

```
//DSNFILTR DD  *
*.BL2*.**
*.BU2*.**
```

Setting up SCANSMF

After you build the LEGTABLE, set up SCANSMF.

Procedure

1. Run the **CNTL(SSPALLOC)** job to allocate the SSPCKPT checkpoint data set.
2. In USERMAC(VFITABLS), uncomment the SCANSMF statements as shown below:

```
REPLMETH,PARMLIB,REPLMETH,BRNVL017   <SCANSMF REPLACE METHOD
REPLPGM,PARMLIB,REPLPGM,BRNVL017     <SCANSMF REPLACE PROGRAM…
EXCLDDNS,PARMLIB,EXCLDDNS,BRNVL017   <SCANSMF EXCLUDE DDNS
LEGTABLE,PARMLIB,LEGTABLE,BRNVL017   <SCANSFM LEGTABLE
```

Figure 70. VFITABLS member

The following are the members for VFITABLS:

REPLMETH
Updates the backup method field (limited to 7 characters).

REPLPGM
IDCAMS and IEBCOPY have special sequencing requirements. If your EXEC PGM= specifies a replacement module for either, define that relationship here (for example, EXEC PGM=DRIDCAMS).

EXCLDDNS
Eliminates DDs for SCANSMF processing.

LEGTABLE
Contains entries of job records based on the SMF history file.

3. Define the high-level qualifier in USERMAC member RTSSPPRM for the SCANSMF process output data sets. The HLQ must be less than or equal to 28 characters.

```
SSPPFX=hlq.SCANSMF
SSPLPRI=20
SSPLSEC=20
SSPVOL=TST001
SSPUNIT=3390
```

Figure 71. RTSSPPRM member

4. Optional: For non-standard backup utilities (such as IBM IDCAMS, SORT, and IEBCOPY), run the SCANALC and BKUPALC jobs to add allocation data to the Backup Table as described in Adding data to the Backup Table for non-standard utilities.
5. Optional: If your installation uses backup utilities such as IBM's DF/DSS and Innovation's FDR, which do not generate SMF records for input data sets, see “Setting up SCANSMF for utilities that do not produce SMF records” on page 118 for additional steps in the daily SCANSMF process.
6. Set up SCANSMF to run continuously as a subtask of the RT/SMF Dispatcher or schedule it to run periodically as a batch job, as described in the following tasks:
   - Activating the Real-Time SCANSMF subtask
   - Running the SCANSMF batch job

Activating the Real-Time SCANSMF subtask

You can set up SCANSMF to run continuously as a subtask of the RT/SMF Dispatcher. The SCANSMF subtask scans the backup jobs listed in the LEGTABLE and then updates the backup table with details about non-IBM Z Batch Resiliency backups.

Before you begin

“Building the LEGTABLE” on page 111 and “Setting up SCANSMF” on page 114

About this task

To add SCANSMF as a subtask of RT/SMF:

Procedure

1. Uncomment the following DD statement in the CNTL(RTSMFP) member:
   ```
   //SSPCKPT DD DISP=SHR,DSN=hlq.SSPCKPT
   ```
2. Uncomment the START TASK statement in the USERMAC(VFISTASK) member:
   ```
   START TASK=SCANSMF,PGM=BRNVC031,SMF=(14,15,30,64,202)
   ```
3. Confirm that SMF record type 202 is defined for the SMF= parameter in VFISTASK, as shown in the above example. If SMF record type 202 is already in use, add another record type as follows:
   a) In the USERMAC(SMFDSSUO) member, change SMFNUM=202 to the SMF record type provided by your systems programming group.
   b) Run CNTL(SMFDSSUO) to create the SMF record number.
   c) Update the SMF record type in VFIANLY1 statement in USERMAC(VFISTASK) and **CNTL(SMFPsys)**.
4. Uncomment the following statements in USERMAC(VFITABLS):
   ```
   REPLMETH,PARMLIB,REPLMETH,BRNVL017   < SCANSMF REPLACE METHOD
   REPLPGM,PARMLIB,REPLPGM,BRNVL017     < SCANSMF REPLACE PROGRAM
   EXCLDDNS,PARMLIB,EXCLDDNS,BRNVL017   < SCANSMF EXCLUDE DDNS
   LEGTABLE,PARMLIB,LEGTABLE,BRNVL017   < SCANSMF LEGTABLE
   ```

   When RT/SMF is restarted, the SCANSMF subtask begins monitoring the defined non-IBM Z Batch Resiliency backup utilities.

Running the SCANSMF batch job

You can run SCANSMF as a batch process, to scan the jobs listed in the LEGTABLE and update the backup table with details about the non-IBM Z Batch Resiliency backups.

Before you begin

Be sure to complete the following tasks before you begin:

- “Building the LEGTABLE” on page 111
- “Setting up SCANSMF” on page 114

Procedure

In CNTL(SCANSMF), add PARM=UPDATE to run in update mode:

```
//PS020 EXEC PGM=BRNVB052,PARM=UPDATE
```

The default input for the SCANSMF job is the IBM Z Batch Resiliency History file. If you want to scan another input source (such as a History backup file or the Audit file), change the SORTIN statement in the job as necessary.

Adding data to the Backup Table for non-standard utilities

When SCANSMF program adds information to the backup table for non-IBM Z Batch Resiliency generated backups, data set allocation information required by RESTJCL is not included. Backup utilities SORT, IDCAMS, and IEBCOPY do not maintain this kind of data.

About this task

Note: If your backup utility (such as DF/DSS) maintains data set attributes with the backed-up files, SCANALC and BKUPALC are not required.

Procedure

To add allocation information for these backup utilities, run SCANALC and BKUPALC as a daily batch job.

- **SCANALC** – obtains data set names from existing backup table entries of methodologies specified in the member USERMAC(XSCANALC) and builds the USERMAC(INCLALC) member that is used by BKUPALC.
- **BKUPALC** – reads a list of data set names in USERMAC(INCLALC) and obtains allocation data for those data sets. It then updates the backup table with additional entries using a special backup method named 'ALLOCAT'. The following types of data sets can be added, along with a short comment:
  - **GDG BASEs** – All pertinent attributes. (SMS data is not applicable.)
  - **NON-VSAMs** – All pertinent attributes of cataloged DASD data sets (except HFS files). SMS data is recorded. 40 VOLSERS per data set are possible.
  - **CLUSTERs** – All pertinent attributes of the four kinds: KSDS, RRDS, ESDS and Linear. SMS data is recorded. 10 VOLSERs per the Data component and 6 VOLSERs per Index component are possible.
  - **AIXs** – All pertinent attributes. (SMS data is not recorded.) 10 VOLSERs per the Data component and 6 VOLSERs per Index component are possible.
  - **PATHs** – The related cluster name is recorded. (SMS data in not applicable.)
  - **PAGESPACEs** – All pertinent attributes. SMS data is recorded.

Selecting specific data sets (INCLALC)

To select specific data sets, edit the USERMAC(INCLALC) member.

About this task

When editing INCLALC, follow these rules:

- The format of a parameter input INCLALC is:

  application_name,*dataset_name*[, *dataset_name_2*,...]
- A slash and asterisk (/*) in column one is a comment and is not processed. Blank lines are not processed.

- Each input consists of two positional parameters:
  - *application_name* – consists of one to eight characters, with no embedded blanks or commas and is a required parameter.
  - *dataset_name* – Required. Up to 44 bytes (the standard for a DSN). You can code as many data set names as you want. Use commas to separate the parameters. Use a comma to continue the parameters onto another line.
    - Example 1: The GL Application, a single data set: `GL,GL.TRANS.FILE`
    - Example 2: The PAYROLL Application, two data sets, using continuation: `PAYROLL,PAY.TRANS.FILE, PAY.MASTER.FILE`
- Accepted or rejected INCLALC lines are noted in the SYSTSPRT DD SYSOUT.

When BKUPALC runs, all allocation information for the data sets indicated in INCLALC is inserted into the BKUPTBL and BKUPAUD files. A report is generated and placed into the REPORTS file under member name BKUPALC.

Note:

Allocation data is not produced for migrated data sets; these data sets are skipped. The default migration volume name is MIGRAT. If your site uses another migration volume name, you must update the BKUPALC job with a parameter to indicate that new name. Here is a sample:

The EXEC line of the BKUPALC job looks like this:

```
//STEP1 EXEC PGM=IKJEFT01,REGION=0M,PARM=(BKUPALC)
```

Replace it with this:

```
//STEP1 EXEC PGM=IKJEFT01,REGION=0M,PARM=(BKUPALC,'999999')
```

where 999999 is your migration volume VOLSER.

Activating SCANSMF diagnostic output

The SCANSMF job or subtask prints basic information about its operation while evaluating SMF records for possible inclusion in the backup table. If data is not written to the backup table for data sets from a job that you expect to be included, you can obtain detailed information about SCANSMF's analysis of the given job. The SCANSMF subtask has diagnostic support for both RTA and batch processing for either a single job or for all jobs.

About this task

To activate SCANSMF diagnostics, use the RT/SMF Command Processor via the IZBR ISPF Primary Panel Option - Option 7 Miscellaneous, Option 1 RT/SMF Commands.

Procedure

1. To turn on support for all jobs that match a LEGTABLE job, specify:

   ```
   DIAGNOSE SCANSMF
   ```

   The output is sent to the RTDPRINT DD statement, which can produce a large amount of data.
2. To activate SCANSMF diagnostic support for job name PRODJOB1 only, specify:

   ```
   DIAGNOSE SCANSMF,RSC=PRODJOB1
   ```

   See message ‘BRNVC05310I’ in Appendix H, “IBM Z Batch Resiliency Messages,” on page 263 within this guide for processing results.

3. To deactivate SCANSMF diagnostics, specify:

```
DIAGNOSE SCANSMF,OFF
```

Setting up SCANSMF for utilities that do not produce SMF records

If you use a backup utility that does not produce SMF records, SCANSMF can capture backup information from data movers, such as DF/DSS and FDR, which do not explicitly open the input data sets that are being backed up.

Before you begin

Before you set up SCANSMF to produce SMF records, verify the following:

- APF authorize *hlq*.DIST.LIBRARY.
- Check the *hlq*.DIST.LIBRARY for the required modules:
  - **SMFDSSU** – invokes designated data mover and creates SMF record type 202 (or another site-determined SMF type).
  - **SMFDSSUO** – defines the options table for SMFDSSU.
  - **VDUOPT** – loads option modules, using the LOAD macro, for SMFDSSU:
  - **SMFDSSUO** – options for SMFDSSU.
  - **SMFFDRO** – options for FDR and FDRDSF. After a successful LOAD, the options values in effect are displayed to DD:VDUPRINT. If DD:VDUPRINT is not allocated, it is set to SYSOUT=*.
  - **BRNVYNEX** – loads the LEGACY table loader and BRNVIUJV exit manager.
  - **BRNVLEGX** – builds the LEGACY table for BRNVYNEX.
  - **BRNVRAD** – builds the SMF data for files in backup.
  - **BRNVIUJV** – modifies execution program name using.
- For the FDR utility only, enable the Disk Volume Open exit and a Data Set Security exit. If these exits are not already enabled, run the FDRZAPOP step (described below). Do not place the installation exits in the link pack area (LPA).

Procedure

1. To indicate to the BRNVIUJV exit which programs to track, code the replacement program SMFDSSU in column 65 of the LEGTABLE for every ADRDSSU program execution that you want BKUPTBL to monitor for inclusion by SCANSMF. Make equivalent additions for other programs (like FDR) that are replaced. For example, see the final three entries in this excerpt of a sample LEGTABLE:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
 #WEEKLY  JS30       3 IEBGENER  4 TESTAPP
 #BACKUP  JS10       1 ICEGENER  4 TESTAPP
 #BACKUP  JS50       4 IDCAMS    4 TESTAPP
 #BACKUP  JS60       5 ADRDSSU   4 TESTAPP                      SMFDSSU
 SDJSDSST PS10       1 ADRDSSU   4 TESTAPP                      SMFDSSU
 SDJSTST  JS10       1 ADRDSSU   4 TESTAPP                      SMFDSSU
```

Figure 72. Sample LEGTABLE

2. Optional: For FDR only, modify the FDR options module (FDROPT) to enable these exits:
   - Disk Volume Open exit
   - Data Set Security exit

   Note: Skip this step if your installation already uses these exits. Your versions of these exits preload at run time by SMFFDR and the installation exits are run before the IBM Z Batch Resiliency versions of these exits.

   Sample JCL to Enable the Exits

Note: Do not place these installation exits in the link pack area (LPA).

```
//FDRZAPOL EXEC PGM=FDRZAPOP
//SYSPRINT DD   SYSOUT=*
//STEPLIB  DD   DISP=SHR,DSN=FDR.FDR5465.MODFDR5465
//SYSLIB   DD   DISP=SHR,DSN=HLQ.APF.LIBRARY
//SYSIN    DD   *
  CKKIZAP ENABLE=(OPENEXIT,SECEXIT),OPENEXIT=BRNVFFXO,SECEXIT=VFIFXD
/*
```

Figure 73. Sample FDRZAPOP to Enable Exits

Program	Description
STEPLIB	Library that contains module FDRZAPOP.
SYSLIB	Library that contains the FDROPT module.
BRNVFFXO	IBM Z Batch Resiliency-suggested name for the Volume Open exit.
VFIFXD	IBM Z Batch Resiliency-suggested name for the Security exit.

3. For each system running the SMF extractor, add SMF record type 202 to the **SMFREC=** parameter in CNTL(SMFP*sysname*), where *sysname* is replaced by the system name on which the SMF extractor is running.

   ```
   SMFREC=14,15,30,60,61,64,65,202
   ```

   This captures the SMF type 202 records that are output from the SMFDSSU.
4. For RTA processing, add SMF record type 202 to the SMF= parameter in (USERMAC)VFISTASK:

   ```
   START TASK=BRNVANLY1,PGM=BRNVC001,SMF=(14,15,30,61,64,65,202)
   ```

Results

After type 202 records have been collected, they are ready to be processed by the SCANSMF utility.

How SCANSMF produces SMF records for data movers

The IBM Z Batch Resiliency SCANSMF process can capture backup information from data movers such as DF/DSS and FDR that do not explicitly open the input data sets that are being backed up.

To produce SMF records, the IBM Z Batch Resiliency program SMFDSSU calls original programs ADRDSSU, FDR, or FDRDSF, and produces SMF type 202 records to record the input and output data sets. Both physical and logical dumps are supported for ADRDSSU and FDR. SMF records are written for each volume and each data set within a logical dump. For physical dumps, SMF records are written for the volume for program ADRDSSU and for the volume and optionally each data set for the FDR programs. The SMF data can be written to the system MAN files or, in the same format, to a separate file using the DDName SMFDSSUS. Alternatively, to allow SMF data generation after the backups have been completed, use the BRNVRAD module, to read an existing ADRDSSU dump and generate SMF records. Generating SMF data from FDR dumps is not supported.

To facilitate this process, there is an optional exit and started task available to modify the executing program in the JCL. This can eliminate the need to modify all JCL at once, and also eases implementation and back out. The facility changes EXEC PGM=ADRDSSU to EXEC PGM=SMFDSSU, EXEC PGM=FDR to EXEC PGM=ZDR or EXEC PGM=FDRDSF to EXEC PGM=ZDRDSF depending on the backup data mover you use.

This process is supported by the following programs:

- **BRNVYNEX** – A started task that loads the table that associates the EXEC PGM= replacements and is the system exit SYS.IEFUJV manager. The module must be linked, authorized and run from an authorized library.

- **BRNVIUJV** – The function of this module is to change the program name on the EXEC JCL statement for jobs that are defined in a table (built by BRNVLEGX – see below) created by name/token services. This table, which contains the Job name, Step name, old program and new program, is retrieved using name/token services and if a match occurs then the program name is modified on the EXEC JCL statement from old program to new program. It is an SMF Job Validation Exit.
- **BRNVLEGX** – This module reads the sorted LEGACY file and builds a header record and 400-byte entry records that associate job, step, and program information used to replace the executing program in the JCL. This is the information used by BRNVIUJV to replace program names. See the example in LEGTABLE entries that demonstrates the required coding for program name replacement.

Modifying interface modules for ADRDSSU or FDR/FDRDSF

If the options provided in the distribution do not meet your requirements, you can modify the installation options module. The same macro is used for both option modules, but some parameters might be present that are not used by the interface module.

About this task

IBM Z Batch Resiliency provides the following two interface modules:

- **SMFDSSUO** – options module for the ADRDSSU interface
- **SMFFDRO** – options module for the FDR/FDRDSF interface

Examples of options modules are provided in the USERMAC library.

Procedure

1. Assemble and link edit JCL the following jobs in the CNTL library:
   - ASMDSSUO to build the SMFDSSUO module
   - ASMFDRO to build the FDROPTO module
2. Compile these options modules into the same library where the main modules reside.

Table 36. ADRDSSU options

Parameter	Value	Description
ADRDSSU	ADRDSSU	Name of the ADRDSSU module.
TRACE	No	Do not produce a trace data set
DSN		Produce a trace data set in file SMFDSSUT if it is allocated otherwise send the trace data to SYSOUT=*.
(SYSOUT,class)		Send the trace data to file SMFDSSUT if it is allocated, otherwise send the trace data to SYSOUT=class.
SMFNUM	202	The SMF record type to write in the range 128-255. 202 is optimal because it is recognized by the SMF extractor and the Analysis process.
SMF	BOTH	Write SMF records to the active MAN data set and to the SMFDSSUS file.
SMFWTM		Write SMF records to the active MAN data set even if file SMFDSSUS is allocated.
DSN		Write SMF records to file SMFDSSUS only.
AUTO		Write SMF records to file SMFDSSUS if it allocated otherwise write them to the active MAN data set.

Table 36. ADRDSSU options (continued)

Parameter	Value	Description
X37	SUPPRESS	Do not generate a dump when file SMFDSSUS fills or becomes unusable. ADRDSSU execution continues.
ON		Do not suppress the dump. ADRDSSU execution continues.
ADRRCNZ	NOSMF	If ADRDSSU ends with a return code greater than ADRRCMX, do not write SMF records. This parameter applies to options module SMFDSSUO only.
SMF		Write SMF records even if ADRDSSU ends with a return code greater than ADRRCMX. This parameter applies to options module SMFDSSUO only.
FDRRCNZ	NOSMF	If SMFFDR processing ends with a return code greater than FDRRCMX, do not write SMF records. This option applies to options module SMFFDRO only.
SMF		Write SMF records even if SMFFDR processing ends with a return code greater than FDRRCMX. This option applies to options module SMFFDRO only.
ADRRCMX	N (a number), 4	Write SMF records if the return code is less than this value depending on the value of ADRRCNZ. The default value is 4. This parameter applies to options module SMFDSSUO only.
FDRRCMX	N (a number), 4	Write SMF records if the return code is less than this value depending on the value of FDRRCNZ. The default value is 4. This parameter applies to options module SMFFDRO only.
PACESMF	YES	If a write to the MAN data set returns a code indicating that data was lost, or there is an SMF buffer shortage, or SMF could not establish recovery, retry the write using the PACECNT and PACINT values below.
NO		Ignore the error and continue.
PACECNT	N (a number), 5	If the PACESMF value is YES, repeat the attempt to write the SMF records N times. The default value is 5.
PACEINT	N (a number), 10	If the PACESMF value is YES, wait N 100ths of a second before trying to write the SMF record again. The default value is 10.
MSGLVL	N (a number between 0 and 9), 1	The parameter applies to the TRACE data set and should only be changed from 1 at the prompting of Technical Support. 0 is the most restrictive value and 9 is the least restrictive value.
CRY	YES	Logic to process hardware compression and encryption records and flags is used. This is only available on z/OS 1.8 and later releases.
NO		Logic to process hardware compression and encryption records and flags is bypassed.
PHYSVLD	DSN	Generate SMF records for data sets dumped by physical full volume requests.

Table 36. ADRDSSU options (continued)		
Parameter	**Value**	**Description**
	NODSN	Generate only the VOLUME SMF records for physical full volume dump requests.
U0888	NOSMF	If SMFFDR processing ends with return code U0888, abend this process and do not write any SMF records.
SMF		Write SMF records even if the SMFFDR process ends with return code U0888.

Tracking backups (SMFDSSU)

About this task

To run SMFDSSU to track backups, make the following changes:

Procedure

1. Replace PGM=ADRDSSU with PGM=SMFDSSU in the backup execution JCL.
2. Optional: Override the writing of SMF records to the system MAN files by adding the following DD statement to the backup JCL:

   ```
   //SMFDSSUS DD  DSN=yourhlq.SMFDSSU.DUMPSMF,
   //             DISP=(NEW,CATLG,DELETE),
   //             UNIT=SYSDA,
   //             SPACE=(TRK,(60,15),RLSE),
   //             RECFM=U,LRECL=0,BLKSIZE=27998
   ```

   For physical dumps, SMF records are written for the volume for program ADRDSSU and for the volume, and optionally each data set, for the FDR programs. The SMF data can be written to the system MAN files, or in the same format, to a separate file using the DDName SMFDSSUS.
3. To make dynamic changes, without modifying the JCL, enable the BRNVYNEX dynamic exit to intercept the production JCL for specific programs, and allow creation of the SMF records needed to track the backup. See Enabling the BRNVYNEX dynamic exit.

   When SMFDSSU is executed, the FDR-produced DD:SYSPRINT statements are not changed, but a separate trace DD:SMFFDRT is created. Trace information is written to file SMFDSSUT.

Listing options (VDUOPT)

You can list the options from the SMFDSSUO/SMFFDRO user options modules as needed.

About this task

To list the options from the

SMFDSSUO/

SMFFDRO user options modules:

Procedure

Use program VDUOPT to load the option modules and then display the options within them.

```
//VDUOPT  EXEC PGM=VDUOPT
//STEPLIB  DD  DISP=SHR,DSN=hlq.APF.LIBRARY
//VDUPRINT DD  SYSOUT=*
```

Results

If file VDUPRINT is not pre-allocated, the module dynamically allocates it.

Enabling the VFIDYNEX dynamic exit

VFIDYNEX is the SYS.IEFUJV exit manager in addition to loading the LEGFILE file. The LEGFILE is built by program BRNVLEGX from the LEGACY file to assign replace program cards. The edited LEGFILE is loaded into extended CSA for the exit program to use during JCL processing to identify job steps where the executing program should be changed to SMFDSSU.

About this task

Best practice is to run this module as a started task, although you can also run it as a job.

```
//VFIDYNEX EXEC PGM=VFIDYNEX,PARM='DSN=library that contains module BRNVIUJV'
//STEPLIB  DD  DISP=SHR,DSN=<authorized library>
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//INFILE   DD  DISP=SHR,DSN=yourhlq.USERMAC(LEGFILE)
```

Figure 74. VFIDYNEX JCL Example

Procedure

1. Copy the PARM for VFIDYNEX started task in the library that contains the module BRNVIUJV. This can be the same as the library specified in the STEPLIB.
2. Ensure STEPLIB is an authorized library.
3. If the LEGFILE is AG.V420.LEGFILE, the card in the LEGFILE member in USERMAC is:

   ```
   AG.V4X0.LEGFILE
   ```

   Note: You can use the sample JCL stream in the IBM Z Batch Resiliency CNTL data set. Define an authorized STEPLIB and modify the PARM value to specify the data set that contains the BRNVIUJV module.
4. To configure the VFIDYNEX started task so that it cannot be canceled, add the following PPT (Program Properties Table) entry to your SCHEDXX PARMLIB:

   ```
   12345678901234567890123456789012345678901234567890123456789012345678901234567890
            1         2         3         4         5         6
   ....+....0....+....0....+....0....+....0....+....0....+....0....+....0..

   PPT  PGMNAME(VFIDYNEX) /* VFIDYNEX                          */
           NOCANCEL       /* CANNOT BE CANCELLED               */
           SWAP           /* PROGRAM IS SWAPPABLE              */
           DSI            /* REQUIRES DATA SET INTEGRITY       */
           PASS           /* NOT ALLOWED TO BYPASS PASS PROT   */
           SYST           /* SYSTEM TASK SO NOT TIMED          */
           AFF(NONE)      /* NO PROCESSOR AFFINITY             */
           NOPREF         /* NO PREFERRED STORAGE FRAMES       */
   ```

Stopping or reloading the BRNVYNEX task or job

To stop or reload BRNVYNEX, use the STOP and RELOAD commands.

About this task

Procedure

1. Enter the commands from the console as follows for each function:

```
P BRNVYNEX
F BRNVYNEX,STOP
```

The two commands above stop the task or job. As part of this process, the exit program BRNVIUJV is terminated and the ECSA storage is released.

2. To reload the LEGFILE and release the old ECSA table storage, use the RELOAD command.

```
F BRNVYNEX,RELOAD
```

Any other command text return an error message.

Enabling the IBM Z Batch Resiliency SMF job validation exit (BRNVIUJV)

BRNVIUJV is the IBM Z Batch Resiliency version of the SMF Job Validation Exit (IEFUJV) that receives control before interpreting each job control (JCL) statement in the input stream.

About this task

Procedure

1. To use BRNVIUJV, define the IEFUJV installation exit in SMFPRMxx.
2. To activate and deactivate the BRNVIUJV exit routine, use the BRNVYNEX procedure.

   BRNVIUJV gains control after any IEFUJV exit routines that are already defined at your installation.

Results

The BRNVIUJV exit routine works in conjunction with programs BRNVYNEX and BRNVLEGX to dynamically alter the program name on a // EXEC PGM statement.

When BRNVIUJV encounters a // EXEC PGM statement, it reads a table created by the BRNVLEGX program and loaded into memory by BRNVYNEX. This table consists of entries which define a job name, step name, old program name and new program name. When the JCL statement being processed matches one of the entries in the table, BRNVIUJV alters the program name on the // EXEC PGM statement from old program name to new program name.

BRNVIUJV should always issue return code zero.

Defining the IEFUJV exit

Before you begin

Before defining the IEFUJV exit:

- Install the BRNVLEGX and BRNVYNEX programs.
- Ensure BRNVIUJV resides in the same library as BRNVLEGX and BRNVYNEX.

About this task

To define the IEFUJV exit:

Procedure

1. Specify IEFUJV on the EXITS option of the SYS parameter in the SMF PARMLIB member (SMFPRMxx).

The following is a sample of the SYS section of an SMFPRMxx member that contains the IEFUJV exit (the other parameters may differ from the ones shown here):

```
SYS(TYPE(00:41,43:98,100:255),EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,
     IEFUJV,IEFUSI,IEFUJP,IEFUSO,IEFUJI,IEFUTL,IEFU29,IEFUAV),
     INTERVAL(SMF,SYNC),DETAIL)
```

For more information on defining IEFUJV, see the description of SMFPRMxx in the *IBM z/OS Initialization and Tuning Reference* .

2. To display the active SMF options in effect, issue the system command:

   ```
   D SMF,O
   ```

3. To activate the BRNVIUJV exit routine, start the BRNVYNEX procedure on your system.

Results

When the BRNVYNEX procedure is terminated, the BRNVIUJV exit routine is deactivated.

Validating Legacy file program names (BRNVLEGX)

About this task

The BRNVLEGX module takes the sorted legacy file and validates the old and new program names. If all of the entries are good, the LEGFILE is built for use by the exit program BRNVIUJV. A non-zero return code indicates an invalid condition. You can check the REPORT output for diagnostic information. JCL for BRNVLEGX is located in your IBM Z Batch Resiliency CNTL file in member BRNVLEGX.

Procedure

Check the REPORT output for diagnostic information.

JCL for BRNVLEGX is located in your IBM Z Batch Resiliency CNTL file in member BRNVLEGX.

Adding DSS or FDR backup entries to BKUPTBL

IBM Z Batch Resiliency provides two utilities for adding DSS or FDR backup entries to the BKUPTBL.

The two utilities are:

- VFIFDSS – Adds DSS full volume backup entries to BKUPTBL
- BRNVIFDR – Adds FDR backup entries to BKUPTBL

Adding DSS full volume backup entries to BKUPTBL (VFIFDSS)

Use the VFIFDSS utility to track files backed up by ADRDSSU full volume DUMP process and record backup information for these files in the IBM Z Batch Resiliency BKUPTBL for subsequent retrieval. Having the backup information in the IBM Z Batch Resiliency BKUPTBL lets you restore a file which was backed up by DSS full volume DUMP utility if it becomes corrupted. You can use the ISPF panel INQUIRY functions to restore these files.

The VFIFDSS program uses one of three possible input sources to record backup information for files which were backed up by the ADRDSSU full volume DUMP utility. This is based on which JCL DD statement is specified during the execution of the VFIFDSS program as follows:

- If TAPE1 DD statement is present and specifies the output backup data set from the ADRDSSU full volume DUMP, VFIFDSS dynamically invokes ADRDSSU for a TYPRUN=NORUN RESTORE request and parse the output to obtain the files that were backed up. The IBM Z Batch Resiliency CNTL library member VFIFDSS contains an example of a JCL procedure with steps that execute ADRDSSU full volume DUMP, followed by steps that execute VFIFDSS, which uses the output backup data set allocated in the TAPE1 DD statement as its input source. The first step (PS010) in the sample JCL is the actual DSS full volume backup step. If you have your own JCL for the DSS full volume backup process,

the remaining steps (PS020 through PS030) should be added to your JCL to take advantage of the VFIFDSS utility's functions.

- If DSSPRINT DD statement is present, VFIFDSS assumes that an ADRDSSU PRINT was issued along with the full volume backup and the output was captured in a data set, as specified by the DSSPRINT DD statement. VFIFDSS uses the output from the ADRDSSU PRINT request to obtain the files that were backed up. The IBM Z Batch Resiliency CNTL library member BRNVFFDSSP contains an example of a JCL procedure with steps that execute VFIFDSS which uses the output from ADRDSSU PRINT as specified by the DSSPRINT DD statement as its input source.
- If DCOLLECT DD statement is present, VFIFDSS assumes an IDCAMS DCOLLECT was issued for the DASD volume of the full volume backup and the output was captured in a data set, as specified by the DCOLLECT DD statement. VFIFDSS uses the output from the DCOLLECT to obtain the files that were backed up. The IBM Z Batch Resiliency CNTL library member BRNVFFDSSV contains an example of a JCL procedure with steps that execute VFIFDSS, which uses the output from IDCAMS DCOLLECT as specified by the DCOLLECT DD statement as its input source.

The VFIFDSS program records backup information for all files that were backed up in the DSS full volume step. If you want to record backup information only for the files that are critical to IBM Z Batch Resiliency, remove the comment for the //*CRITICAL DD JCL statement found in the sample JCL procedures and verify that it points to the production IBM Z Batch Resiliency CRITICAL data set.

VFIFDSS Parameters

You can specify execution parameters for BRNVIDSS on the EXEC statement in the JCL. You can also specify these parameters in a file which is referenced in the JCL by USERPARM DD statement. Parameters on the EXEC statement in the JCL override the same parameter if it is also specified in the file referenced by the //USERPARM DD statement.

The following table describes the VFIFDSS execution parameters and default values:

Table 37. VFIFDSS parameters

Parameter	Description	Default value
APPLID=	Application ID assigned to record in BKUPTBL.	FULLDSS
DASDVOL=	Volume serial of the DASD volume of the full volume backup. This parameter is required.	None
BKUPDD=	DDNAME of DSS full volume backup file.	TAPE1
BDSNAME=	Backup DSNAME of full volume backup file. This parameter is used for DSSPRINT and DCOLLECT input source only and is required.	
BDSDATEQ=	This parameter is used for DSSPRINT input source only and is 1 to 3 characters that, along with the date (YYDDD) in ADR006I message from DSSPRINT, is appended to BDSNAME to get the full BDSNAME value. For example, if BDSNAME=PROD.BACKUP, BDSDATEQ=D and the date in ADR006I is 16100, BDSAME would have the value PROD.BACKUP.D16100.	
ERRORRC=	Return code with which VFIFDSS is forced to end for any error condition. If not specified, then VFIFDSS ends with a different return code for different error conditions.	
EXCLUCAT=	This parameter is used for TAPE1 input source. A user catalog DSNAME mask may be specified so that an additional EXCLUDE card is generated for ADRDSSU RESTORE TYPRUN=NORUN request to exclude user catalogs matching this DSNAME mask. This may be used if the name of a user catalog is other than what is automatically excluded which is CATALOG.** and USERCAT.**	

Adding FDR backup entries to BKUPTBL

The BRNVIFDR utility allows IBM Z Batch Resiliency to track files backed up by FDR full volume and FDR/ABR incremental processes and record backup information for these files in the BKUPTBL for subsequent retrieval. Having the backup information in the BKUPTBL allows you to restore a file backed up by FDR, if it becomes corrupted. You can use the ISPF panel INQUIRY functions to restore these files.

About this task

The CNTL library contains two samples of JCL procedures which execute the BRNVIFDR utility to record backup information in the BKUPTBL:

- **BRNVIFDRF** – FDR full volume backups
- **BRNVIFDRI** – FDR/ABR incremental backups

The first step (PS010) of the BRNVIFDRF procedure is the actual FDR full volume backup step.

Procedure

1. If you have your own JCL for the FDR full volume backup process, add the remaining steps (PS020 through PS050) to your JCL to take advantage of the BRNVIFDR utility's functions.
2. Modify your FDR full volume backup JCL by changing the //SYSPRINT DD statement in the backup step as follows:

   ```
   //SYSPRINT DD DISP=(,PASS),DSN=&&FDRPRINT,
   //            UNIT=SYSDA,SPACE=(CYL,(1,1))
   ```

   This is done so that the BRNVIFDR program executed in step PS030 can read the FDR output from the FDR backup step and gather the necessary information to update the BKUPTBL.

   The BRNVIFDR program in step PS030 records backup information for all files that were backed up in the FDR full volume step.
3. Optional: If you want to record backup information only for the files that are critical to IBM Z Batch Resiliency, remove the comment for the `//*CRITICAL DD JCL` statement in step PS030 and verify that it points to the production IBM Z Batch Resiliency critical file.

BRNVIFDR parameters

You can specify execution parameters for BRNVIFDR on the EXEC statement in the JCL. You can also specify these parameters in a file which is pointed to in the JCL by DD statement USERPARM. Parameters on the EXEC statement in the JCL override the same parameter if it is also specified in the file referenced by the `//USERPARM DD` statement.

The following table describes the BRNVIFDR execution parameters and default values:

Table 38. BRNVIFDR execution parameters and default values

Parameter	Description	Default value
APPLID=	Application ID assigned to record in BKUPTBL	FDRFULL
FDRTYPE=FULLFDR	FDR or FDR/ABR full volume backup	None
FULL	same as FULLFDR	
FDR	same as FULLFDR	
INCR	FDR/ABR incremental backup	
ABR	same as INCR	

Table 38. BRNVIFDR execution parameters and default values (continued)		
Parameter	**Description**	**Default value**
ENQF=INCL EXCL	This option is used for Incremental backups only Add 'DATASET ENQ FAILED' entries to BKUPTBL Exclude 'DATASET ENQ FAILED' entries from BKUPTBL	INCL
BKUPDD=	DDname of FDR backup file	TAPE1

Generating JCL for Full Volume Backups (BKUPFULL)

The BKUPFULL utility produces JCL for full volume backups of DASD packs to tape. After the JCL is submitted, information on those backups is recorded in the BKUPTBL and BKUPAUD data sets for subsequent retrieval. The BKUPFULL process consists of several jobs that must run to obtain the required information from the system with the final step being the job that builds the backup jobs themselves from the gathered information and parameters that are set by the user.

BKUPFULL Parameters

Table 39. BKUPFULL parameters	
Parameter	**Consideration**
BKUPMETH	These parameters come from the input file defined by the DD statement BKUPMETH. The two backup methods;. applicable to this process are DFDSS and FDR. They are indicated by the keywords FULLDSS and FULLFDR, respectively, and are inserted into the BKUPMETH file. These keywords point to an associated PDS file (or files if both methods are used), where the JCL produced by the BKUPFULL program for full pack backups resides. The associated PDS must exist. See Installation and Setup for information on the BKUPMETH file.
DCOLLECT	When BKUPFULL runs, one of its first steps is an IDCAMS DCOLLECT execution. This step gathers volume information on all the ONLINE DASD. That information is passed onto the BKUPFULL program which reads the file via DD statement DCOLIN.
BKUPHLQ	The USERMAC member BKUPHLQ is used to define the high-level qualifier for ALL backup names generated by IBM Z Batch Resiliency. This member is modified during the installation process. The value of the BKUP_HLQ parameter should be limited to 19 characters or less.
BKUPJOB	The USERMAC member BKUPJOB is used if present. It has the JCL JOB statement for backup executions. Sequence numbers are inserted into the job name as necessary.
AFULLDSS AFULLFDR	The USERMAC members AFULLDSS and AFULLFDR are templates used in the generation of backup JCL. Before use they must be modified to specify the IBM Z Batch Resiliency CNTL data set name in the JCLLIB statement.
INCLVOLS	The USERMAC member INCLVOLS is used to select the DASD volumes to be backed up. The selection criteria is derived here and matched against the DCOLLECT list of DASD volumes to produce the proper backup JCL.

INCLVOLS Format and Fields

Table 40. INCLVOLS format and fields

Format/Field	Description
Line Format	jobname, volume, backup-skeleton, stack, device-type, rtpd, snap, parallel.
Commentary	A /* in column one is treated as a comment and is not processed. Blank lines are not processed.
Job Name	One to eight bytes in length, with no embedded blanks or commas. It is an optional parameter, and if it is not coded, a comma must be inserted to indicate so. If the Job Name is coded, it is also used as the member name in the output JCL PDS. If more than one Volume is chosen via masking (see below), BKUPFULL automatically produces JCL (in one member) such that the volume backups are stacked on the output tapes; this is handy for the 3590-tape device.
Volume	This required parameter is one to six bytes in length with no embedded blanks or commas. Masking can be used. The ? (question mark) masks a single position; the * (asterisk) masks its position and all to the right. The output JCL PDS member names are derived from the Volume, if a Job Name is not chosen.
FULLFDR or FULLDSS	JCL Templates (see the entry under BACKJCL) that is used to create the JCL and are inserted into the BKUPMETH file. These keywords point to an associated PDS (or PDSs, if more methods are used) where the JCL produced by the BKUPFULL program for full pack backups resides. The associated PDS must exist. There can be multiple skeletons based on the number of different methods that are used.
Stack / Nostack	This parameter tells BKUPFULL to stack the backups onto a tape or not to stack and create a separate tape for each volume. Stack is the default.
Device Type	This is the device type for the output of the backup. If omitted, the device is determined from the backup skeleton JCL.
RTPD	This is the retention period - in days - that is used in the skeleton JCL.
Snap / Nosnap	If set to SNAP, this parameter tells BKUPFULL to resolve the DASD volume to the snapped unit address. Nosnap is the default value.
PARALLEL	Designates the number of output tape drives.

INCLVOLS Example

All CICS® volumes are placed in the job SRFD0504, using the FULLFDR skeleton, with the default of STACK, a retention of 10 days, and these volumes are snapped using one tape drive to back up the volumes.

```
SRFD0504,CICS*,FULLFDR,,,10,SNAP,1
```

INCLVOLS lines either accepted or rejected are noted in the SYSTSPRT DD SYSOUT.

An Application or a Volume may start with a number: If so, the member name output has a $ (dollar sign) prefix.

The BKUPFULL utility can be accessed from IBM Z Batch Resiliency ISPF Option 3 (Execute), Sub-Option 2 (Backup/Recovery). After the job is submitted, review the report in the REPORTS file, member BKUPFULL. Informational and error messages may be viewed in the SYSTSPRT DD SYSOUT. Submitting

the jobs in the output PDS initiates the volume backups. A volume backup tape data set name (created in the BKUPFULL program) is formatted as:

```
hlq.vvvvvv.ddddddd.ttttttt.bbbbbbb
```

where:

hlq = the backup high-level qualifier from BKUPHLQ.

vvvvvv = the DASD volume serial (prefixed with $ if needed).

ddddddd = the date stamp as Dyymmdd.

ttttttt = the time stamp as Thhmmss.

bbbbbbb = the backup method as FULLDSS or FULLFDR.

Within a volume backup job, after a volume backup step, the next two steps execute IBM Z Batch Resiliency programs that insert recovery information into the BKUPTBL and the BKUPAUD files.

Chapter 8. Virtual Recovery Simulator

The Virtual Recovery Simulator option helps you to determine your preparation level for a disaster. The Analysis component provides critical data set information to the Virtual Recovery Simulator process.

The Virtual Recovery Simulator takes a snap shot of the backup files available at the time it is started and monitors all critical file access for the next 24 hours to determine if the snap shot contains a valid backup of each critical file that is accessed during those 24 hours.

While it can be run on usual days, it is perhaps of more use if run on week, month, quarter or year-end days as these infrequent special processing days often have longer and more complex batch flows that don't usually run on other days.

Activating the Virtual Recovery Simulator subtask

Virtual Recovery Simulator is a real-time subtask that runs under the control of the RT/SMF Dispatcher.

About this task

To activate the Virtual Recovery Simulator subtask, complete these steps:

Procedure

1. Uncomment the Virtual Recovery Simulator subtask in USERMAC(VFISTASK) by removing the leading asterisk (*) for this statement:

   ```
   START TASK=IZBR/VRS,PGM=BRNVC011 < VIRTUAL RECOVERY SIMULATOR
   ```

2. To allocate SIMTSNAP and VRSCKPT, use the CNTL(VRSALLOC) member.
   The following figure shows a sample VRSALLOC member.

   ```
   //*===================================================================*
   //* JOB:  VRSALLOC                                                    *
   //*                                                                   *
   //* FUNCTION: ALLOCATE FILES REQUIRED FOR VIRTUAL RECOVERY SIMULATOR.*
   //* *** NOTE: CHANGE SYSDA IN UNIT TO REFLECT THE UNIT TO PLACE DSN  *
   //*                                                                   *
   //*===================================================================*
   //STEP1 EXEC PGM=IEFBR14
   //DD1    DD DSN=IZBRHLQ.SIMTSNAP,
   //          UNIT=SYSDA,
   //          DISP=(,CATLG),SPACE=(CYL,(500,500)),
   //          DCB=(RECFM=FB,LRECL=76,BLKSIZE=0,DSORG=PS)
   //DD2    DD DSN=IZBRHLQ.VRSCKPT,
   //          UNIT=SYSDA,
   //          DISP=(,CATLG),SPACE=(TRK,(1)),
   //          DCB=(RECFM=FB,LRECL=135,BLKSIZE=0,DSORG=PS)
   ```

   Figure 75. VRSALLOC job

3. Define the following data set in the JCL:

   ```
   //SIMTSNAP DD DSN=HLQ.SIMTSNAP,DISP=SHR
   ```

   This data set is used to make a copy of the internal SIM BKUPTBL when Virtual Recovery Simulator initializes. Therefore, it should be sized to accommodate the number of records in the BKUPTBL. It is a condensed copy of the data in the BKUPTBL that Virtual Recovery Simulator uses to verify the existence of a valid backup for the currently identified critical data set by ANALYSIS. The characteristics of this data set are fixed block, physical sequential, logical record length 76 bytes.

4. Define the virtual recovery system checkpoint data set in the JCL:

   ```
   //VRSCKPT DD DSN=HLQ.VRSCKPT,DISP=SHR
   ```

© Copyright IBM Corp. 2019, 2020

When the Virtual Recovery Simulator begins to capture data, a header record is written to the VRSCKPT data set for status and restart purposes.

5. Define the parameters for the Virtual Recovery Simulator process in member USERMAC(RTVRSPRM). Include all parameters separated by commas on one line.

   The following table describes Virtual Recovery Simulator parameters.

Table 41. Virtual recovery simulator parameters

Parameter	Description
SIMPFX=	The Virtual Recovery Simulator Log File High Level Qualifiers. This parameter defines the high-level qualifiers for the log file and must be supplied. This file is dynamically allocated. For example: SIMPFX=IZBR.V1R1M0
SIMNAME=	The name of the Virtual Recovery Simulator run - a 1- to 8-byte field that uniquely identifies the execution of the simulator. It is used to build the name of the SIMLOGF data set names and, therefore, must conform to the rules for data set name qualifiers. This parameter must be supplied. **Note:** The value for SIMNAME must be unique for each Virtual Recovery Simulator run. You might add a date and name that is unique for each run. For example, for October 31, 2019, you would use SIMNAME=SIM91031, structuring the name as SIM*ymmdd* where *y* is year, *mm* is month, and *dd* is day.
SIMLPRI=	Defines the primary allocation value for the SIMLOGF, the log file, in cylinders. If this parameter is not supplied, the default value is 10; its maximum value is 1000.
SIMLSEC=	Defines the secondary allocation value for the SIMLOGF in cylinders. If this parameter is not supplied, the default value is 10; its maximum value is 1000.
SIMVOL=	The VOLSER to allocate the SIM$ files.
SIMUNIT=	Defines the unit for the SIM$ files.

As shown in the following example, the suffix, *SIM$%%.T%%%%%%%*, is appended to the values of SIMNAME and SIMPFX. This indicates that the number of characters available for the SIMNAME and SIMPFX is 28.

The suffix value is changed to the date and time the data set is created.

Using the following example, the files are dynamically allocated at the start of the Virtual Recovery Simulator process and a new one is allocated as required during the processing.

```
HLQ.SIMRCVRY.SIM$07.T0011000
```

A Virtual Recovery Simulator run is not initialized if the defined SIMNAME and SIMPFX combination data sets exist. You must update the RTVRSPRM member for each run or at least delete/rename the data sets with a similar high-level qualifier as defined at Virtual Recovery Simulator initiation.

Starting the Virtual Recovery Simulator

You can start the Virtual Recovery Simulator using the RT/SMF Dispatcher subtask or the Command Processor.

Starting the Virtual Recovery Simulator as an RT/SMF subtask

About this task

To start the Virtual Recovery Simulator as a subtask of the RT/SMF Dispatcher:

Procedure

Add the following statement to the (USERMAC)VFIPARMS member:

```
SIMRCVRY = YES          TURN ON SIMRCVRY AT STARTUP
```

This control statement initializes Virtual Recovery Simulator and starts collecting data at RT/SMF Dispatcher startup time.

Starting the Virtual Recovery Simulator from the Command Processor

The most common method for running the Virtual Recovery Simulator is to activate the process at the Command Processor.

Procedure

1. To start the Virtual Recovery Simulator, use the batch interface `CNTL(RTCMDB)` with `PARM='STARTVRS'`.

   The STARTVRS command starts the Virtual Recovery Simulator process at the next checkpoint – normally within the next ten minutes.
2. To check the status of the Virtual Recovery Simulator, change the PARM= to one of the following commands at the Command Processor and rerun the job:

   ```
   DISPLAY VRS
   DISPLAY RT/VRS
   DISPLAY SIMRCVRY
   ```

   When VRS is scheduled, the returned display is similar to:

   ```
   IZBR RT/SMF:RT/VRS DISPLAY

     TASKNAME  PROGRAM    TCB ADDR  CPU TIME
     --------  --------   --------  ----------
     RT/VRS    BRNVC011   008DE2C8  00:00:00.2
   *ACTIVATION SCHEDULED
   IZBR RT/SMF:RT/VRS DISPLAY COMPLETE
   ```

Results

During the next RT/SMF checkpoint, the following tasks are performed while starting VRS:

- Builds and snaps the SIMTBKUP table from the BKUPTBL.
- Builds and saves run information to the VRSCKPT file. This is used during restart and, unless there is an active run in progress, this data set is empty. This information is also used to determine the Virtual Recovery Simulator end time. The SMF date and time of the first analyzed record is used to create the end time by adding 24 hours to the SMF date/time stamp of the first collected record. Once this is established, the Virtual Recovery Simulator continues to run and collect data until the calculated end

time, unless a STOPVRS command is issued via the Command Processor or the TSO Command Processor interface.

- Allocates SIMLOGF files using SIMNAME and SIMPFX values and the optional SIMLPRI and SIMLSEC values. A new SIMLOGF file is created when 75% of the space allocation for the current file is used.
- Processes critical data sets identified by Analysis. When a data set is deemed critical by BRNVC006, the RT/VRS subtask is notified. RT/VRS uses the snapped consolidated backup table to identify the latest backup status for the critical file. An appropriate VRS record is built and written to the SIMLOGF file for reporting by the batch report module.

Stopping the Virtual Recovery Simulator

The most common method for stopping the Virtual Recovery Simulator is at the Command Processor, using the batch interface.

About this task

Procedure

To stop the Virtual Recovery Simulator, use the batch interface CNTL(RTCMDB) with PARM='STOPVRS'.

The following messages are returned at the next checkpoint:

```
RTS008001I - COMMAND PASSED BY CALLER:  STOPVRS
RTS008004I - RT/VRS TERMINATION SCHEDULED
```

Mirror Volume Support

You can use VRS to identify data that resides on mirrored volumes.

About this task

When mirrored volumes is active, after the system interrogates the consolidated BKUPTBL, an inquiry is made against the mirror volume list to see if the first volume of the data set is a mirror volume. If it is, the data set is processed as a backed-up data set for simulated recovery. In this case, the backup application is reflected as <MIRVOL> and the volume is identified. If it is a multi-volume data set a plus (+) sign is appended to the volume; for example, PROD01+.

To activate mirror volume support:

Procedure

1. Add the following card to the USERMAC(VFITABLS) member in your PDS:

   ```
   MIROVOLS,PARMLIB,MIROVOLS,BRNVL014     OPT MIRROR VOLS TABLE
   ```

2. Edit member USERMAC(DCOLFILT) to list your mirrored volumes by VOLSER, Storage Group and UCB range.
3. Run the CNTL(VRSMIROR) job.

Results

The process creates a list of volumes, filters out non-mirrored volumes and establishes the USERMAC(MIROVOLS) member.

The output is a header card followed by a list of the mirror volumes. When simulated recovery initiates, the system checks for this information, and, if present, builds the MIROVOLS table internally to participate in the data set evaluation.

Reporting

After the VRS data capture is complete, you can build the batch report job using CNTL(VRSREPT).

The JCL to run the reports follows:

```
//jobname JOB ...
//STEP001 EXEC PGM=BRNVB011,PARM='SIMNAME=SIMyyjjj,SIMPFX=IZBR.V110'
//STEPLIB  DD  DSN=IZBR.LIBRARY,DISP=SHR
//SYSUDUMP DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//NOBKUP   DD  SYSOUT=*
//SORTFILE DD  DSN=&&SORTFILE,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(10,10)),
//             DCB=(RECFM=FB,LRECL=85,DSORG=PS,BLKSIZE=0)
//SORTOUT  DD  DSN=&&SORTOUT,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(10,10)),
//             DCB=(RECFM=FB,LRECL=85,DSORG=PS,BLKSIZE=0)
//MESSAGES DD  SYSOUT=*
```

The values for SIMNAME (that is, SIMRCVRY) and SIMPFX (that is, IZBR.V110) are the same values specified in the RTVRSPRM member at the invocation of the VRS feature. The report programs identify the SIM$ data sets associated with the VRS run and dynamically allocate the data sets for the reports. Optionally, you can specify the input SIM$ data sets by including a SIM$IN DD statement and omitting the execution PARM. You must include all the associated SIM$ data sets in the created order for this concatenation.

Virtual Recovery Simulator Reports

There are three output reports produced by the Virtual Recovery Simulator batch report program.

- Virtual Recovery Simulator (Report 1)
- DSN Recovery report (Report 2)
- DSN Recovery report (Report 3)

You can also generate similar information by running the BACKWO report. BACKWO generate a report of all critical data sets without backups. See “Critical data set report (CRITREPT)” on page 194 for more information.

Virtual Recovery Simulator (Report 1)

The Virtual Recovery Simulator (report 1) is one of two reports that go to the REPORT DD statement. This report lists, by application, all the critical data sets accessed during the run period. The report header includes the data sets accessed, along with the run name, run activation time and the collection start and end times.

For each application, the following data is reported:

- data set
- backup application
- backup date
- backup time
- data set open date and time
- job that opened the data set
- data set status
- first access indicator
- volume where the data set exists

If the data set is on a mirror volume the backup application is reported as <MIRVOL>. When a current backup does not exist for the data set, an asterisk (*) is placed in front of the data set name.

After all the applications are reported, a summary is produced that documents the number of applications reported as well as the number of recoverable and non-recoverable applications respectively. Finally, the summary also includes a listing of all the recoverable applications and non-recoverable applications.

```
                     DISASTER RECOVERY/IBM Z BATCH RESILIENCY
   IZBR - VERSION 1.1.0    TERACLOUD S.A. AND IBM CORP.                              DATE: 06/08/2017
REPORT: VIRTUAL RECOVERY SIMULATOR (REPORT1)  RUN NAME: SIMRCVRY  COLLECTION START: 2017.249 13:29:18.34
         ACTIVATED: 2017.249 12:24:29.46                          COLLECTION END  : 2017.250 13:29:18.34

 APPL: APPL1  OPEN O390  CYCLE: 2017.249-00 JOB: SBJCX420  DATE: 2017.249 TIME: 13:31:39.63
              CLOSE O390 CYCLE: 2017.249-00 JOB: SBJC      DATE: 2017.250 TIME: 08:09:15.39

 APPLICATION TOTALS
  APPLS TOTAL:                 12
   RECOVERABLE:                 5
   NOT RECOVERABLE:             7
   NO DSN REF:                  0

  RECOVERABLE APPLICATIONS:
   APPLB1   DAILY    RTA      WEEKLY   PAYROLL

  NON-RECOVERABLE APPLICATIONS:
   APPL1    APPL3    LOSKOBOS OTHERS   TEST     APPL4    APPL5
```

Figure 76. Virtual Recovery Simulator Report 1

DSN Recovery Report (Report 2)

The DSN Recovery report (Report 2) lists, in data set order, the data set along with all the applications and jobs that accessed the data set during the simulation run.

If there is a valid backup, that is listed along with the volume on which the data set resides and the backup date. If there is not a valid backup for the data set, an asterisk precedes the data set name. There is a summary for this report that documents the number of backed up data sets and the number of data sets not backed up respectively.

```
                                  DISASTER RECOVERY/IBM Z BATCH RESILIENCY
 IZBR - VERSION 1.1.0     TERACLOUD S.A. AND IBM CORP.                              DATE: 06/08/2017
REPORT: DSN RECOVERY REPORT (REPORT2)     RUN NAME: SIMRCVRY   COLLECTION START: 2017.249 13:29:18.34
         ACTIVATED: 2017.249 12:24:29.46                       COLLECTION END  : 2017.250 13:29:18.34


 DSN: AG.C21INS.REPORTS.POC              <MIRVOL>: TSO100   BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     APPL4    STFH
 DSN: AG.GDGTEST1.G0004V00               <MIRVOL>: TSO103   BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     WEEKLY   BRNVW002
 DSN: AG.GDGTEST3.G0141V00               <MIRVOL>: TSO102   BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     WEEKLY   BRNVW005
 DSN:*SYS1.LOCAL.CLIST                   BKUPAPPL: NOBACKUP BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     APPL2    SCMB     APPL3    SKWB     LOSKOBOS SJMK     OTHERS   IBMUSER  TEST     SPJE1
 DSN:*SYS1.LOCAL.ISPFPNLS                BKUPAPPL: NOBACKUP BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     APPL2    SCMB     LOSKOBOS SJMK     OTHERS   SCAB     TEST     SPJE1    APPL4    STFH1
     ...
     ...
     ...
 DSN RECOVERY REPORT SUMMARY:  BACKED UP DATA SETS:         291 NON-BACKED UP DATA SETS:          8
```

Figure 77. Virtual Recovery Simulator Report 2

DSN Recover Report (Report 3)

The third report is a subset of report 2 in that it reports only the data sets that do not have a valid backup. This report goes to the VRSNOBKP report.

Each data set that does not have a backup is preceded with an asterisk.

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY                                        PAGE:          1
 IZBR - VERSION 1.1.0     TERACLOUD S.A. AND IBM CORP.                              DATE: 06/08/2017
REPORT: DSN RECOVERY REPORT (REPORT3) NOBKUP  RUN NAME: SIMRCVRY COLLECTION START:2017.249 13:29:18.34
         ACTIVATED: 2017.249 12:24:29.46                      COLLECTION END : 2017.250 13:29:18.34

 DSN:*GDDM.SADMMOD                          BKUPAPPL: NOBACKUP BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     APPL2    SCMB     LOSKOBOS SJMK     OTHERS   SCAB     TEST     SPJE1    APPL4    STFH1
 DSN:*ISP.SISPEXEC                          BKUPAPPL: NOBACKUP BKUPDATE:          BKUPTIME:
     APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB      APPL     JOB
     APPL2    SCMB     APPL3    SKWB     LOSKOBOS SJMK     OTHERS   IBMUSER  TEST     SPJE1
     ...
     ...
     ...
```

Figure 78. Virtual Recovery Simulator Report 3

Chapter 9. IBM Z Batch Resiliency Maintenance

IBM Z Batch Resiliency uses several databases and data sets to track critical applications, manage backups, generate reports, and track historical analysis data.

To ensure accuracy of the IBM Z Batch Resiliency databases and timely data set recoveries, schedule a daily batch process to run these utilities:

- Maintaining the Inventory database (INVMAINT)
- Maintaining the Backup Table (BKUPMNT)
- Maintaining the REPORTS data set (RPTMAINT)
- Maintaining the HISTORY and AUDIT data sets (HISMAINT)
- Maintaining the Keyed History databases
- Updating the critical file list (MAINTLST)

Maintaining the Inventory database (INVMAINT)

The IBM Z Batch Resiliency Inventory database (INVENTRY) contains entries for all data sets used by production jobs. It identifies the data sets within the Inventory that are critical to recovery. The database is continually updated by the Real-Time SMF collection and analysis process.

There are two ways to run Inventory maintenance:

- INVMAINT – runs Inventory maintenance as a batch job. If you run the INVMAINT batch utility, set up automation to stop and restart the Real-Time started task.
- RTINVPRM – runs Inventory maintenance as a real-time started task. Stopping and restarting the Real-Time started task is not required.

Running Inventory maintenance as a batch job (INVMAINT)

To set up Inventory maintenance as a batch job use the USERMAC(INVMAINT) member. The INVMAINT batch utility internally executes both the VSAM NEGFILE and the INVMAINT processes. The NEGFILE contains data set entries, within the Inventory database, that are not cataloged.

Procedure

1. Run CNTL(INVTUNLD) to back up the Inventory database.
2. If the Inventory VSAM NEGFILE has not been defined, modify and run the CNTL(INVTNEG) member.
3. Set up automation to stop the Real-Time started task.
4. Run the CNTL(INVMAINT) JCL.

   The INVMAINT JCL creates two data sets:

   - INVENTRY– contains the data sets required for recovery. This data set format is: *hlq*.INVR1213.INV$BK01
   - INVENTRY – contains the entries to be deleted. This data set format is: *hlq*.INVR1213.INV$18.T3471146
5. Set up automation to restart the Real-Time started task.

Results

The uncataloged data sets in the INVENTRY are removed

© Copyright IBM Corp. 2019, 2020

Setting up R/T Inventory maintenance within the Real-Time started task (RTINVPRM)

The USERMAC(RTINVPRM) member defines the run-time environment for the INVMAINT subtask and backs up the Inventory database prior to running INVMAINT. The USERMAC(RTDICPIN) member schedules when the INVMAINT process occurs. Stopping and restarting the Real-Time started task is not required.

About this task

To set up Real-Time Inventory maintenance, complete these steps:

Procedure

1. Verify the SCKKLOAD and SBRNLOAD libraries are APF authorized and the process has RACF ALTER authority.
2. If the NEGFILE has not been defined, modify and run the CNTL(INVTNEG) member.
3. Uncomment the following two entries in the USERMAC(VFISTASK) member, by removing the leading asterisk (*):

```
START TASK=INVMAINT,PGM=BRNVC059              < RT/INVMAINT
START TASK=RT/ICP,PGM=CKKRC036,SMF=(30)       < RT/ICP
```

4. Uncomment the following line in the USERMAC(VFITABLS) member:

```
INVMAINT,PARMLIB,INVMAINT,BRNVL020
```

   Note: The default BRNXLIST option statement in the USERMAC library member INVMAINT removes all data sets that are not cataloged. DO NOT MODIFY THIS FOLLOWING STATEMENT.

```
REMOVE,APPLID=**,DSN=**,BRNXLIST
```

5. Optional: To change the size of any of the default values, modify the USERMAC(RTINVPRM) member. This member defines the run-time environment for the INVMAINT subtask. Consider increasing the size of BKUPPRI and BKUPSEC, depending on your site requirements.

   The following figure shows an example RTINVPRM member:

```
* INVPFX -HLQ OF INV$ FILE (MAX 19 CHARACTERS IN LENGTH).
* INVNAME-INV$ FILE NODE (8 CHARACTERS).
* IE: INVR#### WHERE #### WILL BE REPLACED BY SYSTEM MONTH (MM) AND DAY (DD).
  IE: INV$ FILE - AG.WORK4X0B.INVR0501.INV$18.T1211130
* INVVOL  -OPTIONAL PARM FOR VOLSER OF INV$ FILE (INVVOL=TSO100)
* INVUNIT -OPTIONAL PARM FOR UNIT OF INV$ FILE (DEFAULT=SYSDA)
* INV$PRI -INV$ FILE PRIMARY CYLS ALLOCATION (DEFAULT=50)
* INV$SEC -INV$ FILE SECONDARY CYLS ALLOCATION (DEFAULT=50).
* BACKUP  -YES WILL BACK UP THE INVENTORY DATABASE PRIR TO RT/INVMAINT.
*        BACKUP FILE GENERATED USING 'INVPFX' AND 'INVNAME'.
*        IE: BKP$ FILE - AG.WORK4X0B.INVR0501.BKP$14.T1211130
* BKUPPRI -BKP$ FILE PRIMARY CYLS ALLOCATION (DEFAULT=50).
* BKUPSEC -BKP$ FILE SECONDARY CYLS ALLOCATION (DEFAULT=50).
* BKPDUNIT -BKP$ FILE DASD DEVICE (DEFAULT=SYSDA)
* BKPTUNIT -BKP$ FILE TAPE DEVICE. (IE 3480)
* NEGFILE  -DEFINES DSNAME OF NEGFILE.
* Example of RTINVPRM member:
INVPFX=QA.WORK4X0B,INVNAME=INVR####
INV$PRI=50
INV$SEC=20
BKUPPRI=15
BKUPSEC=15
BACKUP=YES
NEGFILE=QA.WORK4X0B.INVMAINT.NEGFILE
```

Figure 79. RTINVPRM Real-Time Inventory maintenance

RTINVPRM parameter description

The RTINVPRM parameters are defined as described here.

Parameter
Description

INVR####
This parameter is a constant and becomes part of the INV$ and BKP$ data set name. Month (MM) and day (DD) replaces the ####. For example, if today is June 23, the INVR#### is replaced by INVR0623.

QA.WORK4X0B.INVR0623.INV$20.T1740730
This data set contains the Inventory records that were deleted.

INV$20
Displays the year; in this case 2020.

T1740730
Julian date for June 23 is 174 and time is 07:30 a.m.

QA.WORK4X0B.INVMAINT.NEGFILE
NEGFILE contains the data sets that reside within the Inventory file that are not cataloged.

Scheduling Real-Time Inventory maintenance using ICP

You can use the Interval Control Program (ICP) to schedule Inventory maintenance to run every day, at a specified time. Plan to start the Inventory maintenance process after daily application backups are complete.

About this task

The following figure shows an example USERMAC(RTDICPIN) member:

```
*PCT ASSIGNMENTS, PCT= MUST START IN COL 1 FOR PCT/PPT DEFINITION
*
PCT=INVMAINT PPT=CKKRC039
PCT=RT/ICP   PPT=CKKRC041
PCT=CMDPROC  PPT=CKKRCCMD
*
* ICP REQUESTS, ICP MUST START IN COL 1 FOR ICP DEFINITIONS
*
ICP START TASK(RT/ICP) INTERVAL(0060) PARM(REPEAT)
ICP START TASK(INVMAINT)  PARM(INVMAINT,T:0600)
```

Figure 80. RTDICPIN Sample

Edit the following parameters in RTDICPIN:

Procedure

1. Uncomment the ICP START TASK(INVMAINT) subtask statement, by removing the leading asterisk (*):

   ```
   ICP START TASK(INVMAINT) PARM(INVMAINT,T:0600)
   ```

2. Change the start time for the INVMAINT process. Select a time that occurs after the last scheduled backup completes.

   If you are using Backup Dynamic Dataset Scheduling, start the Inventory process to begin after the BACKJCL completes. For example, if your backups are scheduled to run between 9:00 p.m. and 6:00 a.m., enter the Inventory start time to occur at T:0700 (7:00 a.m.).

RTDICPIN parameter description

The RTDICPIN parameters are defined as follows:

Table 42. RTDICPIN parameters

Parameter	Description
PCT= PPT=	Defines the process control tasks (PCT) and program processing task (PPT) entries. The first character of the parameter must begin in column one and followed by its associated processing entry, PPT. For example, PCT=INVMAINT PPT=CKKRC039.
TASK INVMAINT	Associated with program CKKRC039.
PARM	Indicates the RTINVPRM starts at 1:00 p.m. every day and references the USERMAC(INVMAINT) member.
TASK RT/ICP	Displays the ICP statistics for the RT/SMF started task RTDPRINT DD statement, and repeats every 60 minutes.

Removing specific data sets from the Inventory database (INVREMOV)

You can remove specific data sets that are no longer in use, including obsolete applications or data sets no longer used in production, by modifying USERMAC(INVREMOV).

About this task

When removing data sets from the Inventory database, the preferred method is to first stop Real-Time processing as described in the following steps.

To remove data sets from the Inventory database:

Procedure

1. Stop Real-Time SMF processing.
2. Back up the Inventory database.
3. Modify the USERMAC(INVREMOV) member to remove selected data sets.
   For example, to remove all data sets from the Inventory database with the pattern `USER.HISTORY.TEST.` enter:

   ```
   REMOVE,APPLID=**,DSN=USER.HISTORY.TEST.**
   ```

   For the REMOVE syntax, see Inventory REMOVE requests.
4. Run the INVREMOV job.
5. Review the output. Check the results in the Inventory database to be sure the selected data sets were removed.
6. Restart the Real-Time started task.

   Note: The APPTABLE is refreshed when RT/SMF is restarted.

INVMAINT REMOVE parameter examples

The following statement removes Inventory records for data set names beginning with `ABC.` in any application with an 8-character APPLID beginning with PROD. Option MEMPLATE was not requested, so base type = M records remain.

```
REMOVE,APPLID=PROD%%%%,DSN=ABC.**
```

The following statement removes Inventory records for data set names beginning with ABC., that are no longer cataloged in any applications, with an 8-character APPLID beginning with PROD. Option MEMPLATE was not requested, so base type=M records remain.

```
REMOVE,APPLID=PROD%%%%,DSN=ABC.**,BRNXLIST
```

The following statement removes Inventory records for data sets beginning with ABC., that are no longer cataloged. Data sets that have not been referenced in the last 30 days, and are in any application, with a 4-character APPLID of PROD is removed. Option MEMPLATE was not requested, so base type = M records remain.

```
REMOVE,APPLID=PROD,DSN=ABC.**,LRAGE=30
```

The following statement removes all Inventory base and component data set records matching the data set name mask of ABC.MASTER.SMFEXT.SYSO390.SMF$08.T%%%%%%%

Note: The percent character (%) is a wildcard in any application.

The MEMPLATE statement deletes the base record type. Omitting it keeps the matching base type=M record.

```
REMOVE,APPLID=*,DSN=ABC.MASTER.SMFEXT.SYSO390.SMF$08.T%%%%%%%,MEMPLATE
```

Example of records removed from the INVENTRY database: (rectype=M)

```
B...ABC.MASTER.SMFEXT.SYSO390.SMF$08.T%%%%%%%         M
B...ABC.MASTER.SMFEXT.SYSO390.SMF$08.TESTAPPL
C...ABC.MASTER.SMFEXT.SYSO390.SMF$08.T3333333         M
C...ABC.MASTER.SMFEXT.SYSO390.SMF$08.TESTAPPL
```

The following statement removes all Inventory base records data sets matching the data set name mask of ABC.MASTER.SMFEXT.SYSO390.SMF$08.T%%%%%%%. The percent (%) character is a wildcard in any application.

```
REMOVE,APPLID=*,DSB=ABC.MASTER.SMFEXT.SYSO390.SMF$08.T%%%%%%%,MEMPLATE
```

Example of records removed from the INVENTRY database: (rectype=M)

```
B...ABC.MASTER.SMFEXT.SYSO390.SMF$08.T%%%%%%%         M
B...ABC.MASTER.SMFEXT.SYSO390.SMF$08.TESTAPPL
```

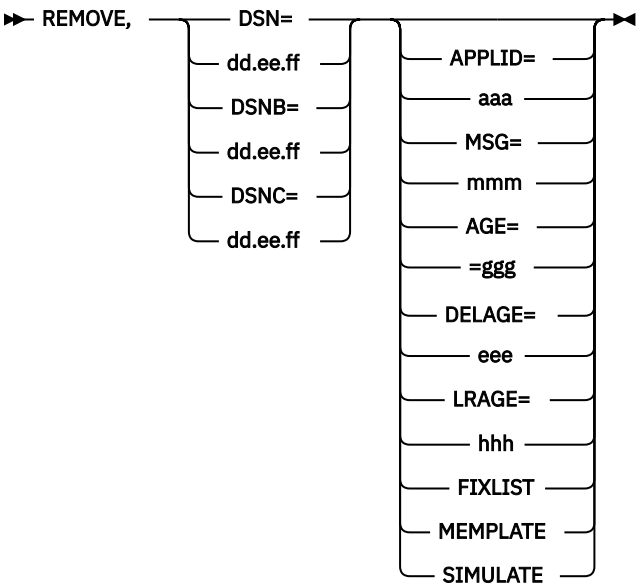
Inventory REMOVE requests

The REMOVE function provides two options for removing Inventory base or component records.

- REMOVE BRNXLIST – deletes only those specified Inventory records whose associated data set names are no longer cataloged.
- REMOVE – deletes all specified Inventory records. Use to remove selected data sets.

Any REMOVE request is required to specify the INVENTRY base or component record data set name using one of these options: DSN, DSNB, or DSNC. Other non-required options available are: APPLID, DELAGE, AGE, LRAGE, MSG, BRNXLIST and SIMULATE. See the table below for option details.

The REMOVE syntax is as follows:



Note: For a REMOVE request, use only one of the DSB, DSN, or DSNAME parameters.

Table 43. DSB, DSN, DSNAME parameters

Parameter	Description
REMOVE,	Signals the start of new records to be processed; this is a required parameter. Note the comma following the parameter.
DSN=	Specifies the data set name of the base record (that is, a GDG base or VSAM cluster). Processes the associated component records. DSN is a complete (up to 44 characters) or masked data set name. Masking follows DFSMS conventions. A percent sign (%) is the wildcard indicator for a single position, and an asterisk (*) indicates any number of positions within a node. The use of a double asterisk (**) matches all subsequent qualifiers; for example, HLQ.** matches HLQ.INFILE.TEST in addition to HLQ.INFILE. To remove all GDG files with a GDG base of ABC.XYZ, specify ABC.XYZ.G%%%%V%%.
DSNB=	The data set name of the base record (that is, a GDG base or VSAM cluster name). Does not process the associated component record. DSNB is a complete (up to 44 characters) or masked data set name. Masking follows DFSMS conventions as defined above.
DSNC=	The data set name of a component record (that is, a GDG generation name) or VSAM data or index record. DSNC is a complete (up to 44 characters) or masked data set name. Masking follows DFSMS conventions.
APPLID=	The application ID associated with the data set.
DELAGE=	The relative delete age for a component record. It must be less than 9999. This parameter is optional and is ignored for base records. Inventory records whose last deleted date is older than the current Julian date minus DELAGE value is deleted if the DSNC is not cataloged.
AGE=	The relative time stamp age for the base and component records. It must be less than 9999 and is an optional parameter. Inventory records whose timestamp is older than the current Julian date minus AGE value is deleted.
LRAGE=	Relative last reference age for base and component record. It is an optional parameter and must be less than or equal to 9999. Inventory records whose last referenced date is older than the current Julian date minus LRAGE value is deleted if the DSN and DSNAME is not cataloged.

Table 43. DSB, DSN, DSNAME parameters (continued)	
Parameter	**Description**
MSG=	Y or N. This option turns program input messages on or off and is an optional parameter. Y is the default value and indicates that all messages are reported. N option suppresses the deleted records messages. However, all other messages, including the summary totals, are displayed.
BRNXLIST	Deletes records depending on whether the DSN/DSNC is cataloged or not. An inventory record with a cataloged DSN/DSNC is kept. Records with an uncataloged DSN/DSNC are deleted.
SIMULATE	Performs the REMOVE function up to, but not including, the inventory record update or delete. The generated report is a listing of all records eligible for update or deletion.
MEMPLATE	Option for REMOVE processing against MEMPLATE (rectype=M) Inventory base b records. The MEMPLATE base b record is bypassed for remove processing if this option is omitted, resulting in the deletion of the component rectype=M, as well as base and component non-rectype=M records. The MEMPLATE option in the remove request allows matching Inventory base b rectype=M records as well as all other matching rectypes to be removed from the file. REMOVE, MEMPLATE must be used with DSN= or DSNB=. The data set name requested can be masked. It is invalid to use options DSNC= BRNXLIST with option MEMPLATE. An invalid option combination generates a warning message and the request is bypassed. For more information on INVENTRY MEMPLATES see MEMPLATE – Grouping data sets by patterns.

Maintaining the Backup Table (BKUPMNT)

IBM Z Batch Resiliency tracks all critical data set backups and where they reside in the central database backup table (BKUPTBL). To ensure the accuracy of the backup table and timely recovery of this data set, schedule a daily batch process to run the maintenance utilities.

Backup table maintenance consists of two jobs that must be run in the following order:

1. BKUPUNLD – backs up the entries in the backup table.
2. BKUPMNT – evaluates the backup table based on defined selection criteria and removes entries that meet these criteria. The BKUPMNT job invokes the XBACKMNT and XBACKMN2 user exits which can be coded to alter the criteria.

Note: You can combine BKUPUNLD and BKUPMNT into one job.

Setting up maintenance for the Backup Table

About this task

To unload entries from the backup table:

Procedure

1. Run CNTL(BKUPUNLD).

   This utility also backs up the backup table (BKUPTBL) and audit file (BKUPAUD), then clears BKUPAUD.
2. Run CNTL(BKUPMNT).

   This utility removes entries from the Backup Table based on defined selection criteria.

   If you would also like to modify the execution of BKUPMNT using the member USERMAC(BACKMANT).

See BKUPMNT processing options for a description of options.

XBACKMNT exit

Use the USERMAC(XBACKMNT) exit to change the value of the retention period (RETPD option) of a record. XBACKMNT is invoked during BKUPMNT processing. This exit point is executed while processing each backup table record.

You can also use XBACKMNT to force a backup to be deleted (physically or logically) from the backup table by setting variable DROP = 1. For example, the following statement deletes any backup for 'APPLID='TEST':

```
IF APPLID = 'TEST' THEN DROP = 1
```

The BKUPTBL fields that are available to be examined in this user exit are DSNAME, APPLID, BKPMETH, BKPDATE, BDSNAME and BKPVOL.

XBACKMN2 exit

Use the XBACKMN2 user exit to issue a delete of tape data sets. XBACKMN2 is invoked during BKUPMNT processing and defined as a DD statement within the JCL. This exit looks for tape names that are no longer cataloged and cleans up the tape management system.

BKUPMNT processing options

To modify the execution of BKUPMNT, use the USERMAC(BACKMANT) member. The following table describes the BKUPMNT processing options.

The default values are underlined.

Table 44. BKUPMNT processing options

Option	Parameter	Description
DELFLAG=	YES/NO	The default value, NO, removes data set entries from the Backup Table when any one data set on tape is no longer valid. **Note:** Do not change the default value unless instructed to by Technical Support. A value of YES keeps all entries in the backup table until all data sets on the backup tape are no longer valid.
TMSAPI=	YES/NO	Invoke Tape Management System API to determine if backup tape is valid. If TMSAPI=NO, then RETPD option is used to determine if backup tape is expired. See also RETPD below. Tape Management Systems CA1 version 5, CA Dynam/TLMS, Control-M/Tape (previously known as Control-T) and RMM are supported.
RETPD=*nnnn*	15	Age of backup in days relative to today's date. Backup data sets prior to this period are considered to be expired and are deleted. By periodically deleting the backups of individual data sets, fewer backup data sets - and therefore fewer tapes - are used because the backups for an application is consolidated into fewer data sets by re-backing up less frequently updated files. Default is 15. This option is only used when TMSAPI option is set to NO.

Table 44. BKUPMNT processing options (continued)		
Option	**Parameter**	**Description**
VERSIONS=*n*	2	Number of the most recent valid backup versions to keep, per data set, being backed up when DELOVER=YES is specified. Default is 2. See also DELOVER below.
DELBREC=	YES/NO	Delete backup table entries whose tapes expired or whose volumes do not match the tape catalog. Default is YES.
DELUCAT=	YES/NO	Delete backup table entries for the backed-up data sets that are not cataloged. Default is YES.
DELBUCAT=	YES/NO/ DASDONLY	Delete backup table entry if the backup data set is not cataloged. DELBUCAT=DASDONLY checks if backup data set is not cataloged for backups residing on DASD only.
DELROLL=	YES/NO	Delete backup table entries for rolled-off generations of GDGs, but keep GDGOFFSET (see below) extra generations. Default is NO.
DELOVER=	YES/NO	Delete backup table entries for any valid version older than the number specified for VERSIONS option (see above).
DNOLINK=	YES/NO	Delete backup table entry if data set is not critical, as indicated in the Inventory database or the Inclusion List.
DELOALC=	YES/NO	Delete all ALLOCAT backup table entries except for most recent version for each data set.
DELUALC=	YES/NO	Delete all ALLOCAT backup table entries for data sets that are not cataloged.
DNLXTND=	YES/NO	Delete Extended backup table entries without an associated base record (for ABARS and FDR).
GDGOFFSET=*n*	1	Number of additional generations of a GDG to keep beyond the number of critical files, as defined in the Inventory. Default is 1.

Maintaining the REPORTS data set (RPTMAINT)

The CNTL(RPTMAINT) member removes older versions of the IBM Z Batch Resiliency reports in the REPORTS data set.

About this task

The CNTL RPTMAINT also provides a detailed list of these reports in the REPORTS($INDEX) member.

Procedure

1. In USERMAC(VFIPARMS) you can edit any of the following options:
   - REPORTS – routes reports to the IBM Z Batch Resiliency Reports library for online viewing.
   - RPTRETPD – sets the number of days, from the current date, for which reports are kept. For example, to keep reports for 60 days, change the value to 60. The default is 999.
   - RPTGENS – sets the number of the most current generations for which reports are kept. The default is 5. For example, to generate one week of reports set RPTGENS to 7.

   See IBM Z Batch Resiliency parameter description (VFIPARMS) for a description of parameter values.
2. Optionally, edit the RPTMAINT options described in the following table.

RPTMAINT options

The RPTMAINT options and parameters are described in the following table. The default values are underlined.

Table 45. RPTMAINT options

Option	Parameter	Description
MAINTAIN=	Yes	Process Report File maintenance.
	No	Bypass Report File maintenance.
BU=	No	Do not process Report File backup. Prevents a pre-maintenance backup of the REPORTS file from being taken, no matter the state of the REPORTS file.
	Force	Always backs up the REPORTS file. Additionally, the input parameter BU={FORCE,NO} controls when and if the REPORTS file backup is taken.
DD2=	Yes	Supply DD=DD2 in the execution JCL to allow a pre-maintenance backup of the input DD=DD1 REPORTS file to be taken into DD=DD2 DSN=*user_dsname* in the JCL statement.
	No	Do not supply DD=DD2 in the execution JCL. Allows a pre-maintenance backup of the REPORTS file to be taken into a dynamically allocated data set. This data set name is created from the *hlq*.REPORTS data set name with an added qualifier at the end. The ending qualifier is made up as follows: X*dddhhmm* with *ddd* indicating the current Julian date and *hhmm* indicating the current time.

Maintaining the HISTORY and AUDIT data sets (HISMAINT)

The HISMAINT member removes older entries from the HISTORY and AUDITDB data sets, based on specified options, and rebuilds the data sets.

HISMAINT performs the following functions:

- Creates the retention period (number of days to be saved), based on the current date and retention period in USERMAC(VFIPARMS).
- Sorts the IBM Z Batch Resiliency HISTORY and AUDITDB database to eliminate duplicate records and rebuilds the HISTORY file.
  - All records that are greater than the cutoff dates are included in the new generation of the database.
  - All other records are removed from the database and moved to level 2 files:
    - HISTORY Level 2, HISTRYL2 file
    - AUDIT Level 2, AUDTL2 file

To control the retention period for the HISTORY and AUDIT data sets, modify parameters in USERMAC(VFIPARMS).

Changing the retention period for the HISTORY data set

The HISTORY data set contains SMF analysis data related to jobs defined in the APPTABLE. You can change the number of days to save the analysis data by modifying the RETPD1 parameter in USERMAC(VFIPARMS).

The value you specify for RETPD1 depends on factors such as the number of production jobs that are run daily, and the number of files opened by those jobs.

In the following example, the HISTORY data set contains records greater than the cutoff date 18130. Therefore, records for 18130 and 18131 are included in HISTORY. Everything earlier than the cutoff date goes into the level 2 file (HISTORYL2).

```
HISTORY file contains SMF records from dates: 18128, 18129, 18130, 18131
VFIPARMS RETPD1 = 1
Current date = 18131
Cutoff date = Current date - RETPD1
     18130    =    18131        -     1
```

To change the resource usage retention period, modify the RETPD2. See IBM Z Batch Resiliency parameter description (VFIPARMS) for more information.

Changing the retention period for the AUDIT data set

The AUDITDB data set contains SMF records related to jobs **not** defined to IBM Z Batch Resiliency in the APPTABLE. To control the retention period of AUDITDB, modify the ARETPD1 parameter in USERMAC(VFIPARMS).

ARETPD1 sets the number of days, counting backwards from the current date, for which non-production SMF analysis data is retained in the AUDITDB. The size of AUDITDB depends on factors such as the number of non-production jobs run daily, and the number of files opened by these jobs.

Critical file auditing

You can use the CNTL(AUDITCHK) member to produce an Audit report. AUDITCHK monitors access to critical files by jobs, TSO sessions, and started tasks that are not identified in the APPTABLE. AUDITCHK then produces a report showing any units of work in the SMF audit data that are opening critical files.

You can use the Audit report to:

- Identify non-production jobs that are opening — and possibly updating — critical files. You should investigate these jobs. If the accesses are insignificant for recovery purposes, add these jobs to the job exclusion table to bypass future reporting.
- Identify jobs that might be critical but have not been entered in the APPTABLE. For subsequent tracking by IBM Z Batch Resiliency, you can add these jobs to the APPTABLE manually.

Maintaining the Keyed History databases

The Keyed History databases contain historical usage of jobs and data sets used by the IBM Z Batch Resiliency ISPF Inquiry panels. The number of Keyed History records and size of the databases can become large in a short period of time.

Maintenance jobs are run daily to remove old records from the History and Keyed History files. Use the following jobs, residing in the CNTL library, to set up daily maintenance for keyed history. Run the jobs in this order:

1. KEYLOAD – sets up Keyed History data collection. Always run this job first.
2. KEYMNTD – purges records from the Keyed History Data Set file CNTL(KEYDHIST).
3. KEYMNTJ – purges records from the Keyed History Job file CNTL(KEYJHIST).

Defining Real-Time Keyed History data collection

You can activate the Keyed History data collection during RT/SMF processing dynamically.

Procedure

1. Update the USERMAC(VFIPARMS) member to include the statement:

```
KEYDHIST = YES
```

2. Define the KEYDHIST and KEYJHIST data sets by running the CNTL(KEYDEFN) member. Optionally, update KEYDEFN by modifying the JCL and IDCAMS control statements to meet your installation standards.
3. Update the library member CNTL(KEYDEFN) by modifying the JCL and IDCAMS control statements to meet your installation standards.
4. Run the CNTL(KEYDEFN) job.
5. Add the two Keyed History files to the Real-Time Analysis JCL in CNTL(RTSMFP) by un-commenting the KEYDHIST and KEYJHIST DD statements.

Defining Keyed History data collection using a batch job

Perform the following steps to activate Keyed History data collection as a batch process.

Procedure

1. Define the KEYDHIST and KEYJHIST data sets by running the CNTL(KEYDEFN) member. Optionally, update KEYDEFN by modifying the JCL and IDCAMS control statements to meet your installation standards.
2. Run the CNTL(KEYDEFN) job.
3. After you define the KEYED History files, modify CNTL(KEYLOAD) and run the job to add the records from the History file to the Keyed History files.

Running the Keyed History maintenance jobs

Perform the following steps to run a Keyed History maintenance job.

Before you begin

Define the KEYDHIST and KEYJHIST data sets by running the CNTL(KEYDEFN) member.

Procedure

1. After you define the Keyed History data sets, modify the CNTL(KEYLOAD) member batch job.

```
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL)
//*================================================================*
//* THIS JCL IS USED TO LOAD THE VFI HISTORY FILE INTO THE 2       *
//* KEYED VSAM FILES KEYDHIST AND KEYJHIST, WHICH ARE USED BY THE  *
//* HISTORY BROWSER APPLICATION.                                   *
//*                                                                *
//* NOTE!!!! THE REGION MUST BE 128M OR GREATER                    *
//*================================================================*
//JS005        EXEC PGM=BRNVB020,REGION=128M
//STEPLIB      DD DSN=HLQ.SCKKLOAD,DISP=SHR
//             DD DSN=HLQ.SBRNLOAD,DISP=SHR
//RTDPRINT     DD SYSOUT=*
//SYSUDUMP     DD SYSOUT=*
//PARMLIB      DD DSN=HLQ.USERMAC,DISP=SHR
//SEQDHIST     DD DSN=HLQ.HISTORY,DISP=SHR
//KEYDHIST     DD DSN=HLQ.KEYDHIST,DISP=SHR
//KEYJHIST     DD DSN=HLQ.KEYJHIST,DISP=SHR
//*
```

Figure 81. KEYLOAD JCL Member

2. Run CNTL(KEYLOAD) to add the records from the History file to the Keyed History files.
3. Run CNTL(KEYMNTD) to purge records from KEYDHIST.
4. Run CNTL(KEYMNTJ) to purge records from KEYJHIST.

Keyed History PARM values

The Keyed History maintenance process separates records to be deleted from the records to be kept, based on the value of PARM.

- PARM='KEYDSN, DAYS=*dd*'
- PARM='KEYDSN, JGENS=*jj*'
- PARM='KEYDSN, DELDATE=*MM/DD/YYYY*'
- PARM='KEYDSN, DAYS=02, REPORT=DETAIL'

Examples of PARM values are described in the following sections.

Deletion selection criteria 1

Job records are deleted when the reader start date and reader start time are greater than the number of days specified in the **DAYS=dd** parameter, based on the reader start date and reader start time.

```
PARM='KEYDSN, DAYS=dd
```

In the following example, if DAYS=02 and the reader start date is 2018.274, then record 1 is removed.

```
                                Reader      Reader      Record       Record
  Dsname               Jobname  Start Date  Start Time   Type        Subtype      SYSID
1 AG.BRNV001.Master    BRNVSRT1   2018.271  11:48:00.99    30          001         0390
2 AG.BRNV001.Master    BRNVSRT2   2018.272  11:57:01.91    30          001         0390
3 AG.BRNV001.Master    BRNVSRT3   2018.273  09:16:01.91    30          001         0390
4 AG.BRNV001.Master    BRNVSRT4   2018.273  10:54:01.91    30          001         0390
5 AG.BRNV001.Master    BRNVSRT4   2018.274  11:54:01.91    30          001         0390
```

Figure 82. Keyed History maintenance job deletion selection criteria 1

Deletion selection criteria 2

Job records are removed when the number of Job generations is greater than the number of generations specified in the **JGENS=jj** parameter. Keep the most recent unique jobs, based on the reader start date and reader start time.

```
PARM='KEYDSN, JGENS=jj'
```

In the following example, if JGENS=02, then record 4 is removed because only the last two versions of a unique job are kept.

```
                                Reader      Reader      Record       Record
  Dsname               Jobname  Start Date  Start Time   Type        Subtype      SYSID
1 AG.BRNV001.Master    BRNVSRT1   2018.271  11:48:00.99    30          001         0390
2 AG.BRNV001.Master    BRNVSRT2   2018.272  11:54:01.91    30          001         0390
3 AG.BRNV001.Master    BRNVSRT3   2018.273  10:16:01.91    30          001         0390
4 AG.BRNV001.Master    BRNVSRT4   2018.272  09:54:01.91    30          001         0390
5 AG.BRNV001.Master    BRNVSRT4   2018.273  10:54:01.91    30          001         0390
6 AG.BRNV001.Master    BRNVSRT4   2018.274  11:54:01.91    30          001         0390
```

Figure 83. Keyed History maintenance job deletion selection criteria 2

Deletion selection criteria 3

This option removes all records prior to the reader start date based on the DELDATE date specified, regardless of any other parameters selected.

```
PARM='KEYDSN, DELDATE=MM/DD/YYYY'
```

For example, if DELDATE=01/20/2018, then all records with a reader start date prior to the specified DELDATE date is removed from the file.

Deletion selection criteria 4

This option calculates backward from the current date by days specified and removes all records prior to the calculated date, regardless of any other selection parameters. You must specify a leading zero if less than 10 days; for example (05). The maximum days allowed is 99.

```
PARM='KEYDSN, DELDATE=(DD)'
```

For example, if DELDATE=(45), then all records with a reader start date prior to 45 days are removed from the file.

By default, REPORT=SUMMARY generates a summary of the process. To obtain detailed reports, you can add the following parameter to the PARM. For example:

```
PARM='KEYDSN, DAYS=02, REPORT=DETAIL'
```

- DAYS and JGENS – parameters are mutually exclusive.
- DAYS and JGENS – specification is limited from 01-99.
- REPORT=SUMMARY – generates a summary report.
- REPORT=DETAIL – generates a detailed report.
- REPORT=SIMULATE – analyze all of the records to be processed.
- Return code=100 – generated to bypass the remaining steps.
- REPORT=ONLY – lists all records that are being processed in the PRINT DD file.
- Return code = 100 – generated to bypass the remaining steps.

Updating the critical file list (MAINTLST)

Critical file list maintenance is performed by the MAINTLST process.

Its functions are to:

- Add newly identified critical files to the Critical File List
- Verify all previous Critical File List entries

The Critical File List contains a complete list of all critical files (both tape and DASD) that IBM Z Batch Resiliency has identified up to that point in time. The list serves as the main data repository for determining which files must be backed up and tracked in the remaining phases and contains an entry for each critical file at the application level. All fields are automatically updated in the Critical File List from the inventory database when the MAINTLST process is run. These fields include: the job name, date, and time the file was last updated, created, and referenced; the number of tracks allocated for DASD files; and critical file status.

The critical file maintenance process has five JCL members that support the management of the critical file (HQL.CRITICAL):

- MAINTLST – Builds and updates the CRITICAL file.
- MANTLSTN – Adds new records to the CRITICAL file by APPLID and DSNAME.
- MANTLSTP – Generates a CRITICAL file report by application and DSNAME.
- MANTREMV – Removes obsolete records from the CRITICAL file.

Building and updating the critical file list (MAINTLST)

The MAINTLST process reads the INVENTRY file and builds a file containing mismatched VSAM base/cluster names. It builds a new critical list file based on the high status of the INVENTRY B records, compares the new critical list file to the existing CRITICAL file (based on the options selected in USERMAC(MAINTLST), and builds a new CRITICAL file after merging and updating records using data from both files.

As part of the MAINTLST process there are two reports produced and directed to the REPORTS file. The MAINTCRT lists all critical files sorted by application.

```
IZBR -VERSION 1.1.0     TERACLOUD S.A. AND IBM CORP.                DATE: 06/08/2017
 CRITICAL FILE REPORT: LIST OF EXISTING CRITICAL FILES BY APPLICATION
RPT: CRITICAL

 ----------------------------- APPLICATION=APPL1 -----------------------------------
DSN                     STAT FILE TYPE LASTRJOB 1STACC  UPDJOB  LAST REF DATE .. FLAGS

AG.SYSA.JULY3TO8           1 NON-VSAM  JRAB     INPUT           25AUG16:12:54    OLD
AG.SYSB.JULY2018.DISASTER  1 NON-VSAM  JRAB     INPUT           25AUG16:14:15    OLD
AG.SYSB.JULY2018.DIS.CPY   1 NON-VSAM  JRAB     INPUT           25AUG16:14:10    OLD
ISP.SISPEXEC               1 PDS       JRAB1    INPUT           01SEP16:07:03    OLD
ISP.SISPMENU               1 PDS       JRAB     INPUT           28AUG16:08:02    OLD
ISP.SISPSLIB               1 PDS       JRAB     INPUT           28AUG16:08:02    OLD
ISP.SISPTENU               1 PDS       JRAB     INPUT           28AUG16:08:02    OLD
TCPIP.ETC.PROTO            1 NON-VSAM  JRAB     INPUT           27AUG16:14:51    OLD
TCPIP.TCPIP.DATA           1 NON-VSAM  JRAB     INPUT           27AUG16:14:51    OLD
USER.CLIST                 1 PDS       JRAB1    INPUT           01SEP16:07:03    OLD
 ..
 ..
 ..
          CRITICAL REPORT TOTALS:       1,112
```

Figure 84. Critical File Report

The second report produced is the MAINTLST report. This report lists files that are new, updated or exceptions. You can identify the reason the data set made it to this report by interpreting the FLAGS portion of the report line.

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY                                PAGE:          1
IZBR -VERSION 1.1.0     TERACLOUD S.A. AND IBM CORP.                DATE: 06/02/2017
 MAINTLST UPDATE CRITICAL FILE REPORT: LIST OF UPDATED CRITICAL FILES BY APPLICATION
RPT: UPDENTRY

 ----------------------------- APPLICATION=APPL1 -----------------------------------

 DSN                     STAT FILE TYPE LASTRJOB 1STACC  UPDJOB  LAST REF DATE  ..FLAGS

 SBAP.ALL430.USERMAC        1 PDS                INPUT                            UPD

          UPDENTRY REPORT TOTALS:           1
```

Figure 85. MAINTLST Update Critical File Report

To retain consistency with earlier maintenance processes, the USERMAC(MAINTLST) member is still utilized. The distributed member is printed here with a brief current description of the parameters.

MAINTLST parameters

MAINTLST includes the following parameters contained in the USERMAC(MAINTLST) member.

```
/*================================================*/
/*                                                */
/*           MACRO CALL FOR MAINTLST REPORT       */
/*                                                */
/*================================================*/

%MAINTLST(ERRORPT=NO,
    NEWENTRY=YES,
    EXCPFILE=YES,
    OLDAPPS=YES,
    EXCLUDED=YES,
    TAPECOPY=YES,
    CFLDUPS=NO,
    STATCHNG=YES)
```

Table 46. MAINTLST parameters	
Parameter	**Description and Values**
ERRORPT=	• Yes - BRNVB034 checks the system catalog to verify the existence of all files. If the file is not cataloged the CFNOCAT flag bit is set. **Note:** Setting this option increases run time for the maintenance job. • No - The catalog is not checked. Default.
NEWENTRY=	**Note:** This option is retained for compatibility with prior versions. Record status is determined for each run and carried in the critical file flag byte. The flag CFNEW is set for a new record. Additionally, CFOLD and CFUPD reflect an old or updated record respectively. Values: • Yes - Determines whether the New Critical File report is produced. Default. • No - Does not produce report.
EXCPFILE=	• Yes - Retained from prior release. Exclude file entries are written to EXCLFILE DD statement rather than the critical file. Default. • No
OLDAPPS=	**Note:** This option is retained for prior versions. The APPTABLE specified in the JCL is used to verify the application existence. Values: • Yes - Process the record application against the USERMAC(APPTABLE) member. Critical flag CFOLDAPP is set if the application does not exist in the APPTABLE. Default. • No - Do not process.
EXCLUDED=	• Yes - The USERMAC member identified for EXCLDSN is used to flag records that match a EXCLDSN entry. The default member EXCLDSN is used unless a different member is identified as PARM input on the EXEC BRNVB034 card, for example EXEC PGM=BRNVB034, PARM='EXCLDSN=EXCLALT'. This allows the user to retain files in the INVENTRY file as critical, but exclude the same data set from the CRITICAL file and IBM Z Batch Resiliency backup processing. The flag CFEXCLF is set to reflect this status. Default. • No - Don't perform EXCLDSN processing.
STATCHNG=	• Yes - Identify files that have a status change; that is, the corresponding record on the INVENTRY file does not exist or the IVB_HSTATUS field is greater than 2, indicating a noncritical data set. The critical flag, CFSTACHG is set to indicate the status change. Default. • No - N/A Option remains from prior releases. Check is done as part of processing.
TAPECOPY=	• Yes - Determines whether tape processing for backup of critical tapes is assigned to the IBM Z Batch Resiliency BRNVCPY program or the global DMS. Yes indicates BRNVCPY controls tape processing. Default. • No - The global DMS controls tape processing (from VFIPARMS).
CFLDUPS=	• Yes - Eliminate duplicate data set names and application IDs. Default. • No - Do not eliminate duplicate data set and application IDs.

MANTLSTP and MANTLSTN Reports

- The CNTL(MANTLSTP) job – generates a critical file report by APPLID and DSN

  See the MAINTLSTP sample report section for a sample MANTLSTP report.
- The CNTL(MANTLSTN) job – generates a critical file report with the following PARM=:
  - CRITRPT – builds a critical report (default)
  - CRITNEW – reports new critical files by application
  - MAINTRPT – builds new, updated and exception report
  - ALL – builds CRITRPT and MAINTRPT

    See the MANTLSTN sample report section for a sample MANTLSTN report.

MAINTLSTP sample report

The following report is a sample of a MAINTLSTP report.

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY                                PAGE:          1
IZBR - VERSION 1.1.0     TERACLOUD S.A. AND IBM CORP.                 DATE: 06/09/2017
CRITICAL FILE REPORT: LIST OF EXISTING CRITICAL FILES BY DATA SET
RPT: CRITICAL

DSN                       STAT FILE TYPE LASTRJOB APPLID   UPDJOB   LAST REF DATE    FLAGS

AG.APPLG1.BKJCL              1 NON-VSAM  JDJS     APPL1              31AUG16:13:08:53 OLD
AG.APPLG1.BRNVANL.DUMP       1 NON-VSAM  JDJS     APPL1    JDJSTER   27AUG16:08:14:19 OLD
AG.APPLG1.BRNVANL.DUMP       1 NON-VSAM  JJMK     APPL2              31AUG16:07:43:47 OLD
AG.APPLG1.D082615.RTDPRINT   1 NON-VSAM  JJMKTST  APPL2              01SEP16:00:47:15 OLD
AG.APPLG1.INVENTRY.CONVERT   1 NON-VSAM  JDJS     APPL1              01SEP16:10:03:26 OLD
AG.APPLG1.INVENTRY.CONVERT   1 NON-VSAM  JMJV     APPL4              01SEP16:10:15:37 OLD
AG.APPLG1.INVENTRY.FLATFILE  1 NON-VSAM  JMLA     APPL5              10AUG16:16:37:36 OLD
AG.APPLG1.INVENTRY.FLATFILE  1 NON-VSAM  JMJV     APPL4              21AUG16:11:28:36 OLD
AG.APPLG1.I1531.SBRNSRCE     1 PDS       JBAP     APPL3              31AUG16:15:34:56 OLD
AG.APPLG1.I1672.CNTL         1 PDS       JDJS     APPL1              31AUG16:13:22:03 OLD
AG.APPLG1.I1672.SBRNSRCE     1 PDS       JBAP     APPL3              12AUG16:15:50:39 OLD
AG.APPLG1.I1672.USERMAC      1 PDS       JBAP     APPL3              12AUG16:15:15:07 OLD
AG.APPLG1.I1672.USERMAC      1 PDS       JMJV     APPL4              12AUG16:15:50:27 OLD
AG.APPLG1.I1672.WS.CNTL      1 PDS       JBAP     APPL3              13AUG16:13:14:52 OLD
AG.APPLG1.I1672.WS.SBRNSRCE  1 PDS       JBAP     APPL3              31AUG16:15:31:57 OLD
AG.APPLG1.I1672.WS.USERMAC   1 PDS       JBAP     APPL3              13AUG16:11:36:07 OLD
AG.APPLG1.JCL.LIB            1 PDS       JDJS     APPL1              31AUG16:13:12:50 OLD
AG.APPLG1.RTDPRINT.XMI       1 NON-VSAM  JBAP     APPL3              26AUG16:17:08:58 OLD
AG.APPLG1.SEAJ               1 NON-VSAM  JJMK1    APPL2    JJMK1     28AUG16:13:26:24 OLD
AG.APPLG1.SEAJ.NEW           1 NON-VSAM  JJMK     APPL2    JJMKTST   01SEP16:15:10:23 OLD
AG.APPLG1.SEAJ.TRACE         1 NON-VSAM  JJMK     APPL2    JJMKTST   01SEP16:15:40:20 OLD
AG.APPLG1.SEAJIX             1 NON-VSAM  JJMK     APPL2    JJMK1     01SEP16:15:02:55 OLD
AG.APPLG1.SEAJIX.NEW         1 NON-VSAM  JJMK     APPL2    JJMKTST   01SEP16:15:05:00 OLD
AG.APPLG1.SBRNSRCE           1 PDS       JBAP     APPL3              31AUG16:15:32:14 OLD
AG.APPLG1.USERMAC            1 PDS       JBAP     APPL3              21AUG16:16:15:28 OLD
AG.APPLG1.USERMAC            1 PDS       JJMK     APPL2              26AUG16:08:49:14 OLD
AG.APPLG2.I1654.INVENTRY     1 VSAM      JCXJ41IR APPL7              21AUG16:08:47:04 OLD
AG.BASE400.VTOCDUMP          1 NON-VSAM  JPJEDCOL APPL8              21AUG16:13:15:15 OLD
AG.BCBS.ESP.SIN.OPCO.XMI     1 NON-VSAM  JPJE     APPL8              21AUG16:14:23:53 OLD
AG.BCBSKS.CNTL               1 PDS       JBAP     APPL3              31AUG16:15:00:38 OLD
AG.BCBSKS.ESP.FORECAST       1 NON-VSAM  JBAP     APPL3    JBAP      31AUG16:15:42:13 OLD
AG.BCBSKS.ESP.FORECAST       1 NON-VSAM  JCXJ     APPL7              26AUG16:15:23:05 OLD
AG.BCBSKS.ESP.FORECAST       1 NON-VSAM  JPJEESP1 APPL8              01SEP16:10:16:26 OLD
AG.BCBSKS.USERMAC.XMI        1 NON-VSAM  JPJE     APPL8              27AUG16:14:22:01 OLD
..
..
..
CRITICAL REPORT TOTALS:       1,112
```

Figure 86. Critical File Report by Data Set, sample report

MANTLSTN sample report

To report new critical files, use CNTL member MANTLSTN. Note the 'NEW' designation in the FLAGS column of the report.

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY                                PAGE:          1
IZBR - VERSION 1.1.0     TERACLOUD S.A. AND IBM CORP.                DATE: 06/09/2017
 MAINTLST NEW CRITICAL FILE REPORT: LIST OF NEWLY ADDED CRITICAL FILES BY APPLICATION
 RPT: NEWENTRY

 ---------------------- APPLICATION=APPL2 ----------------------------------------

DSN               STAT FILE TYPE LASTRJOB  1STACC   UPDJOB    LAST REF DATE    FLAGS

HLQ.DIST.LIBRARY     1   PDS       SMJV     INPUT             02SEP16:09:40:55 NEW
HLQ..SBRNLOAD        1   PDS       SMJV     INPUT             02SEP16:09:13:24 NEW

          NEWENTRY REPORT TOTALS:           2
```

Removing obsolete records (MANTREMV)

The CNTL(MANTREMV) utility removes uncataloged files and exception records from the CRITICAL file. Note that the BRNXLIST option forces a check of the MVS catalog. If the data set is not cataloged, the record is removed from the critical file.

The NOCAT option processes against the CFNOCAT flag in the critical file record which is set via the ERROPTS option in USERMAC(MAINTLST) and the result of the catalog check that was previously performed in the MAINTLST job.

To remove records from the CRITICAL file, select from the following options.

PARM=

BRNXLIST
Remove uncataloged records by Checking MVS Catalog (Default).

OLDAPP
Remove obsolete APPLID records not defined (Flag on).

NOCAT
Remove uncataloged records (Flag on).

STACHG
Remove record if status has changed (Flag on).

EXCLF
Remove record if Flag is on (File was in the USERMAC(EXCLDSN)).

Defining exclusions

Before running the MAINTLST program, you can designate whether the files noted as critical on the ANALYSIS report need to be maintained by IBM Z Batch Resiliency. Add files that are reported as critical but not considered vital for backup purposes to the exclusion list USERMAC(EXCLDSN) before running the MAINTLST program.

- Method 1: Initially, the ANALYSIS report identifies newly found critical files. Review this report for special considerations (for example, files that are being backed up by alternate methods – full-volume or in stream JCL backups). After you identify exceptions and enter them in the data set exclusion list, execute MAINTLST to add the remaining newly found critical files to the critical file list. When you execute the remaining phases of IBM Z Batch Resiliency, it automatically generates backups and tracks them for offsite rotation based on the updated critical file list.
- Method 2: An alternate method is to execute MAINTLST immediately after the daily collection to automatically keep the critical file list current. Periodically, you can use the MAINTCRT report to review exceptions for placement in the data set exclusion table. The advantage of this method is that there is less exposure to the loss of critical data in the event of a disaster and less data has to be kept on-line (the RETPD1 = parameter may be left at 2 days because MAINTLST is executed daily). The disadvantage

of this method is that you may be backing up more files than necessary until the exception data sets are removed from the critical file list.

Running the MAINTLST program using the IBM Z Batch Resiliency ISPF Menu

You can run the MAINTLST program from the main IBM Z Batch Resiliency Primary Panel.

Procedure

1. Type 3 on the OPTION line to select the Execute option.
2. Type 3 on the OPTION line of the Execute Utilities and Reports menu to choose SYSTEM MAINT.
3. Type the appropriate option, shown in the figure below, next to MAINTLST.

Results

```
---------------------- IZBR 1.1.0 System Maintenance Utilities and Reports ------------------
OPTION  ===>
     Enter one of the following selections next to member name:
           S - to Edit JCL                        H - to View Documentation
           P - to Edit Execution Parameters       X - to Edit User Exit Code
           R - to View Report                     Z - to Submit JCL

   .  MAINTLST     - Utility to Update Critical File List
   .  MANTREMV     - Utility to Remove Invalid Critical File Entries
   .  CRITREMV     - Utility to Remove Selective Critical File Entries
   .  INVMAINT     - Utility to Remove Uncataloged Inventory Database Entries
   .  INVREMOV     - Utility to Remove Selective Inventory Database Entries
   .  RPTMAINT     - Utility to Maintain Reports Database
   .  HISMAINT     - Utility to Maintain History/Audit Databases
   .  CKKIHIST     - Utility to Delete and Reallocate History Database
   .  INVTUNLD     - Utility to Create a Sequential Backup of Inventory Database
```

Figure 87. MAINTLST Sample

Reporting non-critical data sets

An additional function of MAINTLST is an option to report non-critical data sets.

There are two members in the CNTL library that execute this function:

NONCRITA
The JCL stream NONCRITA reviews the INVENTRY file, builds a data set NONCRITS, sorts the NONCRITA data set by application and reports the non-critical data sets. A report is directed to the REPORTS file NONCRITA.

NONCRITD
The other JCL stream NONCRITD reviews the INVENTRY file, builds the data set NONCRITS and reports the non-critical data sets in DSN order. A report is directed to the REPORTS file NONCRITD.

Maintaining the InstaSnap zDP audit data set (ZDPMAINT)

The CNTL(ZDPMAINT) member removes obsolete records from the TimeLiner zDP Audit data set.

About this task

The TimeLiner zDP Audit data set is defined using the CNTL(DEFZDPAU) member. This data set name is used as the zDP Dataset option when configuring TimeLiner Preferences, using the IBM Z Batch Resiliency Primary Panel

Note: RT/SMF MUST be shut down *prior* to running ZDPMAINT.

Procedure

1. Ensure RT/SMF has been shut down.
2. Submit member ZDPMAINT from the CNTL library.
3. Ensure RT/SMF is restarted.

Chapter 10. IBM Z Batch Resiliency Utilities

IBM Z Batch Resiliency provides several utilities used to maintain IBM Z Batch Resiliency databases and perform other important tasks.

The more commonly used maintenance utilities are described in IBM Z Batch Resiliency Maintenance.

The less frequently used utilities are described in the following sections:

- MEMPLATE – Grouping data sets by patterns
- DB2®- Auditing the Db2 environment
- IMS – Auditing the IMS environment
- BRNVGENR – Using the high-speed tape copy utility
- BRNMGJBB – Backing up DFSMShsm migrated data sets
- MOVEJCL – Moving critical data sets to other volumes
- BRNVCAT8 – INCLDSN testing utility
- CRITREMV – Removing critical data sets entries
- BKUPRELD,BKUPRBLD - Recovering a corrupted Backup Table

MEMPLATE – Grouping data sets by patterns

The MEMPLATE utility provides the ability to group similarly patterned data sets into a single INVENTRY base record in much the same way GDGs resolve to a single base record. This can be useful to reduce the number of entries in the Inventory database.

Some examples of data sets that would benefit from MEMPLATE are:

- SMF$ data sets
- Application data sets that contain a date in one of its qualifiers
- Data sets with a sequence number in one of its qualifiers

Note: You cannot use the MEMPLATE option with GDG or VSAM data sets.

Activating MEMPLATE Processing

MEMPLATE is an optional feature. Use the following steps to activate MEMPLATE processing, and each time you want to add additional data sets to MEMPLATE.

Procedure

1. Identify which files you want to place under MEMPLATE control.
2. Stop the Real-Time SMF job or started task.
3. Add the selected data sets to the USERMAC(MEMPLATE) member. For example:

```
/*================================================*/
/* MEMPLATE                                       */
/*================================================*/
USER.V4X0.SMFEXT.SYSP390.SMF$%%.T%%%%%%%%
PROD.USER.D%%%%%%%%
**.T%%%%%%%%
AG.WORK4X0!.CNTL.
```

4. Activate the MEMPLATE statement in the VFITABLS member in USERMAC.

© Copyright IBM Corp. 2019, 2020

See Real-Time SMF Dispatcher. The statement should be as follows starting in column 1; the fields of the statement are defined in the VFITABLS member within the USERMAC library.

```
....+....1....+....2....+....3....+....4
MEMPLATE,PARMLIB,MEMPLATE,BRNVL005     OPT MEMPLATE
```

5. Use the IBM Z Batch Resiliency-provided conversion routine in the CNTL(MEMPAPLY) member. This job copies your current INVENTRY data set to a MEMPLATE conversion data set and then converts to MEMPLATE entries per the definitions in the USERMAC(MEMPLATE) member.
6. Review the new Inventory database output (conversion data set) from the MEMPAPLY job.
7. When you are satisfied with the results, replace the production INVENTRY data set with the new Inventory file (the conversion data set from MEMPAPLY).
8. Restart the real-time job or started task.

MEMPLATE Definition Rules

The rules for defining MEMPLATE entries are the same as the DF/DSS rules except that you cannot end the pattern with a single asterisk (*) or a double asterisk (**).

Also, two additional place holder characters have been included to add granularity to the function.

Table 47. Place holder characters for MEMPLATE definition rules

Character	Usage
?	Replaced by a single character
%	Replaced by a single character
+	Replaced by a single numeric character
!	Replaced by a single non-numeric character
*	Replace any number of positions within the node
**	Replaced by zero or more nodes

Note: A single asterisk (*) or double asterisk (**) cannot be used at the end of a pattern.

Using MEMPLATE, all data sets resolve to the same Inventory base record rather than generate a base record for each. The following example shows how to group all data sets beginning with SMF$:

```
HLQ.SMFEXT.S390.SMF$%%.T%%%%%%%
```

These data sets are grouped:

```
HLQ.SMFEXT.S390.SMF$07.T1001010
HLQ.SMFEXT.S390.SMF$07.T1001020
HLQ.SMFEXT.S390.SMF$08.T1001030
HLQ.SMFEXT.S390.SMF$08.T1001040
```

In the following example:

```
PROD.USER.D%%%%%%%.DAILY.FILE
```

would group these data sets together:

```
PROD.USER.D0107100.DAILY.FILE
PROD.USER.D0107101.DAILY.FILE
```

Both data sets containing a date/time stamp in qualifier three resolves to the same MEMPLATE resulting in only one Inventory base record instead of two to support this group of data sets.

The following example MEMPLATE entry demonstrate the use of this feature:

```
/*================================================*/
/* MEMPLATE                                       */
/*================================================*/
USER.V4X0.SMFEXT.SYSP390.SMF$%%.T%%%%%%%%
PROD.USER.D%%%%%%%%
**.T%%%%%%%%
AG.WORK430!.D??????
```

The system builds the INVENTRY base record key using the MEMPLATEs and the DSN. In the example above, the third entry builds records in the Inventory file as follows:

```
DSN=TEST.TEST.T1234567      Base DSN=>TEST.TEST.T%%%%%%%%
DSN=TEST.TEST.TAAAAAAA
DSN=TEST.TEST.T7654321

DSN=PROD.T1234567           Base DSN=>PROD.T%%%%%%%%
DSN=PROD.T7654321
DSN=PROD.T2345678

DSN=PRD.MAST.PLX.T1234567   Base DSN=>PRD.MAST.PLX.T%%%%%%%%
DSN=PRD.MAST.PLX.T2345678
DSN=PRD.MAST.PLX.T0107101
```

DB2 – Auditing the Db2 environment

The IBM Z Batch Resiliency DB2 utility audits the Db2 environment. It matches the list of all Db2 tablespaces to the image copies that Db2 internally tracks.

Initially, select one Db2 subsystem as a proof of concept. Usually the Database Administrator (DBA) is involved with these initial steps:

1. *BIND the PLAN* to make the program VFIDB2 known to Db2.
2. *Grant EXECUTE authority* to allow the TSO user who runs the program or job to read two Db2 tables.

The output reports are of two types: exception and administrative. The key exception reports are:

- Tablespaces with no image copies
- Db2 latest tape image copies not in the vault

The administrative reports include:

- Image copies no longer cataloged
- Lists associating the tablespaces with the image copies

In addition, you can create the Db2 restore JCL based on your installation requirements.

Member USERMAC(DB2) must be initialized with one line for each Db2 system:

- Db2 subsystem name (for example, DB2P)
- Db2 high-level qualifier name (DB2CAT)

DB2READ

DB2READ invokes the IBM Z Batch Resiliency interface, VFIDB2, to extract information from the Db2 environment.

Two primary Db2 tablespaces are queried. The SYSTABLESPACE table contains information about all the databases and tablespaces that Db2 manages and is read with the IBM Z Batch Resiliency program BRNSWEPT. The SYSCOPY TABLESPACE contains information about image copies including the database name, the tablespace name, the image copy name, and the image copy volume. It is read with the BRNSWEPC program. The corresponding information is saved in the IBM Z Batch Resiliency files: Db2BRNVCR and BRN2FILG, respectively. This information is used as input to the DB2BACK program.

Note: SQLCODE 100 indicates that all rows returned from the query had been read.

```
READY
%BRNSWEPT
                    BRNSWEPT SCAN (SYSTABLESPACE) , READING VFIDB2S : 11:04:34
 ======================================================================
 UMACINFO: ( 1:1) DSN1 DSN910
 ----------------------------------------------------------------------
 PROCESSING 1 OF 1 11:04:35 , DATABASE : DSN1 DBCAT: DSN910
 EXECUTING VFIDB2 QUERY:
 >>> SELECT DBNAME, NAME, PARTITIONS, STATUS
       FROM SYSIBM.SYSTABLESPACE
 RESULTS AT : 11:04:42
          RC: 0
     CAFCODE: 0000
   CAFREASON: CAFREASON
     SQLCODE:  100
        ROWS: 0055
 RECS+PARTIT: 59
 ======================================================================
                    BRNSWEPT ENDS: 11:04:44
READY
END
```

Figure 88. BRNSWEPT Successful Execution Example

```
READY
%BRNSWEPC
                    BRNSWEPC SCAN (SYSCOPY) , READING VFIDB2S : 11:04:04
 ======================================================================
 UMACINFO: ( 1:1) DSN1 DSN910
 ----------------------------------------------------------------------
 PROCESSING 1 OF 1 11:04:04 , DATABASE : DSN1 DBCAT: DSN910
 EXECUTING VFIDB2 QUERY:
 >>> SELECT DBNAME, TSNAME, DSNAME, TIMESTAMP, BRNDEVTP,
            DSNUM, ICTYPE, FILESEQNO, DSVOLSER
       FROM SYSIBM.SYSCOPY
      WHERE ICTYPE IN ('F', 'I', 'L')
 RESULTS AT : 11:04:22
          RC: 0
     CAFCODE: 0000
   CAFREASON: CAFREASON
     SQLCODE:  100
        ROWS: 0055
 ======================================================================
                    BRNSWEPC ENDS: 11:04:24
READY
END
```

Figure 89. BRNSWEPC Successful Execution Example

```
                              LISTING OF ALL FULL IMAGE COPYS          06:09 Friday, September 25, 2018   1

Obs   DBNAME     TBNAME    DSNAME                                      TIME              UNIT  PART  ICTYPE  FILSEQ

  1  DSN8D61A   DSN8S61D   DSN910.SYSCOPY.DSN8D61A.DSN8S61D   2018-02-05-09.30.53   3390     0       F       0
  2  DSN8D61A   DSN8S61E   DSN910.SYSCOPY.DSN8D61A.DSN8S61E   2018-02-05-09.30.54   3390     0       F       0
  3  DSN8D61P   DSN8S61C   DSN910.SYSCOPY.DSN8D61P.DSN8S61C   2018-02-05-09.30.55   3390     0       F       0
  4  DSN8D61A   DSN8S61P   DSN910.SYSCOPY.DSN8D61A.DSN8S61P   2018-02-05-09.30.55   3390     0       F       0
  5  DSN8D61P   DSN8S61C   DSN910.COPYDDN.DSN8D61P.DSN8S61C   2018-02-05-09.32.43   3390     0       F       0
  6  DSN8D61A   DSN8S61E   DSN910.DSN8D61A.DSN8S61E.PART3     2018-02-05-09.33.26   3390     3       F       0
```

Figure 90. DB2READ Report Sample

Db2 Error Conditions

If, while running DB2READ or DB2BACK, you encounter an SQLCODE value of -818, the IBM Z Batch Resiliency interface to Db2 has not been bound properly. An SQLCODE value of 991 indicates that the user does not have the appropriate authority to execute the command. See “DB2 – Auditing the Db2 environment” on page 161 for instructions on binding IBM Z Batch Resiliency to Db2, and granting EXECUTE authority. For other Db2 error codes (SQLCODEs), refer to *Db2 Universal Database (UDB) for z/OS Codes* (GC18-9603).

```
  UMACINFO: ( 1:1) DSN1 dbname
  ------------------------------------------------------------
  PROCESSING 1 OF 1 hh:mm:ss , DATABASE : DSN1 DBCAT: dbname
  EXECUTING VFIDB2 QUERY:
  >>> SELECT DBNAME, TSNAME, DSNAME, TIMESTAMP,
             BRNDEVTP, DSNUM, ICTYPE, FILESEQNO, DSVOLSER
        FROM SYSIBM.SYSCOPY
       WHERE ICTYPE IN ('F', 'I', 'L')
  RESULTS AT : hh:mm:ss
           RC: 200
      CAFCODE: 0000
    CAFREASON: CAFREASON
       SQLCODE: -818
          ROWS: ROWS
     50 +++   DO R = 1 TO ROWS
 Error running BRNSWEPC, line 50: Bad arithmetic conversion
```

Figure 91. Db2 Bind Error Output

DB2BACK

The job DB2BACK is a procedure used to cross reference Db2 tablespaces with their backups.

This report is useful to identify potential missing elements of a Db2 recovery strategy. The reports are based on tablespaces and their associated backups.

When DB2BACK runs, all current Db2 image copies scanned are cross-referenced with the Db2 databases and tablespaces to assure completeness. In addition to cross-referencing the two files, all backups are verified against the tape management system to verify that the volume does not differ from the tape catalog. Additional checks are made to assure that the latest image copy is vaulted offsite. DB2BACK uses the exit member USERMAC(BKUPVALT), the same exit invoked by BACKUP to determine off site rotation.

Sample Reports from DB2BACK

```
BACKUP ENTRIES NOT FOUND IN THE TAPE MANAGEMENT SYSTEM                                    1
                                                          10:10 Wednesday, November 21, 2018

Obs          DSNAME           APPLID      BDSNAME      BKPMETH   BKPDATE   BKPTIME   BKPVOL   TM_RC

 1    DSN8D61A_DSN8S61E_000   BRNDB2   DR.DB2.DBEMP    DB2FULL   2006208   113559    VT1059     7
 2    DSN8D61A_DSN8S61E_000   BRNDB2   DR.DB2.DBEMP    DB2FULL   2006208   154823    VT1059     7
 3    DSN8D61P_DSN8S61C_000   BRNDB2   DR.DB2.DBCRAP   DB2FULL   2006208   150512    VT1080     7
```

Figure 92. DB2BACK sample report 1

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY             DATE: 11/21/15 TIME: 10:12:25
IZBR - VERSION 1.1.0       TERACLOUD S.A. AND IBM CORP.              PAGE: 0001
MASTER INDEX OF TABLESPACE BACKUPS                              (PTDETAIL=NO)
------------------------------------------------------------------------------

--TABLESPACE:DSNDB04_DGORDGOD_000 PARTITIONS:001
 APPLID:BRNDB2 BACKUP:                                    TIME:                . METH:        VOL:       VAULT:
 **** NO BACKUP FOUND ****

--TABLESPACE:DSNDB06_SYSDBASE_000 PARTITIONS:001
 APPLID:BRNDB2 BACKUP:DR.DB2KOPY.DSNDB06.SYSDBASE          TIME: 18SEP15:10:34:59 METH:DB2FULL VOL:DLT033 VAULT:SHELF
 APPLID:BRNDB2 BACKUP:DR.DB2FULL.DSNDB06.SYSDBASE.G0001V00 TIME: 27SEP15:11:15:49 METH:DB2FULL VOL:DLT036 VAULT:SHELF
 APPLID:BRNDB2 BACKUP:DR.DB2FULL.DSNDB06.SYSDBASE.G0002V00 TIME: 27SEP15:11:18:08 METH:DB2FULL VOL:DLT029 VAULT:SHELF
```

Figure 93. DB2BACK sample report 2

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY              DATE: 11/21/15 TIME: 10:12:28
IZBR - VERSION 1.1.0       TERACLOUD S.A. AND IBM CORP.                PAGE: 0001
LIST OF TABLESPACES WITHOUT BACKUPS
--------------------------------------------------------------------------------

 APPLID: BRNDB2

  TBLSPCE:DSNDB04_DGORDGOD_000                              APPLID:BRNDB2    BACKUP METHOD:
  TBLSPCE:DSNDB04_DGORDGOP_000                              APPLID:BRNDB2    BACKUP METHOD:
  TBLSPCE:DSNDB04_PLANRTAB_000                              APPLID:BRNDB2    BACKUP METHOD:
  TBLSPCE:DSNDB04_STAFF_000                                 APPLID:BRNDB2    BACKUP METHOD:
```

Figure 94. DB2BACK sample report 3

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY              DATE: 11/21/15 TIME: 10:12:28
IZBR - VERSION 1.1.0       TERACLOUD S.A. AND IBM CORP.                PAGE: 0001
LIST OF SUCCESSFUL BACKUPS
--------------------------------------------------------------------------------

 APPLID: BRNDB2

  TBLSPCE:DSNDB06_SYSDBASE_000
```

Figure 95. DB2BACK sample report 4

IMS – Auditing the IMS environment

The IBM Z Batch Resiliency IMS utility audits the IMS environment. However, it captures the RECON image copy information and matches it to the databases defined to IMS.

Unlike Db2, there is no requirement initially for DBA involvement. In fact, IMS doesn't need to be running while the IBM Z Batch Resiliency IMS component collects the information for our reports.

IMS auditing helps you determine what IMS database do not have backups. The results are summarized in these reports.

The output reports in REPORTS(IMSBACK) should be reviewed to validate possible discrepancies. The following reports are produced:

- **IMSBKALL** – Master Index of database backups
- **IMSBKREC** – List of databases without backups

BRNVGENR – Using the high-speed tape copy utility

The BRNVGENR utility is a substitute for IEBGENER. Use BRNVGENR to copy a single tape file to another tape.

You can invoke BRNVGENR from the IBM Z Batch Resiliency file backup methods, or you can use it as a standalone utility. In standalone mode, BRNVGENR can produce multiple output files in a single pass.

```
//jobname JOB ...
//*
//*-----------------------------------------------
//* BRNVGENR SAMPLE JCL
//*-----------------------------------------------
//*
//BRNVGENR EXEC PGM=BRNVGENR,REGION=0M
//STEPLIB  DD  DISP=SHR,DSN=HLQ.LIBRARY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DISP=(OLD,PASS),
//             UNIT=TAPE,
//             VOL=SER=xxxxxx
//             DSN=IZBR.TEST.DATASET
//SYSUT2   DD  DISP=(NEW,PASS),
//             UNIT=TAPE,
//             VOL=SER=yyyyyy,
//             DSN=IZBR.TEST.DATASET.COPY
```

Figure 96. BRNVGENR JCL sample

BRNMGJBB – Backing up DFSMShsm migrated data sets

The BRNMGJBB utility generates JCL for backing up files that have been migrated by HSM. You can use this utility in conjunction with the IBM Z Batch Resiliency BKUPJCL process, as an alternative to ABARS, for backing up files that are migrated by HSM.

The files are first recalled by HSM and then backed up using the DSS program ADRDSSU, and optionally, migrated back by HSM. These are all done in groups of files as specified by the MAXFILES execution parameter which you can customize. The information on the backups is recorded in the BKUPTBL similar to any other backup methodology. The input to BRNMGJBB is a member of a PDS that contains Backup INCLUDE control cards in DSS format.

You can find sample JCL to execute the BRNMGJBB utility in the CNTL(BRNMGJBB) member. Sample JCL to submit the jobs generated by BRNMGJBB is in the CNTL(MIGJBSUB) member.

BRNMGJBB Options

The following parameters are used to affect the execution of BRNMGJBB and are in the USERMAC(MIGBKPRM) member.

The default value for each parameter below is underlined.

Table 48. Default parameter values for BRNMGJBB options

Parameter	Description
INTYPE=DSS	Format of the input control cards to be scanned. The only valid value currently supported is DSS.
TEMPLATE=MIGBKUP	Template for the JCL which is generated by BRNMGJBB. The template is a member that resides in the USERMAC library and consists of the letter 'A' followed by the value of TEMPLATE. If you specify TEMPLATE=MIGBKUP, the template resides in member AMIGBKUP in USERMAC.
JOBMEM=MIGBKP	Specifies the first 6 characters of the member name where the JCL is generated in the library specified in the backup methodology member MIGBKUPM in USERMAC. The complete member name has nn appended to the 6 characters specified by JOBMEM, where nn is a number from 01 to 99 based on the number of members that are generated. For example, if you specify JOBMEM=MIGBKP, the JCL is generated in member MIGBKP01. Additional members are MIGBKP02, MIGBKP03, and so on.
JOBNAME=BRNVBKP	Specifies the jobname that is used on the JOBCARD of the job which is generated by BRNMGJBB.
JOBNAM*nn*=	If more than 1 job is generated, JOBNAMn (where *n* is a number between 2 and 99) specifies the jobname that is used for the additional jobs. If not specified, the value of JOBNAME is used for the additional jobs as well.
JOBCOUNT=1	Specifies the number of jobs that are generated. A value between 1 and 99 may be specified. Because of the MVS limitation of 255 steps per job, if you specify a count which causes more than 255 steps, a warning message is issued by BRNMGJBB and additional jobs are generated.
MAXFILES=25	Specifies the number of files that are processed per group. For example, if you specify MAXFILES=10, then 10 files is recalled, backed up and migrated at a time.
STACK=YES	This option specifies whether stacking should be used for backing up the files. If YES is specified, all the backup steps within the generated job is stacked on one output tape. If NO is specified, then each backup step generates a separate backup tape.

BRNMGJBB Considerations

Because the JCL generated by BRNMGJBB contains steps that invoke HSM to recall the migrated files, it is important to set the MAXFILES parameter to a value that accommodates the number of data sets that are recalled to DASD at one time.

Setting the value too high can cause the recalls to fail, if for example, there is not enough DASD space available for the recalls. Conversely, setting the value too low causes the generated job to run longer. Optimally, you should use the highest value possible that would accommodate the number of data sets that are recalled to DASD volumes at one time.

In addition, the JOBCOUNT parameter defines the number of jobs that is generated by BRNMGJBB. If there are a large number of migrated files to be backed up, a higher value for the JOBCOUNT option shortens the run time by splitting the files among the number of jobs specified by JOBCOUNT.

Sample JCL to Execute BRNMGJBB

In the following sample JCL, the DSS control cards in member MIGRAT, pointed to by DDname //BKJCLIN, is scanned by BRNMGJBB and the jobs for backing up the migrated files are generated.

The JCL is generated in the library specified in //BKUPMETH DDname with member names that begin with the characters specified by the value in parameter JOBMEM.

The template used to generate the JCL is specified by DDname //TEMPLATE with a member name that starts with the character 'A' followed by the value specified in the TEMPLATE parameter.

```
//JS20     EXEC  PGM=IKJEFT01,
//               PARM='BRNMGJBB'
//STEPLIB  DD  DISP=SHR,DSN=HLQ.LIBRARY
//SYSEXEC  DD  DISP=SHR,DSN=HLQ.SBRNSRCE
//PARMLIB  DD  DISP=SHR,DSN=HLQ.USERMAC(MIGBKPRM)
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//BRNVPRNT DD  SYSOUT=*
//SYSIN    DD  DSN=&&SYSIN,DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//             SPACE=(TRK,(1,1)),DCB=(LRECL=80,BLKSIZE=27920)
//DFSPARM  DD  DSN=&&DFSPARM,DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//             SPACE=(TRK,(1,1)),DCB=(LRECL=80,BLKSIZE=27920)
//SORTIN   DD  DSN=&&SORTIN,DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(15,5)),DCB=(LRECL=60,BLKSIZE=27960)
//SORTOUT  DD  DSN=&&SORTOUT,DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(15,5)),DCB=(LRECL=60,BLKSIZE=27960)
//BKJCL    DD  DSN=&&BKJCL,DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//             SPACE=(TRK,(5,10)),DCB=(LRECL=133)
//TEMPLATE DD  DISP=SHR,DSN=HLQ.USERMAC
//XMAKEJCL DD  DISP=SHR,DSN=HLQ.USERMAC(XMIGBK01)
//MAKEJCL  DD  DISP=SHR,DSN=HLQ.USERMAC(MAKEJCL1)
//UFIELDS  DD  DISP=SHR,DSN=HLQ.USERMAC(UMIGBKUP)
//BKUPMETH DD  DISP=SHR,DSN=HLQ.USERMAC(MIGBKUPM)
//BKJCLIN  DD  DISP=SHR,DSN=HLQ.BKUPJCL.DSS(MIGRAT)
```

Figure 97. Sample JCL to execute BRNMGJBB

The following example shows the steps in the first group of files when parameter MAXFILES=2. The job processes two files at a time from this group.

```
//BRNVBKP JOB (),CLASS=A,MSGCLASS=X,REGION=6M,NOTIFY=&SYSUID
//*
//*-------------------------------------------------------------
//*--- JSR00001: RECALL MIGRATED DATASETS TO BE BACKED UP  ---
//*-------------------------------------------------------------
//JSR00001 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 HSEND WAIT RECALL HLQ.CNTL
 HSEND WAIT RECALL HLQ.DOC
/*
//*
//*-------------------------------------------------------------
//*--- JSB00001: BACKUP DATASETS THAT WERE MIGRATED        ---
//*-------------------------------------------------------------
//JSB00001 EXEC PGM=ADRDSSU,COND=EVEN
//SYSPRINT DD DSN=&&SYSOUT,DISP=(NEW,PASS),
//            UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
//            DCB=(RECFM=VBA,DSORG=PS,LRECL=121,BLKSIZE=1214)
//TAPE1    DD DSN=DR.BACKUP.MIGRAT.D141114.T084131.BK0001,
//            DISP=(,CATLG),UNIT=VTAPE,
//            VOL=(,RETAIN,,1),
//            LABEL=(0001,SL,RETPD=14)
//SYSIN    DD DISP=SHR,DSN=HLQ.USERMAC(DSSCARD)
//BRNVFILT DD *
 INCLUDE(              -
          HLQ.CNTL     -
          HLQ.DOC      -
          )
/*
//*
//*-------------------------------------------------------------
//*--- JSP00001: PRINT DF/DSS SYSOUT FROM BACKUP STEP      ---
//*-------------------------------------------------------------
//JSP00001 EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1   DD DSN=&&SYSOUT,DISP=(SHR,PASS)
//SYSUT2   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//*-------------------------------------------------------------
//*--- JSQ00001: PROCESS DF/DSS SYSOUT FROM BACKUP STEP    ---
//*-------------------------------------------------------------
//JSQ00001 EXEC PGM=BRNVBUQRY,COND=EVEN
//STEPLIB  DD DSN=HLQ.LIBRARY,DISP=SHR
//BRNVIN   DD DSN=&&SYSOUT,DISP=(SHR,PASS)
//ERRLOG   DD SYSOUT=*
//*-------------------------------------------------------------
//*--- JSV00001: UPDATE DR/IZBR BKUPTBL WITH BACKUP INFO    ---
//*-------------------------------------------------------------
//JSV00001 EXEC PGM=BRNVBDSS,PARM=('GLOBALM1,'),
//            COND=((0,NE,JSQ00001),EVEN)
//STEPLIB  DD DISP=SHR,DSN=HLQ.LIBRARY
//TAPEHOLD DD DSN=*.JSB00001.TAPE1,DISP=OLD,UNIT=VTAPE
//DSSOUT   DD DSN=&&SYSOUT,DISP=(SHR,DELETE)
//DDIN     DD DSN=&&BRNVTEMP,DISP=(NEW,DELETE),
//            UNIT=SYSDA,SPACE=(CYL,(5,5)),
//            DCB=(LRECL=1741,BLKSIZE=31338,RECFM=FB,DSORG=PS)
//ERRORS   DD SYSOUT=*
//ERRVSAM  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//BKUPTBL  DD DISP=SHR,AMP=('BUFNI=12,BUFND=49'),
//            DSN=HLQ.BKUPTBL
//BKUPAUD  DD DISP=SHR,DSN=HLQ.BKUPAUD
//*
//*-------------------------------------------------------------
//*--- JSM00001: MIGRATE RECALLED DATASETS BACK            ---
//*-------------------------------------------------------------
//JSM00001 EXEC PGM=IKJEFT01,COND=EVEN
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 HSEND MIGRATE DS(HLQ.CNTL)
 HSEND MIGRATE DS(HLQ.DOC)
/*
```

Figure 98. JCL Generated by BRNMGJBB

MOVEJCL – Moving critical data sets to other volumes

Use the MOVEJCL utility to produce a report and, optionally, JCL for moving DASD data sets from one volume to another volume.

MOVEJCL is useful to determine if any critical data sets are located on incorrect volumes. For example, if you save a data set to a work volume which should have been saved to a production volume, use MOVEJCL to move it to the appropriate volume.

You can also use MOVEJCL in a replicated environment to ensure that critical data sets are located on replicated volumes. The range of replicated volumes is specified in the USERMAC(INCLMOVE) member. MOVEJCL checks whether each data set in the Inventory database is located on one of those mirrored volumes and presents the findings in a report. Another parameter in INCLMOVE optionally directs the program to produce JCL to move critical data sets not on mirrored volumes to the specified mirrored volumes.

After the job is submitted, review the report in the REPORTS data sets, member MOVEJCL. You can view informational and error messages in the SYSTSPRT DD SYSOUT. Submission of the jobs in the output PDS files causes the actual data sets to be moved. The output PDS files are specified in HLQ.BKUPMETH on the lines beginning with MOVEDSS and MOVEFDR.

MOVEJCL data sets

The following table describes the MOVEJCL data sets.

Table 49. MOVEJCL data sets

Component	Description
BKUPMETH file	The two move methods are the utilities DFDSS and FDR. They are indicated by the keywords MOVEDSS and MOVEFDR, respectively, and are inserted into the BKUPMETH file. These keywords point to an associated PDS file (or files, if both methods are used), where the JCL produced by the MOVEJCL program for data set movement resides. The associated PDS must exist prior to the execution of the utility. See Chapter 2, “Installation and Setup,” on page 5 for information on the BKUPMETH file.
DCOLLECT file	When MOVEJCL runs, its first step is an IDCAMS DCOLLECT execution. This step gathers volume information on all the on-line DASD. That information is passed onto the next step, which is the actual MOVEJCL program.
BKUPJOB member	The USERMAC member BKUPJOB is used if present. It has the Job JCL statement for data set move executions. Sequence numbers are appended to the Job Name.
Critical File List	The Critical File List is read as input to MOVEJCL. Data sets are selected from CRITICAL per the parameters given in INCLMOVE (see INCLMOVE below).
INCLDSN member	The USERMAC member INCLDSN, if present, is read as input to MOVEJCL. Data sets are selected from INCLDSN per the parameters given in INCLMOVE (see INCLMOVE below). See Chapter 9, “IBM Z Batch Resiliency Maintenance,” on page 139 for information on INCLDSN.
INCLMOVE member	The USERMAC member INCLMOVE is used to select the application, volumes, Storage Groups, and data set move method. The selection criteria are derived here and matched against the DCOLLECT list of DASD volumes, the Critical File List, and the INCLDSN member to produce the reports and JCL to move the data sets. INCLMOVE lines rejected are noted in the SYSTSPRT DD SYSOUT.

INCLMOVE Parameters

Table 50. INCLMOVE parameters

Parameter	Description
INCLMOVE Line Format	`Application,Desired-Volumes/Storage-Groups/UCBs[,Movement-Method][,Preferred-Volumes]` Commas must separate each parameter. Code one line per application, using continuation if needed. Continuation to the next line is indicated by a comma at the end of a line. A slash, asterisk (/*) in column one indicates that that line is a comment and all its characters are ignored (that is, not processed). Blank lines are not processed.
Application	**Note:** This is a required parameter. The application consists of one to eight characters, with no embedded blanks or commas and is a required parameter. It is used as the member name in the output JCL PDS, if JCL is desired. If an Application name starts with a number, the corresponding member name generated has a $ (dollar sign) prefixed to it. An asterisk (*) used to represent all applications.
Desired Volumes, Storage Groups, Tape or UCBs	**Note:** This is a required parameter. The Desired Volumes, Storage Groups, Tape or UCBs, consists of volumes and masked volumes, one to six characters each, or SMS Storage Groups or UCB addresses. Volumes, Storage Groups, Tape and UCB addresses may be intermixed. Masking is available for volumes only and is implemented as follows: the ? (question mark) masks a single position, the * (asterisk) masks its position and all positions to the right. Storage Groups are indicated by the prefix SG: followed by a one to eight-character Storage Group name, with no embedded blanks or commas. UCBs are indicated by the prefix "UCB:" followed by a pair of four-character UCB addresses separated by a dash. Use as many volumes and volume masks as needed, separated by commas. Similarly, use as many Storage Groups or UCBs as needed, separated by commas, each prefixed by SG: or UCB: respectively. The volumes or Storage Groups chosen are where the data sets of the Application in question should reside (for example, they could indicate the mirrored volumes).
Movement Method	**Note:** This is an optional parameter. Movement Method consists of either MOVEDSS or MOVEFDR for the desired data set move utility program, DFDSS or FDR, respectively. It is an optional parameter: if chosen, move JCL is produced and inserted into the chosen JCL library, otherwise, only a report is produced. The Movement Method, if present, delimits the Desired Volume parameters from the Preferred Volumes parameters (if chosen). The Movement Method can only be used with VOLUMES.

Table 50. INCLMOVE parameters (continued)	
Parameter	**Description**
Preferred Volumes	**Note:** This is an optional parameter. The description of this parameter is the same as for the volumes in the Desired Volumes parameter. Preferred Volumes, if chosen, must be a subset of Desired Volumes. Masking is implemented in the same manner as for Desired Volumes. Preferred Volumes may only be used with the VOLUMES parameter. Preferred Volumes may not be used if there is no Movement-Method.

Parameter Examples

The GL Application, all PROD- prefixed DASD volumes, using DFDSS:

```
GL,PROD*,MOVEDSS
```

The SYSTEM Application, all SYS- and TECH- prefixed DASD volumes, using FDR:

```
SYSTEM,SYS*,TECH*,MOVEFDR
```

The PAY Application, only Storage Group LARGE:

```
PAY,SG:LARGE
```

The ACCOUNT Application, Storage Groups LARGE, SMALL, and MEDIUM:

```
ACCOUNT,SG:LARGE,SG:SMALL,SG:MEDIUM
```

The SYSTEM Application, all SYS- and TECH- prefixed volumes and the OS390A volume, reporting only:

```
SYSTEM,SYS*,TECH*,OS390A
```

The SYSTEM Application, all SYS- and TECH- prefixed DASD volumes, using FDR, and preferred volumes SYS*:

```
SYSTEM,SYS*,TECH*,MOVEFDR,SYS*
```

The HACP Application, all ABC-, DEF- and XYZ-prefixed volumes, certain TSO volumes with any character in position 4, and volumes WORK01 and WORK99, using DSS, preferred volumes XYZ* (with continuation):

```
HACP,ABC*,DEF*,XYZ*,TSO?00,WORK01,WORK99,MOVEDSS,
XYZ*
```

All applications on devices 190 through 19F with reporting only:

```
*,UCB:0190-019F
```

MOVEJCL Options

These parameters are used to affect the execution of MOVEJCL and are found in member MOVEJCL in library USERMAC. The default values are underlined.

Table 51. MOVEJCL options		
Option	**Parameter**	**Description**
DETAIL	Y or N	Details are printed
EXCEPTIONS ONLY	Y or N	Only exceptions are reported
HEADERS	Y or N	Report headers are printed
VOLINFO	Y or N	Volume information is provided for each data set
GDGEXPAND	Y or N	Data set names are expanded for generation data sets
CATCHEK	Y or N	ICF catalog is checked for data set
MIGMSG	Y or N	If set to Y reports that data set is migrated
UNCATMSG	Y or N	If set to Y reports that the data set is uncatalogued

BRNVCAT8 – INCLDSN testing utility

Because masking rules can be complex and it is not always clear which data sets are selected for a given mask, you can test masking results with the BRNVCAT8 utility before using an INCLDSN member in production.

Use the CNTL(BRNVCAT8) member to test masking results. The data sets selected by the masking rules in USERMAC(INCLDSN) are listed in the INCFILE DD output.

```
//*=======================================================
//*
//* STANDALONE JOB TO RUN BRNVCAT8 TO TEST INCLDSN
//*
//* INCFILE WILL LIST THE DATASETS RESULTING FROM
//* THE MASKING SPECIFIED IN INCLDSN
//*
//*=======================================================
//PS10 EXEC PGM=BRNVCAT8
//STEPLIB DD DISP=SHR,DSN=HLQ.LIBRARY
//INCLDSN DD DSN=HLQ.USERMAC(INCLDSN),DISP=SHR
//INCFILE DD SYSOUT=*,
// DCB=(RECFM=FB,LRECL=176,BLKSIZE=0)
//ERRLOG DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//VFIPARMS DD DSN=HLQ.USERMAC(VFIPARMS),DISP=SHR
```

Figure 99. BRNVCAT8 utility

CRITREMV – Removing critical data sets entries

Use the CRITREMV utility to remove critical data set entries from the critical file list when specific files are specified in the control cards.

The control cards can specify a file and an optional application ID, APPLID. A report is generated to list the critical files that are removed from the IZBR.REPORTS file. This process can also be run in Simulate mode. A report is generated but files are not removed from the critical file list.

Refer to the JCL in CNTL(CRITREMV). The following statements are in the control card. Specify a data set name or optional APPLID to remove critical file entries.

```
//DDIN DD *
DSNAME APPLID
```

BKUPRELD, BKUPRBLD - Recovering a corrupted Backup Table

If the Backup Table (BKUPTBL) becomes corrupted, you can recover it using one of the following methods, depending on the situation that occurred

- BKUPRELD – If no backups have been done since the last unload of the database, run CNTL(BKUPRELD) after reallocating the BKUPTBL with CNTL(BKUPDEFN).
- BKUPRBLD – If the BKUPTBL was modified since its last backup, run CNTL(BKUPRBLD) to add the entries as well.

Reloading the Backup Table (BKUPRELD)

You can use the BKUPRELD process (an IDCAMS procedure) to reload the Backup Table from the output of the BKUPUNLD utility.

About this task

To reload the Backup Table:

Procedure

1. First define the BKUPTBL using the CNTL(BKUPDEFN) utility.
2. Then the CNTL(BKUPRELD) utility uses the standard IDCAMS RELOAD command to populate the database.

Recreating entries in the Backup Table (BKUPRBLD)

The BKUPRBLD utility rebuilds the BKUPTBL using the data from the current production Backup Table Audit data set (BKUPAUD).

Before you begin

Before you can recover the Backup Table, run CNTL(BKUPUNLD) every day to back up the BKUPTBL.

About this task

To recover the BKUPTBL:

Procedure

1. Run the CNTL(BKUPDEFN) utility to define the BKUPTBL.
2. Run CNTL(BKUPRELD) to load the new BKUPTBL with records from the last backup of the BKUPTBL.
3. Run CNTL(BKUPRBLD) to add records from the current BKUPAUD to the new BKUPTBL.

Results

At this point, the recovered BKUPTBL is current.

IBM z/OS Dataset Mobility Facility Integration Installation and Use

The IBM Z processor family uses pervasive encryption to allow the enterprise to encrypt data while in-flight and also at-rest, thereby reducing costs and helping achieve regulatory compliance.

One part of the overall pervasive encryption implementation is the use of z/OS data set encryption through DFSMS access methods. A significant challenge for an enterprise can be determining the best candidates for data set encryption, as well as the actual process of encrypting these data sets.

IBM Z Batch Resiliency can help identify data sets associated with specific applications through use of analytics,including identifying interdependencies with other applications. IBM Z Batch Resiliency can also create the control cards and JCL that are used by IBM z/OS Dataset Mobility Facility (zDMF) to perform the encryption process.

This section provides instructions for installing and using the IBM Z Batch Resiliency interface with IBM z/OS Dataset Mobility Facility.

Version 1 of this interface, provided with the PTF for APAR PH25241, is the capability to generate control cards and JCL to encrypt non-extended and extended data sets. The generated control cards are grouped by application. In addition, for very large applications, the output can be load balanced based on a target maximum of data (MB) in each group.

Installation and Customization

The following steps provide instructions for installing and customizing the interface.

Procedure

Run HLQ.CNTL(ZDMFALLC) to allocate HLQ.ZDMF.DSNLIST, HLQ.ZDMF.DSNLISTX, HLQ.ZDMF.REPORTS, HLQ.ZDMF.JCL and HLQ.ZDMF.JCLX PDSE data sets used by the process described in the topic Using the interface to IBM z/OS Dataset Mobility Facility.

Using the interface to IBM z/OS Dataset Mobility Facility

Now that you have completed the installation, you are ready to begin the analysis of your data and generate the JCL and control cards for IBM z/OS Dataset Mobility Facility.

The first step is to run DCOLLECT to gather information about your data sets that are used in conjunction with the data already collected from SMF and your scheduler in IBM Z Batch Resiliency. A sample DCOLLECT job is included in HLQ.CNTL(IZBRDCOL).

This DCOLLECT job, as distributed, is just a sample. There are other options and parameters that you may want to use to enable the DCOLLECT to run more efficiently in your environment. As distributed, this job selects all DASD volumes. You may want to use the STORAGEGROUP parameter to select specific storage groups.

If you have already run DCOLLECT and have a file that contains 'D' records you can use that file as input to this process.

You are now ready to run the analysis job to generate the control cards and the JCL

Generating the JCL and output reports

To create the JCL and generate output reports, follow these instructions.

Procedure

1. Run HLQ.CNTL(ZDMFIZBR). This job creates JCL and control cards for data sets that have supported block sizes for IBM z/OS Dataset Mobility Facility.

   The following table outlines the output from this job:

Table 52. HLQ.CNTL(ZDMFIZBR) output

Output	Description
HLQ.ZDMF.JCL	Each member is the JCL to run zDMF to encrypt non-extended data sets
HLQ.ZDMF.JCLX	Each member is the JCL to run zDMF to encrypt extended data sets

Table 52. HLQ.CNTL(ZDMFIZBR) output (continued)	
Output	**Description**
HLQ.ZDMF.DSNLIST	Members matching the ZDMFJCL library containing the control cards
HLQ.ZDMF.DSNLISTX	Members matching the ZDMFJCX library containing the control cards
HLQ.ZDMF.DSNLIST(CROSSAPP)	Non-extended data sets that appear in multiple application schedules
HLQ.ZDMF.REPORTS(ZDMFNOTX)	Report showing totals of non-extended data sets
HLQ.ZDMF.REPORTS(ZDMFNOX)	Report showing extended data sets details
HLQ.ZDMF.REPORTS(ENCRYPT)	Report listing all the data sets that are already encrypted
HLQ.ZDMF.REPORTS(ZDMFX)	Report listing all data sets that are extended
HLQ.ZDMF.REPORTS(ZDMFXT)	Report showing totals of extended data sets
HLQ.ZDMF.REPORTS(BLKSIZNS)	DATASET BLOCKSIZE not supported by zDMF

2. Run HLQ.CNTL(ZDMFBSIZ). This job runs against the inventory database to provide a report on data sets that have an unsupported block size for IBM z/OS Dataset Mobility Facility. This job uses members #ZDMFBSZ and ZDMFBSZ in the HLQ.USERMAC library.

   The report output from this job is as follows:

Table 53. HLQ.CNTL(ZDMFBSIZ) report output	
Output	**Description**
HLQ.ZDMF.REPORTS(BLKSIZNS)	DATASET BLOCKSIZE not supported by zDMF

Inventory Data Set Report

The INVREPT job provides a complete listing of all data sets, critical and non-critical, accessed by an application.

The report format depends on the list of variables to be printed. These and other report parameters are specified in the USERMAC(INVRPTR)member.

The following options may be specified in USERMAC(INVREPT) or on the EXEC JCL statement to report on select data sets in the INVENTORY database:

Table 54. Inventory data set report		
Option	**Parameter**	**Description**
APPLID=	Applid or Mask	Report only on specific or masked (DFDSS masking) Application ID
BKUPMETH=	Bkupmeth or Mask	Report only on specific or masked (DFDSS masking) Backup Methodology
STATUS=	Critical Status Value	Report only on specified Critical Status value (1, 2, 3 or 4)
STATUS<	Critical Status Value	Report only on less than specified Critical Status value (2, 3, or 4)
STATUS>	Critical Status Value	Report only on greater than specified Critical Status value (1,2, or 3)

Table 54. Inventory data set report (continued)		
Option	**Parameter**	**Description**
STATUS<=	Critical Status Value	Report only on less than or equal to specified Critical Status value (1, 2, 3, or 4)
STATUS>=	Critical Status Value	Report only on greater than or equal to Specified Status value (1, 2, 3, or 4)

The default options are as follows:

- APPLID=* - Report on data sets for all applications
- BKUPMETH=* - Report on data sets for all Backup methodologies
- STATUS=<=4 - Report on data sets for all Critical Status

In addition to the options in USERMAC(INVREPT), filtering for INVREPT may be accomplished by coding in the user exit specified in USERMAC(XINVREPT). Sample code is provided as comments in USERMAC(XINVREPT).

```
                                IZBR - IBM Z BATCH RESILIENCY
                  5698-BR1 (C) Copyright Teracloud S.S. 1993, 2020 and IBM Corp. 2020
                               INVREPT: LISTING OF INVENTORY DATASETS BY APPLID
-------------------------------------------------APPLID : CHECKING-------------------------------------------------

DATASET NAME                STATUS  FILETYPE  MBYTES  FIRSTJOB  FIRSTDD  FIRSTACC  FIRST DATE/TIME      LASTRJOB
__________________________  ______  ________  ______  ________  _______  ________  ___________________  ________
PROD.TCS.DDA.BR14003.PO          2  PDS          0.4  DDAJD007  SYSUT2   OUTPUT    09/10/2019 11:58:28  DDAJD023
PROD.TESTING.DDA.BR14001.PS      3  NON-VSAM     0.3  DDAJD001           DELETE    09/10/2019 11:40:13  DDAJD023
PROD.TESTING.DDA.BR14002.PS      3  NON-VSAM     0.3  DDAJD001           DELETE    09/10/2019 11:40:14  DDAJD023
PROD.TESTING.DDA.BR14003.PS      3  NON-VSAM     0.3  DDAJD002           DELETE    09/10/2019 11:43:17  DDAJD023
PROD.TESTING.DDA.BR14004.PS      3  NON-VSAM     0.3  DDAJD002           DELETE    09/10/2019 11:43:17  DDAJD023
PROD.TESTING.DDA.BR14005.PS      3  NON-VSAM     0.0  DDAJD999           DELETE    09/10/2019 11:31:09  DDAJD999
PROD.TESTING.DDA.BR14006.PS      3  NON-VSAM     0.0  DDAJD002           DELETE    09/10/2019 11:43:18  DDAJD002
PROD.TESTING.DDA.BR14007.PS      3  NON-VSAM     0.3  DDAJD002           DELETE    09/10/2019 11:43:18  DDAJD023
PROD.TESTING.DDA.BR14008.PS      3  NON-VSAM     0.3  DDAJD002           DELETE    09/10/2019 11:43:18  DDAJD023
PROD.TESTING.DDA.BR14009.PS      3  NON-VSAM     0.3  DDAJD002           DELETE    09/10/2019 11:43:19  DDAJD023
PROD.TESTING.DDA.BR14010.PS      3  NON-VSAM     0.3  DDAJD002           DELETE    09/10/2019 11:43:19  DDAJD023
PROD.TESTING.DDA.FILEIN.PS       1  NON-VSAM     0.1  DDAJD018  SYSUT1   INPUT     09/10/2019 13:17:10  DDAJD023
PROD.TESTING.DDA.GDG001.PS       4  GDG-BASE     0.1  DDAJD007           NEW       09/10/2019 11:58:21  DDAJD023
PROD.TESTING.DDA.GDG002.PS       1  GDG-BASE     5.3  DDAJD001           NEW       09/10/2019 11:40:20  DDAJD023
```

Figure 100. INVREPT sample report

Resolving Application Assignments for Data Set Encryption

The integration with IBM z/OS Dataset Mobility Facility uses the same APPTABLE as the base IBM Z Batch Resiliency functions.

If you already have IBM Z Batch Resiliency up and running in production you probably have most data sets assigned correctly.

If you are just beginning the installation of IBM Z Batch Resiliency, it may take some time to have all of your data sets assigned to the correct application or applications and accounted for in this process.

To help you resolve any conflicts, the CROSSAPP members generated above identify the data sets that require additional attention.

Run HLQ.CNTL(INVBDUPS) to get detailed information on the schedules and applications for those data sets in the CROSSAPP members generated above. You need to assign data sets to one application for encryption ownership.

Review each member in HLQ.ZDMF.DSNLIST and HLQ.ZDMF.DSNLISTX verifying that the application assignments are correct.

IBM z/OS Dataset Mobility Facility Customization and Usage

Review HLQ.ZDMF.JCL and HLQ.ZDMF.JCLX libraries to ensure the data sets are assigned to the correct application ownership for encryption.

Load Balancing

For large applications with large amounts of data to be encrypted, you can modify HLQ.USERMAC(APPLDEFZ) to load balance the data sets in each schedule/application into multiple members.

In most cases, the JCL and control cards are ready to start encrypting files. Once you have validated the data sets are assigned to a single application and all data sets requiring encryption are accounted for, you are ready to run the encryption JCL.

First, ensure that the IBM z/OS Dataset Mobility Facility started task is running. Then submit the JCL and monitor the results accordingly.

If you need to modify the JCL or control cards, you can modify the templates. The IBM z/OS Dataset Mobility Facility templates you need to modify are:

- The template for HLQ.ZDMF.DSNLIST and HLQ.ZDMF.DSNLISTX is HLQ.USERMAC(AZDMFDSN)
- The template for HLQ.ZDMF.JCL is HLQ.USERMAC(AZDMFJCL)
- The template for HLQ.ZDMF.JCLX is HLQ.USERMAC(AZDMFJCX)

Chapter 11. Viewing IBM Z Batch Resiliency Databases and Reports

You can use the ISPF panels to view IBM Z Batch Resiliency databases and reports online. You can also generate reports using the JCL for each report in the HLQ.CNTL library.

The following topics are discussed in this section:

- Viewing IBM Z Batch Resiliency databases using ISPF
- Generating IBM Z Batch Resiliency reports
- Accessing reports using ISPF panels

Viewing IBM Z Batch Resiliency databases using ISPF

You can view IBM Z Batch Resiliency databases using the ISPF Inquiry panels.

The ISPF panels provide access to information for each of the following IBM Z Batch Resiliency databases:

- **Critical File List** – contains a list of critical files, a subset of the Inventory database.
- **Backup Table** (BKUPTBL) – contains backup information about files that have been backed up.
- **Inventory Database** (INVENTxxxx) – contains files used by applications including file size, file usage, and jobs that access and update these files. You can view the Inventory database by base records, component records, or INCLDSN records.
- **Scheduling Database** – provides Backup Dynamic Dataset scheduling information for data sets (viewed from the Inventory database).
- **Data Set History** (HISTDS) – contains data set usage by application jobs from history database.
- **Job History** (HISTJOB) – contains application job information from the History database.

Accessing the ISPF Panels

Use the ISPF Inquiry panels in IBM Z Batch Resiliency to view information about databases.

About this task

To access the ISPF panels:

Procedure

1. From the ISPF environment, run:

```
TSO EX 'HLQ.CNTL(BRN110)'
```

2. From the IBM Z Batch Resiliency Primary Panel, select OPTION 6 INQUIRY.

© Copyright IBM Corp. 2019, 2020

```
---------------------------- IZBR 1.1.0 Primary Panel ------------------------
OPTION  ===> 6


   .  0 PROFILE       - Define IZBR Profile Information       MVSID: HBB7708
   .  1 INSTALLATION  - Perform IZBR Installation             SYSID: SYS1
   .  2 OVERVIEW      - Review IZBR System Documentation      TIME : 09:30:50
   .  3 EXECUTE       - Execute IZBR Utilities and Reports
   .  4 REPORTS       - Access IZBR Reports Online
   .  5 MAINTENANCE   - Maintain IZBR Parameters and Tables
   .  6 INQUIRY       - IZBR Online Inquiry Functions
   .  7 MISCELLANEOUS - Miscellaneous Functions
   .  8 TLN           - TimeLiner Functions

                                          IIII  ZZZZZZ  BBBBB   RRRRR
                                          II   Z   ZZ   B  BB   R  RR
                                         II      ZZ    B  BB   R  RR
                                        II     ZZ     BBBB    RRRR
                                       II    ZZ      B  BB   R RR
                                      II   ZZ   Z    B  BB   R  RR
                                     IIII  ZZZZZZ  BBBBB   RRR  RR

     5698-BR1 Copyright (C) Teracloud S.A. 1998-2019 and IBM Corp. 2019
```

Figure 101. IBM Z Batch Resiliency Primary Panel

3. From the Online Inquiry Menu, select the option for the database you want to view.

```
                        IZBR Online Inquiry Menu
Command ===>

   OPTION
       1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry


Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 102. IBM Z Batch Resiliency Online Inquiry Menu

Detail for each IBM Z Batch Resiliency database is described in the following sections.

Viewing the Critical File List

The Inventory database contains information for each file that has been discovered by the Analysis process to be critical.

About this task

Critical files, along with the files defined in the USERMAC(INCLDSN) member, are those needed for recovery. It is the primary input to the Backup JCL generation process.

Note: The list of critical data sets can also be viewed directly from the Inventory database. See Viewing the Critical File list for more information.

To view the critical files using the ISPF panels:

Procedure

1. Select option 6, Inquiry from the IBM Z Batch Resiliency Primary Panel.
2. From the Online Inquiry Menu, select option 1, CRITICAL – Browse Critical File List.

```
                          IZBR Online Inquiry Menu
Command ===>

   OPTION
   1   1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry


Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 103. IBM Z Batch Resiliency Online Inquiry Menu

All of the entries in the Critical File List are displayed.

3. To locate a specific entry, use the FIND and SORT commands.

   The FIND command has the syntax:

   ```
   F DSN dsname
   ```

   or

   ```
   FIND APPLID application
   ```

   The SORT command has this syntax:

   ```
   SORT APPLID
   ```

   or

   ```
   SORT APPLID DSN
   ```

Viewing the Backup Table

The Backup Table contains information about critical files, and selectively included files from the USERMAC(INCLDSN) member, that have been backed up.

About this task

To view the Backup Table:

Procedure

1. Starting at the IBM Z Batch Resiliency Primary Panel, select option 6, INQUIRY.
2. From the Online Inquiry Menu, select option 2, BKUPTBL – Backup Table Inquiry.

```
                          IZBR Online Inquiry Menu
Command ===>

   OPTION
   2   1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry


Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 104. IBM Z Batch Resiliency Online Inquiry Menu

3. From the Backup Table Filter Table, enter your selection criteria.

   For example, to view all data sets that start with a specific string enter the equal sign (=) compare operator and then a data set name, as shown in the following example.

   Note: For a description of compare operators, press PF1.

```
------------------------ IZBR – Backup Table Filter -------------------------------------
COMMAND ===>                                                    Scroll ===> CSR
Query limit: 5000   records
                                                                Rows 1 to 21 of 25

Column                     Comp Compare Value
Name                       Oper 1---------10---------20---------30---------40---------50-
Data Set Name . . . . . . . =___ PROD.ACCT.TEST_____________________________________________
Application id. . . . . . . ____ _____________________________________________________________
Sub application id. . . . . ____ _____________________________________________________________
Backup method . . . . . . . ____ _____________________________________________________________
Backup date . . . . . . . . ____ _____________________________________________________________
Backup time . . . . . . . . ____ _____________________________________________________________
Backup date-time. . . . . . ____ _____________________________________________________________
Backup flags. . . . . . . . ____ _____________________________________________________________
User info . . . . . . . . . ____ _____________________________________________________________
Change date . . . . . . . . ____ _____________________________________________________________
Change time . . . . . . . . ____ _____________________________________________________________
Change date-time. . . . . . ____ _____________________________________________________________
Set date. . . . . . . . . . ____ _____________________________________________________________
Set time. . . . . . . . . . ____ _____________________________________________________________
Set date-time . . . . . . . ____ _____________________________________________________________
Backup job. . . . . . . . . ____ _____________________________________________________________
Backup job #. . . . . . . . ____ _____________________________________________________________
Size (MB) . . . . . . . . . ____ _____________________________________________________________
Seqment count . . . . . . . ____ _____________________________________________________________
```

Figure 105. Backup Table Filter

   The Backup Table Display lists all matching data set names, as shown in the following example. The listing includes the application ID, sub-application ID, backup method, and date/time when the data set was backed up.

   You can view or restore individual data sets from the Backup Table Display.

4. To restore individual data sets, enter R in the CMD column. See Restoring specific data sets using ISPF panels.
5. To view detailed information for a data set, enter I in the CMD column, before the data set name you want to view.

   Note: Press PF1 for a description of available commands.

```
------------------------- IZBR – Backup Table Display ------------------------------------
 COMMAND ===>                                                          Scroll ===> CSR
 Data Set Name Filter:
 Cols: 1 to 8 of 24                        Read 13 in 00:00.92   Frame 1: Rows 1 of 13 of 13
     S                             Appl  SubAppl. Backup  Backup     Backup
 CMD R Data Set Name               Id    Id       Method  Date       Time
 ___   PROD.ACCT.BAUDFLAT.G0001V00 LMNO            DSS      2018/07/28  14:52:46
 ___   PROD.ACCT.BAUDFLAT.G0002V00 LMNO            DSS      2018/09/08  14:52:46
 ___   PROD.ACCT.BAUDFLAT.G0003V00 LMNO            DSS      2018/09/08  14:52:46
 ___   PROD.ACCT.BAUDFLAT.G0004V00 LMNO            DSS      2018/09/08  14:52:46
 ___   PROD.ACCT.BAUDFLAT.G0005V00 LMNO            DSS      2018/09/08  14:52:46
 ___   PROD.ACCT.BKUPJCL.DSS       LMNO            DSS      2018/06/28  14:52:12
 ___   PROD.ACCT.RESTJCL.DSS       LMNO            DSS      2018/06/28  14:52:12
 ___   PROD.ACCT.RESTJCL.DSS2      LMNO            DSS      2018/06/28  14:52:12
 ___   PROD.ACCT.SORTED.HIST.LONG  LMNO            ALLOCAT  2018/07/28  14:52:46
 ___   PROD.ACCT.VBCLIST           LMNO            DSS      2018/08/28  14:52:46
 _I_   PROD.ACCT.VSAM.TEST02       LMNO            DSS      2018/09/28  14:52:12
```

Figure 106. IBM Z Batch Resiliency Backup Table Display

The Backup Detail Viewer panel is displayed as shown in the following figure:

```
----------------------- IZBR – Backup Detail Viewer --------------------------------
COMMAND ===>                                                          Scroll ===> CSR
 Rows 1 to 22 of 23                                                  Record 54 of 1290
Column                      Column
Name                        Value
Data Set Name . . . . . . . PROD.ACCT.VSAM.TEST02
Application id. . . . . . . LMNO
Sub application id. . . . .
Backup method . . . . . . . DDS
Backup date . . . . . . . . 2018/09/28
Backup time . . . . . . . . 14:52:12
Backup date-time. . . . . .
Backup flags. . . . . . . .
User info . . . . . . . . .
Change date . . . . . . . .
Change time . . . . . . . .
Change date-time. . . . . .
Set date. . . . . . . . . . 2018/09/28
Set time. . . . . . . . . . 07:18:00
Backup job. . . . . . . . .
Backup job #. . . . . . . .
Size (MB) . . . . . . . . .
Seqment count . . . . . . . 1
-
Backup data set name #1 . . PROD.ACCT.BKUP.DDA,T0400.DSS0001
Device type . . . . . . . . 7804081
```

Figure 107. IBM Z Batch Resiliency Backup Detail Viewer

Viewing the Inventory database

To view the records in the Inventory database:

Procedure

1. Starting at the IBM Z Batch Resiliency Primary Panel, select option 6, INQUIRY.
2. Select one of the following Inventory Database Inquiry options:

Inventory database inquiry option	Description
Base records	Views records that define generation data groups, VSAM clusters, sequential and PDS files.
Component records	Views individual generations of GDGs, data and index components of VSAM clusters and detail about individual sequential files.

Inventory database inquiry option	Description
INCLDSN records	Views the Backup Dynamic Data Set Scheduling include records.

```
                          IZBR Online Inquiry Menu
Command ===>

   OPTION
   3   1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry


Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 108. IBM Z Batch Resiliency Online Inquiry Menu panel

3. For example, to view Inventory Component records, select option 4, INVENTCOMP.
4. From the Inventory Component Table Filter, enter your selection criteria as shown in the following example:

```
---------------------- IZBR – Inventory Component Table Filter ----------------------------
 COMMAND ===>                                                        Scroll ===> CSR
 Query limit: 5000   records
                                                                   Rows 1 to 21 of 25

 Column                      Comp Compare Value
 Name                        Oper 1---------10---------20---------30---------40---------50-
 Data set name . . . . . . . =___ PROD.ACCT.BACKUP_____________________________________
 Application id. . . . . . . ____ _____________________________________________________
 Sub application id. . . . . ____ _____________________________________________________
 Backup method . . . . . . . ____ _____________________________________________________
 Backup date . . . . . . . . ____ _____________________________________________________
 Backup time . . . . . . . . ____ _____________________________________________________
 Backup date-time. . . . . . ____ _____________________________________________________
 Backup flags. . . . . . . . ____ _____________________________________________________
 User info . . . . . . . . . ____ _____________________________________________________
 Change date . . . . . . . . ____ _____________________________________________________
 Change time . . . . . . . . ____ _____________________________________________________
 Change date-time. . . . . . ____ _____________________________________________________
 Set date. . . . . . . . . . ____ _____________________________________________________
 Set time. . . . . . . . . . ____ _____________________________________________________
 Set date-time . . . . . . . ____ _____________________________________________________
 Backup job. . . . . . . . . ____ _____________________________________________________
 Backup job #. . . . . . . . ____ _____________________________________________________
 Size (MB) . . . . . . . . . ____ _____________________________________________________
 Seqment count . . . . . . . ____ _____________________________________________________
```

Figure 109. Inventory Component Table Filter panel

The Inventory Component Detail Viewer displays component level detail for the selected data set.

```
------------------------ IZBR - Inventory Component Detail Viewer ----------------------
COMMAND ===>                                                          Scroll ===> CSR
 Rows: 1 to 21 of 53                                                  Record 62 of 251
Column                  Column
Name                    Value
Data Set Name. . . . . . PROD.ACCT.BKUPTBL.DATA
Base data set name . . . PROD.ACCT.BKUPTBL
Device type. . . . . . . 3030200F
File sequence number . . 0
Volume count . . . . . . 0
First volume serial. . . TS0101
Volume serials . . . . .
Application id . . . . . LMNO
Sub application id . . .
Backup method. . . . . . DSS
File type. . . . . . . . C
Status . . . . . . . . . Unassigned
Base status. . . . . . . Unassigned
Base global status . . .
Base highest status. . . 4 Non-Crit
First access job . . . . SJMKBUP1
First access DD. . . . . BKUPTBL
First access . . . . . . Close
First access disposition
First access date. . . . 2018/02/09
First access time. . . . 17:09:34
```

Figure 110. Inventory Component Detail Viewer panel

Viewing History information

The History database contains information about each file used by the production jobs defined in the APPTABLE.

Before you begin

Before you can view the History database, verify that the VSAM Keyed History file is defined using CNTL(KEYDEFN) and loaded. See IBM Z Batch Resiliency Maintenance for details.

About this task

The number of days in the history file is controlled by the retention period parameter (RETPD1), in the USERMAC(VFIPARMS).

Two options are available:

- **HISTDS** – Data Set History Inquiry
- **HISTJOB** – Job History Inquiry

Procedure

1. Start at the IBM Z Batch Resiliency Primary Panel, and select option 6, INQUIRY.
2. From the IBM Z Batch Resiliency Online Inquiry Menu, select option 7, HISTDS.

```
                          IZBR Online Inquiry Menu
Command ===>

   OPTION
   7   1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry


Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 111. Online Inquiry Menu

3. From the Data Set History Table Filter, enter your selection criteria as shown in the following example.

   Note: Press PF1 for a description of compare operators.

```
---------------------- IZBR – Data Set History Table Filter --------------------------
COMMAND ===>                                                        Scroll ===> CSR
Query limit: 5000   records
                                                                  Rows 1 to 21 of 25

Column                      Comp Compare Value
Name                        Oper 1---------10---------20---------30---------40---------50-
Data Set Name . . . . . . . =___ PROD.ACCT.TEST_____________________________________________
Job name. . . . . . . . . . ____ _____________________________________________________________
System ID . . . . . . . . . ____ _____________________________________________________________
Disposition . . . . . . . . ____ _____________________________________________________________
Access type . . . . . . . . ____ _____________________________________________________________
Device type . . . . . . . . ____ _____________________________________________________________
Data set type . . . . . . . ____ _____________________________________________________________
Reader date . . . . . . . . ____ _____________________________________________________________
Reader time . . . . . . . . ____ _____________________________________________________________
Reader date-time. . . . . . ____ _____________________________________________________________
Record date . . . . . . . . ____ _____________________________________________________________
Record time . . . . . . . . ____ _____________________________________________________________
Record date-time. . . . . . ____ _____________________________________________________________
Ddname. . . . . . . . . . . ____ _____________________________________________________________
Volume serial . . . . . . . ____ _____________________________________________________________
SMF record type . . . . . . ____ _____________________________________________________________
```

Figure 112. Data Set History Table Filter panel

The Job History Data Set Display returns information based on your entry, as shown in the following example:

```
-----------------------------IZBR – Job History Data Set Display -----------------------
COMMAND ===>                                                          Scroll ===> CSR
Job name    : GL0001            System        : SYS1
Job number  : JOB03465          Start date    : 2018/10/23 09:28:10
Return code : 47                End date      : 2018/10/23 09:28:10
   Record      Record   Job/step name                    Pgm name D A D T SMF SMF
CMD Date        Time     Dsname                          Ddname   P C V Y Typ SRT
___ 2018/10/23  09:28:10 GLOOO1                          STEPLIB  0 I D .  14
___ 2018/10/23  09:28:10 :: GL0001  Job Initiation   ::                    30 255
___ 2018/10/23  09:28:10 ** VSXMAI  Step termination ** rc: 47 VSXMAI      30   2
___ 2018/10/23  09:28:10 ** GL0001  Job termination  ** rc: 47             30   1
```

Figure 113. Job History Data Set Display

4. To view details for a data set, enter `I` to display the History Data Record Display panel.

```
                  History Data Record Display

COMMAND ===> _____________________________________________________________

Press F3 to return to a previous panel

Record Type   64         Job Name : ATMJD023
_____________________________________________________________________________
System ID  . :  SOW1
Reader Start :  2019/01/28   05:58:42
Open time  . :  2019/01/28.  05:58:59
Close/EOV  . :  2019/01/28.  05:58:59
_____________________________________________
Ddname  .  .  . : SYSUT2
Dsname  .  .  . : PROD.TST.ATM.BRI14008.PS
_____________________________________________________________________________
Number of              First
volumes    .  . : 1    volser  .  :  PRD001
_____________________________________________________________________________
SMF14RIN   :  20      JFCBIND1 : 00   JFCBIND2 :  80
```

Figure 114. History Data Record Display

Displaying Backup Dynamic Dataset Scheduling information

If you are using Backup Dynamic Dataset Scheduling, you can check the schedule database to see if a data set has been backed up or is scheduled to be backed up.

Procedure

1. Starting at the IBM Z Batch Resiliency Primary Panel, select option 6, INQUIRY.
2. From the IBM Z Batch Resiliency Online Inquiry Menu, select option 6, SCHEDULING:

```
                         IZBR Online Inquiry Menu
Command ===>

   OPTION
    6  1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry


Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 115. Online Inquiry Menu

The Dynamic Data Set Scheduling Filters panel is displayed.

```
                Dynamic Data Set Scheduling Filters Selection
Command ===>                                                 Scroll ===> PAGE

                                                        Search Limit ===> 1000
Dsname  ===>  GENLEDGE.CHART*
ApplId  ===>  *          SubAppl ===>  *          Status ===>   EQ  *
Scheduled =>  Y (Y|N)    DB Records=>  C (Comp|Incl)

 ---------------------- Filters for Scheduled data sets ----------------------
Select any or all:    Y Eligible, Backup Not yet submitted
                      Y Eligible, Backup Pending
                      Y Eligible, Backup Completed
                      Y Non-eligible, Cutoff Occurred
                      Y Non-eligible, never made eligible
 ------------------------------------------------------------------------------

Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 116. Dynamic Data Set Scheduling Filters panel

3. Enter the Data Set Name either as a fully qualified name or use DFDSS-style masking to retrieve multiple data sets.
4. Enter the application name in the **APPLID** field, use masking, or place an asterisk in this field to retrieve data sets for all applications.
5. Enter an asterisk (*) in the **Sub-Application Id** and **Status** fields to specify all values for those fields.
6. Enter Y in the **Scheduled** field to select files that are scheduled to be backed up; N in this field selects files that are not scheduled to be backed up.
7. In the **DB Records** field, enter a C to select Component Inventory records or I to select records to be backed up from INCLDSN records.
8. Select the filters to apply for the scheduled backup status. Enter Y or N next to the appropriate status.

```
Command ===>      Dynamic Data Set Scheduling Records Row 1 to 4 of 4    Scroll ===> PAGE

Filters:  Dsn  . . : GENLEDGE.CHART*
          Appl . . : *         SubAppl  : *         Status : EQ *
          Sched  . : Yes(BNS,BSU,BP,BC,CO,NE)
          DB recs  : Comp.

                                                          Backup  S Sch
S Dsname                                    Applid   Sub-Appl Method  T Sta
- ------------------------------------------ -------- -------- ------- - ---
_ GENLEDGE.CHART.DATA                       GL                DSS     1 BNS
_ GENLEDGE.CHART.DATA                       EIS               DSS     1 BNS
_ GENLEDGE.CHART.INDEX                      GL                DSS     1 BNS
_ GENLEDGE.CHART.INDEX                      EIS               DSS     1 BNS
```

Figure 117. Dynamic Data Set Scheduling Records panel

The Scheduling Status field (Sch Sta) has several values:

- BNS: Eligible Backup not yet submitted
- BP: Eligible Backup pending
- BC: Eligible Backup completed
- CO: Non-eligible cutoff occurred
- NE: Never made eligible
- NS: Not scheduled

9. Select the first entry (with an S) to display content similar to the following:

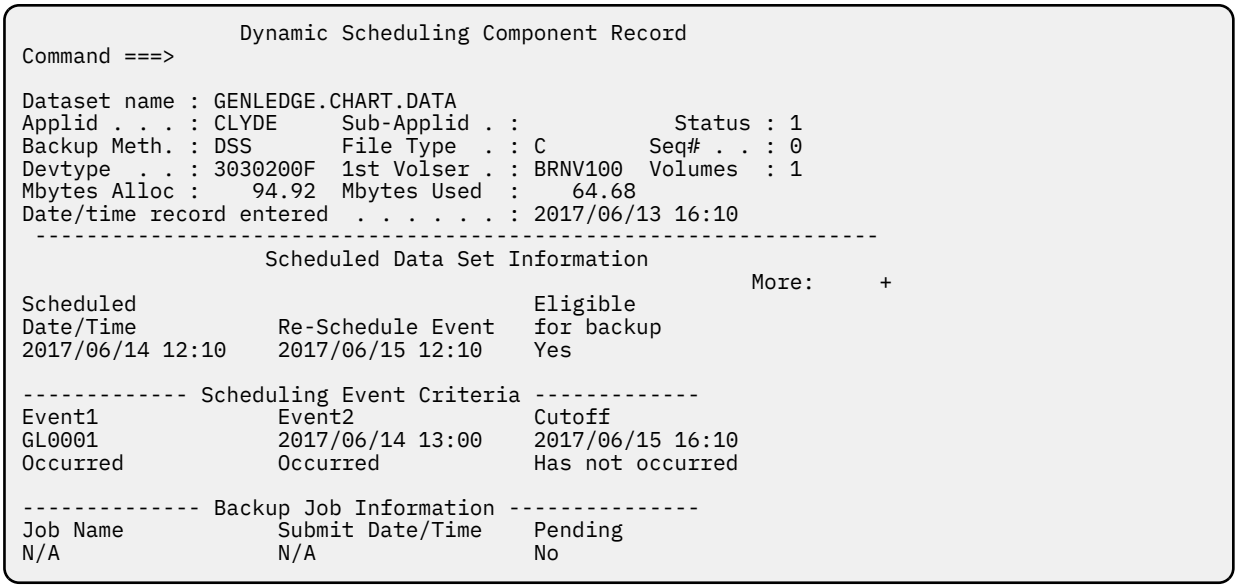


Figure 118. Dynamic Scheduling Component Record panel

Generating IBM Z Batch Resiliency reports

You can generate IBM Z Batch Resiliency reports by submitting the JCL in the CNTL library. In most cases, the member name in the library correspond to the report that is output.

The following table describes the JCL and description for each report:

Table 55. Report JCL descriptions

JCL	Report information
BACKUP	Generates three backup reports: • BACKWO – Critical data sets without backups • BACKUP – Data sets with a backup and data sets without a backup • BACKCMFT - Comfort Index report
CONCATRP	Jobs that have concatenated files defined in the JCL
CRITREPT	Complete listing of all files critical to an application
CROSSDEP	Identifies files updated or accessed across multiple applications
CYCLEMGT	Jobs that were run during an application cycle. Includes the start and end time for the job, the JES job number and the job count within the cycle
DORMANT	Critical files that are no longer in use
HISTRPT	Select data set records, or all data set records in the HISTORY database
INVTYRPT	Select data set records, or the entire INVENTRY database
JOBRPT	All jobs and the files that each job references

You can also generate reports using the ISPF panels using OPTION 3, Execute IBM Z Batch Resiliency Utilities and Reports.

Accessing reports using ISPF panels

After the reports are generated by the JCL, you can view, print, delete, or rename a report.

About this task

To access the reports using the ISPF panels:

Procedure

1. From the ISPF environment, run:

```
TSO EX HLQ.CNTL(BRN110)
```

2. From the IBM Z Batch Resiliency Primary Panel, select OPTION 4 REPORTS to access the IBM Z Batch Resiliency reports.

   The Reports panel displays a list of available reports as shown in the following example.

```
----------------------------- IZBR 1.1.0 Reports -------- Row 1 to 39 of 310
Command ===>                                                Scroll ===> PAGE

    Sel    -Report-     ---Date---     --Time--     --Job---
     .     ABARVFY      LATEST         LATEST       N/A
     .     BACKJCL      LATEST         LATEST       N/A
     .     BACKJCL      06/08/2018     06:33:11     V4XFBJCL
     .     BACKJCL      06/08/2018     06:28:42     V4XFBJCL
     .     BACKJCL      06/08/2018     06:25:41     V4XFBJCL
     .     BACKJCL      06/08/2018     06:13:08     V4XFBJCL
     .     BACKJCL      06/08/2018     06:01:29     V4XFBJCL
     .     BACKJCL      06/08/2018     05:56:23     V4XFBJCL
     .     BACKJCL      06/08/2018     05:35:03     V4XFBJCL
     .     BACKJCL      06/08/2018     05:27:08     V4XFBJCL
     .     BACKJCL      06/08/2018     05:20:56     V4XFBJCL
     .     BACKJCL      06/07/2018     17:31:01     V4XFBJCL
     .     BACKJCL      06/07/2018     17:25:33     V4XFBJCL
```

Figure 119. Reports panel

Outputting reports using BRNYPRNT

The BRNYPRNT utility adds the name to reports and makes duplicates for ISPF.

For a single report, only the current copy of the report is saved, using the same name in CNTL. When multiple reports are produced, the current report and the previous output from specific report jobs are retained.

BRNYPRNT stores a report as a member in the REPORTS data set. It keeps track of all report output in the $INDEX member of the same file. BRNYPRNT saves a current report as its original jobname. BRNYPRNT archives a copy of the same report to a unique name that begins with the number sign (#), for example, #13C2467. It exclusively enqueues the REPORTS($INDEX) member so no updates occur while the multi-reporting procedure runs.

You can regulate report generations and report date spans using the USERMAC(VFIPARMS) member parameters RPTGENS (default=5) and RPTRETPD (default=999).

BRNYPRNT Options

The BRNYPRNT step BRNYPRNT procedure uses the following input values.

Table 56. COPYPRINT options

Option	Parameter	Description
MEMBER=	'Report name'	Name of IBM Z Batch Resiliency report.

Table 56. COPYPRINT options (continued)		
Option	**Parameter**	**Description**
NUMBER=	#	NUMBER refers to an IBM Z Batch Resiliency report-specific temporary data set and should not be changed. Default is NUMBER=1.

The BRNYPRNT step is appended to the IBM Z Batch Resiliency report job JCL. The BRNYPRNT program uses the following PARM= values. Default values are underlined.

Table 57. BRNYPRNT PARM values		
Option	**Parameter**	**Description**
BYPASS=	Yes	Process as a single version of a report. Bypass multiple version reporting. Keeps only current version of a specific report.
	No	Maintains multiple versions of a report in the REPORTS database.

See "BRNYPRNT Messages" on page 277 for BRNYPRNT error codes and messages.

Creating a CSV data set via the IBM Z Batch Resiliency ISPF panels

You can create a CSV data set using Option 6 INQUIRY on the IBM Z Batch Resiliency Primary menu.

About this task

From the IBM Z Batch Resiliency Primary Menu, select Option 6 INQUIRY to open the Online Inquiry Menu, shown in the following figure.

```
                        IZBR Online Inquiry Menu
Command ===>

   OPTION
       1.  CRITICAL   - Browse Critical File List
       2.  BKUPTBL    - Backup Table Inquiry
       3.  INVENTBASE - Inventory Database Inquiry – Base Records
       4.  INVENTCOMP - Inventory Database Inquiry – Component Records
       5.  INVENTINCL - Inventory Database Inquiry – INCLDSN Records
       6.  SCHEDULING – Dynamic Data Set Scheduling Inquiry
       7.  HISTDS     – Data Set History Inquiry
       8.  HISTJOB    - Job History Inquiry

Press PF1 for Help, or position a cursor at input field and press PF1 for
field-level Help
```

Figure 120. IBM Z Batch Resiliency Online Inquiry Menu

Within the Online Inquiry Menu, the following options allow you to create a CSV data set from the listed information they display:

- Option 2. BKUPTBL - Backup table inquiry
- Option 3. INVENTBASE - Inventory Database Inquiry - base records
- Option 4. INVENTCOMP - Inventory Database Inquiry - component records
- Option 7. HISTDS - Data set History Inquiry
- Option 8. HISTJOB - Job History Inquiry

Procedure

1. For example, if you select Option 4 INVENTCOMP, the Inventory Component Table Filter will open, as shown in the following figure.

```
Column                        Comp Compare Value
 Name                         Oper 1--------10--------20--------30--------40-->
 Data set name . . . . . .____
_____________________________________________
 Base data set name. . . . .____
_____________________________________________
 Device type . . . . . .____
_____________________________________________
 File sequence number. . . .____
_____________________________________________
 Volume count. . . . . .____
_____________________________________________
 First volume serial . . . .____
_____________________________________________
 Volume serials. . . . . .____
_____________________________________________
 Application id. . . . . .____
_____________________________________________
 Sub application id. . . . .____
_____________________________________________
 Backup method . . . . . .____
_____________________________________________
 File type . . . . . .____
_____________________________________________
 Status. . . . . .____
_____________________________________________
 Base status . . . . . .____
_____________________________________________
 Base global status. . . . .____
_____________________________________________
 Base highest status . . . .____
_____________________________________________
 First access job. . . . . .____
_____________________________________________
 COMMAND ===>                                          Scroll ===> PAGE
  F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cancel
```

Figure 121. Inventory Component Table Filter

2. Use the filter to find and display the information you would like to export; such as an Inventory data set.
3. To create a CSV data set from this information listing, enter the EXPORT command from the command line.
4. Enter a title for the report in the Title field.
5. Enter a data set name in the Dsname field.
6. Ensure both the `Store headers and data in CSV format` and `Replace like-named data set` options are selected.

Example

```
                                        - Export Tool
    Record count 919
    Title MyReport_____________________________________________________________________
    Dsname HLQ.CNTL.DATASET ___________________________________________________________
     / Replace like-named data set
     _ View data set after creation
     / Store headers and data in CSV format
     _ Use maximum column width

     Press ENTER to export data
     Press PF3   to terminate the dialog
      F1=Help    F2=Split   F3=Exit    F9=Swap    F12=Cancel
```

Figure 122. Export Tool

A CSV data set will be created, ready to download, for use by your spreadsheet.

To download the created CSV data set, use the method outlined in “Downloading the sample Excel spreadsheet reports from the IZBR installation package ” on page 191.

Downloading the sample Excel spreadsheet reports from the IZBR installation package

The following information outlines how to download the sample Excel spreadsheet reports included with the IBM Z Batch Resiliency installation package.

Procedure

1. Locate the installed member BRNGRAPH, residing in the *HLQ*.SBRNUMAC data set, where *HLQ* represents the high-level qualifier for the IZBR installation. The following figure displays the *HLQ*.SBRNUMAC data set.

```
           Name     Prompt       Size   Created          Changed          ID
 _________ BRNDISTL
 _________ BRNGRAPH
 _________ BRNUDIST
 _________ BRNUSERM
 _________ BRNUUMAC
           **End**
Command ===>                                                 Scroll ===> PAGE
  F1=Help    F2=Split   F3=Exit    F5=Rfind   F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cancel
```

Figure 123. View Display for SBRNUMAC

2. Select the BRNGRAPH data set member. The Action for Member BRNGRAPH screen opens, as shown in the following figure.

```
                  Action for Member BRNGRAPH
                                          More:     +
 Member Action
 ____    1.  Edit                    8.  Copy
         2.  View                    9.  Reset
         3.  Browse                  10. Open    **None**
         4.  Delete                  11. Submit
         5.  Rename                  12. TSO Cmd
         6.  Print                   13. WS Cmd
         7.  Move                    14. Select VIEW
                                     15. Info

 Prompt Action . .           (For prompt field)
  F1=Help         F2=Split       F3=Exit
  F7=Backward     F8=Forward     F9=Swap
```

Figure 124. Action for Member BRNGRAPH

3. Use a binary method to transfer BRNGRAPH to your workstation as BRNGRAPH.zip.
4. Extract the files in BRNGRAPH.zip to a nominated folder.

What to do next

Your nominated folder will now contain a number of spreadsheet reports and a sample presentation. These reports can now be updated with data from IBM Z Batch Resiliency using the method outlined in “Updating your spreadsheet reports ” on page 192.

Updating your spreadsheet reports

The following task outlines the method used to update the data in your reports from information in IBM Z Batch Resiliency.

About this task

To update your reports in programs such as Microsoft Excel, with the latest information from IBM Z Batch Resiliency, follow the steps outlined here.

Procedure

1. Download the CSV data sets (generated using the method outlined in “Creating a CSV data set via the IBM Z Batch Resiliency ISPF panels ” on page 189) to your workstation.
2. Open the relevant sample report from the /batch or /inquiry folders (as extracted using the method outlined in “Downloading the sample Excel spreadsheet reports from the IZBR installation package ” on page 191.
3. Using your preferred CSV-capable spreadsheet program, open the CSV file that you downloaded using the method in “Creating a CSV data set via the IBM Z Batch Resiliency ISPF panels ” on page 189.
4. Copy the contents of your downloaded CSV file, and either replace or append the data in the 'Data for Pivot Chart' (or equivalent) tab of your report workbook.
5. Refresh your data. In Excel, for example, click on the Data menu and select Refresh All. Your chart should reflect your changes.
6. Repeat steps 1 to 5 above for other required reports.

Example

The following is an example of the INVENTRY_excel report located in the /inquiry folder.

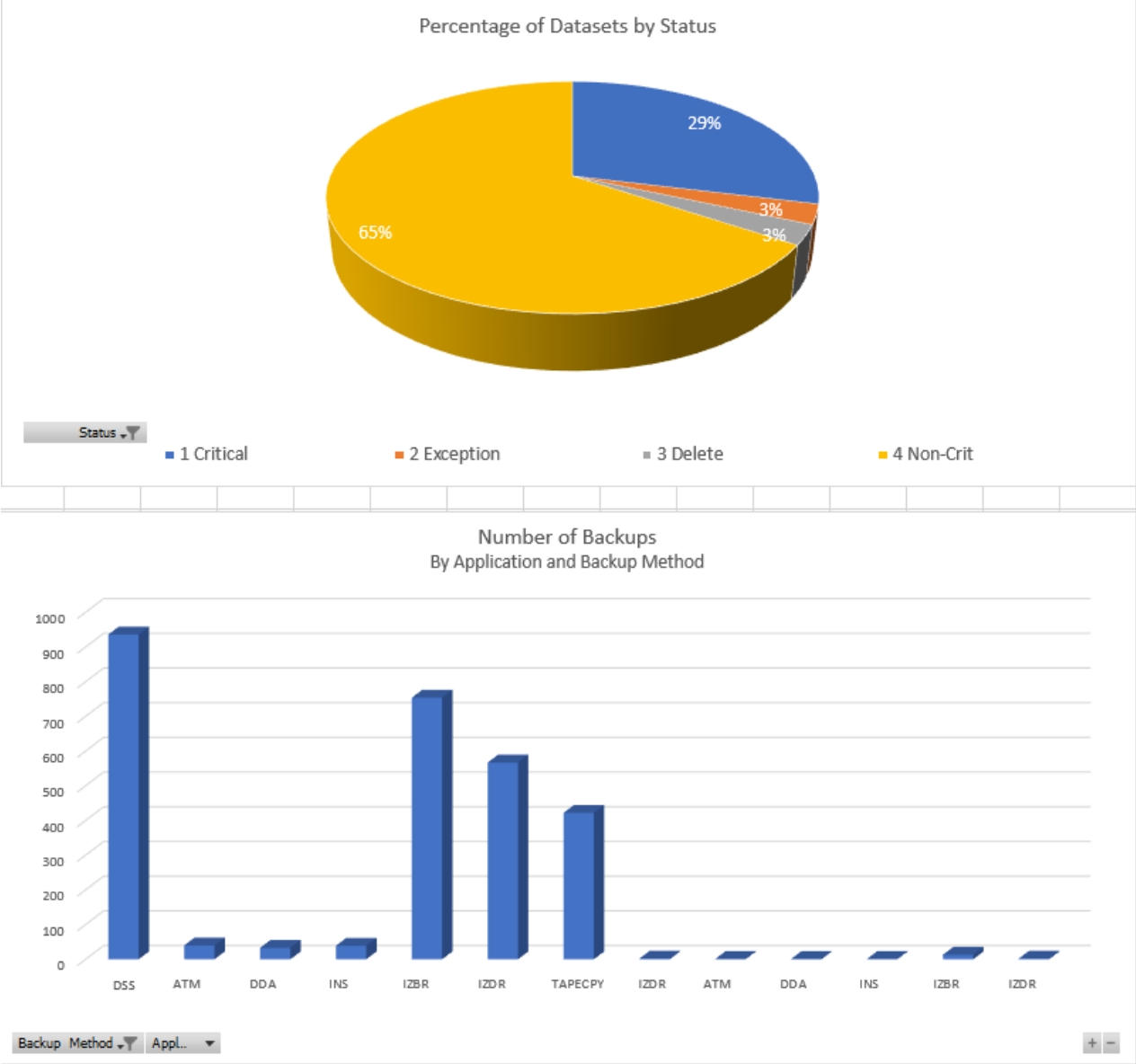


Figure 125. Spreadsheet report example

Jobs report (JOBRPT)

The CNTL(JOBRPT) job produces a report of all jobs and the files that each job references using the History file.

About this task

Use this report to determine the usage of a file or to determine the first job for an application.

Procedure

1. To select a data set, a job or the SMF record type, use the USERMAC(XJOBRPTR) member.
   - The data set or job can be masked.
   - Only one control card is permitted per run.
2. To select all jobs, use an empty USERMAC(XJOBRPTR) member. This is the default.

   The following are five different examples of the control cards:

```
DSN=AP.MASTER.FILE

DSN=AP.DETAIL.**

JOB=PR2345RP

JOB=PR45*

TYPE=14
```

A sample of the JOBRPT report follows.

```
           IZBR - DISASTER RECOVERY/IBM Z BATCH RESILIENCY VERSION 1.1.0
                                                    16 JUN 2018 12:30:31 PAGE: 1
              PROPRIETARY SOFTWARE OF TERACLOUD S.A. AND IBM CORP.
---------------------- DSNAME : AG.WORK4X0C.CNTL.--------------------------------
  SMF DATE, TIME    UPDATE JOBNAME VOLUME RECTYPE DDNAME   VOLUME DATA SET/DISPOSITION
_________________ _____________ ______ _______ ________ ______ ____________________
02/16/2018, 9:59:45 SCXJ          RAID34 INPUT   ISP09343 RAID34 OLD
02/16/2018,10:04:46 SCXJ          RAID34 INPUT   ISP09344 RAID34 OLD
02/16/2018,10:12:23 SCXJ          RAID34 INPUT   ISP09344 RAID34 OLD
02/16/2018,10:12:24 SCXJ          RAID34 OUTPUT  ISP09344 RAID34 OLD
02/16/2018,10:12:29 SCXJ          RAID34 INPUT   ISP09344 RAID34 OLD
02/16/2018,10:12:32 SCXJ          RAID34 INPUT   ISP09343 RAID34 OLD
02/16/2018,10:17:24 SKWB          RAID34 INPUT   ISP10163 RAID34 OLD
```

Figure 126. JOBRPT report

Critical data set report (CRITREPT)

The CNTL(CRITREPT) job provides a complete listing of all data sets critical to an application.

Use the CRITREPT report to verify the critical data sets with applications personnel, or to simply list all critical data sets.

The report format depends on the list of variables to be printed. These and other report parameters are specified in the USERMAC(CRITRPTR) member.

A sample of CRITREPT follows.

```
            IZBR - DISASTER RECOVERY/IBM Z BATCH RESILIENCY VERSION 1.1.0     22 Apr 2018 09:10:09    PAGE: 1
                  PROPRIETARY SOFTWARE OF TERACLOUD S.A. AND IBM CORP.
                        CRITRPT: LISTING OF CRITICAL DATA SETS BY APPLID

----------------------------------------- APPLID : ACCTPAY  ------------------------------------------------
DATA SET NAME                 STATUS FILETYPE MBYTES FIRSTJOB FIRSTDD  FIRSTACC LASTRJOB FIRST DATE/TIME
----------------------------- ------ -------- ------ -------- -------- -------- -------- -------------------

AG.WORK4X0L.LIBRARY                1 PDS        11.9 SBAPCHK1 STEPLIB  INPUT    SBAPCHK1 04/16/2018,17:21:53
AG.WORK4X0L.REPORTS                1 PDS         0.0 SBAP     ISP09026 INPUT    SBAP     04/20/2018, 9:51:31
AG.WORK4X0L.SBRNSRCE               1 PDS       1.1.0 SBAPCHK1 INFILE   INPUT    SBAPCHK1 04/16/2018,17:21:48
AG.WORK4X0M.BKUPAUD                1 CLUSTER    10.3 SBAPBALC BKUPAUD  CLOSE    SBAPBALC 04/15/2018,10:21:40
AG.WORK4X0M.BKUPMETH               2 SEQUENTL    0.0 SBAP     ISP08333 OUTPUT   SBAPREST 04/19/2018, 8:59:13
AG.WORK4X0M.BKUPTBL                1 CLUSTER     7.0 SBAPBALC BKUPTBL  CLOSE    SBAPFIX  04/15/2018,10:33:44
AG.WORK4X0M.CNTL                   1 PDS         2.7 SBAP     ISP08257 INPUT    SBAP     04/22/2018, 8:44:45
AG.WORK4X0M.HSMACT                 1 SEQUENTL    5.6 SBAP     HSMACT   INPUT    SBAP1    04/20/2018,10:48:18
AG.WORK4X0M.LIBRARY                1 PDS         5.9 SBAP     ISP10445 INPUT    SBAP1    04/20/2018,10:57:30
AG.WORK4X0M.REPORTS                1 PDS         0.0 SBAP     ISP09023 INPUT    SBAP     04/20/2018, 9:50:15
AG.WORK4X0M.RESTJCL.DSS            2 PDS         6.7 SBAPREST SYS00005 OUTPUT   SBAP     04/19/2018, 9:09:17
AG.WORK4X0M.RESTTBL                1 SEQUENTL    8.9 SBAPREST RESTTBL  INPUT    SBAPREST 04/19/2018, 9:03:51
AG.WORK4X0M.SBRNSRCE               1 PDS         3.3 SBAPREST INFILE   INPUT    SBAPREST 04/19/2018, 9:02:12
AG.WORK4X0M.USERMAC                1 PDS         4.0 SBAP     IFID     INPUT    SBAP1    04/20/2018,10:47:44
AG.WORK4X0M.SBRNEXEC               1 PDS         2.4 SBAP     ISP10439 INPUT    SBAP1    04/20/2018,10:47:34
AG.WORK4X0M.SBRNMLIB               1 PDS         0.8 SBAP     ISP10443 INPUT    SBAP1    04/20/2018,10:57:30
AG.WORK4X0M.SBRNPLIB               1 PDS         3.2 SBAP     ISP10441 INPUT    SBAP1    04/20/2018,10:57:31
AG.WORK4X0M.SBRNSLIB               1 PDS         0.8 SBAP     ISP10442 INPUT    SBAP1    04/20/2018,10:57:31
```

Figure 127. Critical data sets by APPLID sample report

HISTORY report (HISTRPT)

The CNTL(HISTRPT) job reports on selected records in the History database.

To report on selected data sets, use the SELECT DD statement. When there is no input in the SELECT DD, all data sets are reported.

Application selection is handled by tailoring the APPTABLE statement or copying selected cards from the real APPTABLE or pointing to a PDS member containing the APPTABLE cards.

DSN selection:

```
....+....0....+....0
SYS1.** WILL PROCESS ONLY SYS1 DATA SETS
```

Application selection:

```
....+....0....+....0
DSTEST1 *DSTEST TEST APP
```

Two reports are generated in the REPORTS data set:

- Reports by JOBS/DSN
- Reports by DSN/JOBS

Concatenated files report (CONCATRP)

The CNTL(CONCATRP) job generates a report of jobs that have concatenated files defined in the JCL.

Additionally, an output file (DCOPTFLE) lists the jobs in a format usable by the BRNVCOPT facility to report in detail the jobs and all the concatenated files. This report lists the job and the first data set in the concatenation as well as the number of files in the concatenation.

Cross dependencies report (CROSSDEP)

The CNTL(CROSSDEP) job scans critical data sets for duplicate entries and identifies files that were updated or accessed across multiple applications.

```
DISASTER RECOVERY/IBM Z BATCH RESILIENCY              PAGE:          1
DR/IZBR - VERSION 4.5     TERACLOUD S.A. AND IBM CORP.DATE: 09/27/2017

REPORT NAME: CROSSDEP - LIST OF CRITICAL FILES UPDATED BY MORE THAN ONE
APPLICATION


DSN:  PROD.TESTING.FILEIN.PS
UPDATED BY:  ATM                      REFERENCED BY: ATMJD022 28JUL17 20:42:02
             DDA                                     DDAJ D022 28JUL17 21:42:55

DSN:  PROD.TESTING.INVENTRY
UPDATED BY:  ATM                      REFERENCED BY: ATMJD023 28JUL17 20:43:30
             DDA                                     DDAJD023 28JUL17 21:44:23
             IZBR                                    BRNV1009 28JUL17 22:50:48
```

Figure 128. CROSSDEP report

Inventory records report (INVTYRPT)

The CNTL(INVTYRPT) job generates a report of INVENTRY records.

To report on select data sets use standard masking in the SELECT DD statement. You can include multiple cards included in SELECT DD.

With no selection, the entire INVENTRY file is dumped resulting in a large report.

```
//*-------------------------------------------------------------------
//* THIS JOB WILL PRINT DETAIL RECORDS FROM THE INVENTRY FILE.
//* UNLESS A SELECT OPTION IS PROVIDED, THE ENTIRE FILE WILL PRINT.
//* STANDARD DSN SELECTION RULES APPLY.
//* EG. //SELECT DD *
//*     ABC.**
//* WILL SELECT ALL DATASETS IN THE INVENTRY THAT BEGIN AS ABC.
//*-------------------------------------------------------------------
//PS10         EXEC PGM=BRNVB024
//STEPLIB      DD DSN=HLQ.SCKKLOAD,DISP=SHR
//             DD DSN=HLQ.SBRNLOAD,DISP=SHR
//RPTOUT       DD DSN=&&REPORT,DISP=(,PASS),
//                UNIT=SYSDA,SPACE=(CYL,(25,25)),
//                DCB=(RECFM=FBA,LRECL=133,DSORG=PS,BLKSIZE=0)
//SYSUDUMP     DD SYSOUT=*
//INVENTRY     DD DSN=HLQ.INVENTRY,
//                DISP=SHR,DCB=(BUFNO=10)
//* NEW FILES W/SELECT OPTION
//PARMLIB      DD DSN=HLQ.USERMAC,DISP=SHR
//RTDPRINT     DD SYSOUT=*
//SELECT       DD *
     OPTIONAL DSN SELECT CARDS
/*
//COPYRPT1     EXEC
COPYPRNT,MEMBER=INVTYRPT
```

Figure 129. INVTYRPT JCL

Cycle management report (CYCLEMGT)

The CNTL(CYCLEMGT) job analyzes the History database for the selected application and date. The report lists, in job start time order, the jobs that ran as part of the application cycle.

About this task

Each job run for that cycle is listed along with the start and end time for the job, the JES job number, the job count within the cycle. Additionally, jobs on the critical path for the application are distinguished from the non-critical jobs by an asterisk (*) after the job name for critical path jobs on the report line. A job is considered on the application critical path if the job starts with no prior jobs executing.

At the end of the job, totals are reported for total jobs, number of critical path jobs and the maximum number of application jobs running concurrently.

Procedure

1. To run the job, ensure the LIBRARY, USERMAC, and CRITICAL file names are updated to reflect your installation requirements.
2. Update the JOB and PARM cards.

   There are two formats for the PARM cards based on whether you want the Critical data sets reported:

   ```
   //   PARM='APPL=<APPL>,DATE=YYYYDDD'
   //*  PARM='APPL=<APPL>,DATE=YYYYDDD,#CRIT=YES'  OPTIONAL DSN
   ```

   For example, `// PARM='APPL=DAILY,DATE=2015047'`

   - The first card produces the job report only, based on the application you select (APPL=) and the date (DATE=YYYYDDD).
   - Optionally, you can request a listing of critical data sets.
   - During the execution, the CRITICAL file is checked. On the first use of the file during the cycle, the data set is listed with the job information at STEP end processing. Included with the data set information is the DDname, file type and job step.
3. If you want to report on all applications, use the following application selection:

APPL=.ALL*

By selecting ALL* for the application all applications processed on the selected day are reported.

Dormant report (DORMANT)

The CNTL(DORMANT) job generates a report of critical data sets that have not been referenced for a specified period of time.

About this task

If these DATA SETS are no longer in use, you can delete them from the Critical File List. After running IBM Z Batch Resiliency for an extended period of time, there may be files in the Critical File List that are no longer being used and, therefore, no longer need to be vaulted.

Note: Consult with people who have knowledge of the application, before removing any files listed on the DORMANT report.

Procedure

1. Before running the DORMANT job, ensure entries exist in the Critical File List.

   Note: GDG bases are listed on this report only if ALL of the generations have not been referenced since the specified date.
2. Use the DORMANT job to generate input cards. The input cards are used to remove files from the Inventory database (job: INVMAINT) and the critical file (job: CRITREMV). Review the lists before running the INVMAINT and CRITREMV jobs.
3. Specify the date to use in the evaluation criteria in the user exit.

   Depending on the date you specify, the listing may include some files, such as year-to-date accumulation files, that you may reference only once every quarter or yearly. For this reason, people with knowledge of the application should investigate any files listed on the DORMANT report before they are removed.

   For example, if you want to list any critical data sets that have not been referenced since March 1, 2017, enter the following date in USERMAC (XDORMANT):

   03/01/17

   The following sample report lists all files in the critical file list that have not been referenced since 03/01/17. To generate this report, enter 03/01/17 in the user exit.

```
                          IZBR - DISASTER RECOVERY/IBM Z BATCH RESILIENCY VERSION 1.1.0
                          PROPRIETARY SOFTWARE OF TERACLOUD S.A. AND IBM CORP.
                                                                                PAGE: 001
REPORT NAME: DORMANT
--------------------
LISTING OF FILES NOT REFERENCED SINCE: 03/01/17                                  DATE: 04/10/18 TIME:  8:59:36
APPLICATION: BOA
                                     FILE  DATE             LAST DATE        DATE             LAST DATE
DATA SET NAME                        TYPE  ENTERED          REFERENCED       CREATED          UPDATED
------------------------------------ ----  ---------------- ---------------- ---------------- ----------------
AG.PROD0.AUDITDB                           16NOV16:12:06:01 14NOV16:16:02:27                . 14NOV16:16:02:27
AG.PROD0.AUDTDS                            16NOV16:12:06:01 14NOV16:16:03:08                . 14NOV16:16:03:08
AG.PROD0.BKUPAUD                      C    16NOV16:12:06:01 05DEC16:20:03:41                .                .
AG.PROD0.BKUPJCL                      P    16NOV16:12:06:01 29OCT16:06:32:58                . 29OCT16:06:32:58
AG.PROD0.BKUPJCL.ABARS                P    16NOV16:12:06:01 29OCT16:06:43:38 29OCT16:06:43:38 29OCT16:06:43:38
AG.PROD0.BKUPMETH                          16NOV16:12:06:01 24NOV16:17:21:14                .                .
AG.PROD0.BKUPTBL                      C    16NOV16:12:06:02 19DEC16:14:23:13                . 19DEC16:10:50:34
AG.PROD0.SBRNSRCE                     P    16NOV16:12:06:02 17DEC16:14:43:37                .                .
AG.PROD0.SCKKLOAD                     P    22FEB16:17:59:01 18DEC16:13:26:33                .                .
AG.PROD0.SUBFILE.G%%%%V%%             G    16NOV16:12:06:02 29OCT16:06:46:41                . 29OCT16:06:46:41
AG.PROD0.SUBFILE4                          22FEB16:17:59:01 14DEC16:16:58:14                . 04DEC16:09:24:44
AG.PROD0.USERMAC                      P    16NOV16:12:06:02 19DEC16:14:23:19                . 19NOV16:17:00:36
AG.PROD411.A141109.V410.INVENTRY      C    22FEB16:17:59:01 19NOV16:17:00:32                .                .
AG.DEV430A.CNTL                       P    22FEB16:17:59:01 25NOV16:15:05:47                .                .
AG.TEST.FIXED.TAPE                         22FEB16:17:59:02 05DEC16:20:04:12                .                .
AG.TEST.FIXED.TAPE2                        22FEB16:17:59:02 05DEC16:20:02:09                .                .
```

Figure 130. DORMANT sample report

Chapter 12. Restoring Critical Data Sets

The RESTJCL utility generates the JCL that restores all critical data sets marked in the BKUPTBL as successfully backed up.

IBM Z Batch Resiliency supports the concept of defining and tracking multiple restoration methodologies. In many cases, the need to use multiple backup utilities are required for backing up mainframe data. For example, a mainframe shop might back up and restore critical sequential data with DF/DSS but also use FAVER to back up VSAM data.

You can restore individual data sets for operational recovery using ISPF panels or run RESTJCL to restore all application data sets.

- Restoring specific data sets using ISPF panels
- Restoring all critical data sets using RESTJCL

Restoring specific data sets using ISPF panels

To restore specific data sets for operational recovery, use the ISPF panels.

Procedure

1. Start the ISPF environment using TSO EX HLQ.CNTL(BRN110).
2. From the IBM Z Batch Resiliency Primary Panel, select option 6, INQUIRY.
3. Select option 2, Backup Table Inquiry.
4. Enter your selection criteria in the Backup Table Filter panel.
   For example, to list all data sets that start with PROD.ACCT.xxxx, enter:

```
------------------------ IZBR - Backup Table Filter ---------------------------------------
COMMAND ===>                                                    Scroll ===> CSR
Query limit: 5000   records
                                                              Rows 1 to 21 of 25

Column                    Comp Compare Value
Name                      Oper 1---------10---------20---------30---------40---------50-
Data set name . . . . . . =___ PROD.ACCT.xxxx ____________________________________________
Application id. . . . . . ____ ____________________________________________________________
Sub application id. . . . ____ ____________________________________________________________
Backup method . . . . . . ____ ____________________________________________________________
Backup date . . . . . . . ____ ____________________________________________________________
Backup time . . . . . . . ____ ____________________________________________________________
Backup date-time. . . . . ____ ____________________________________________________________
Backup flags. . . . . . . ____ ____________________________________________________________
User info . . . . . . . . ____ ____________________________________________________________
Change date . . . . . . . ____ ____________________________________________________________
Change time . . . . . . . ____ ____________________________________________________________
Change date-time. . . . . ____ ____________________________________________________________
Set date. . . . . . . . . ____ ____________________________________________________________
Set time. . . . . . . . . ____ ____________________________________________________________
Set date-time . . . . . . ____ ____________________________________________________________
Backup job. . . . . . . . ____ ____________________________________________________________
Backup job #. . . . . . . ____ ____________________________________________________________
Size (MB) . . . . . . . . ____ ____________________________________________________________
Seqment count . . . . . . ____ ____________________________________________________________
```

Figure 131. Backup Table Filter panel

Note: Press PF1 for a description of compare operators.

The Backup Table Display lists all of the data sets that match your selection criteria.

5. From the Backup Table Display, select the data set you want to recover.
   - To recover an individual data set, enter R next to the data set name.

© Copyright IBM Corp. 2019, 2020

- To recover multiple data sets, type S next to each data set name, then press the Enter key.

```
--------------------------------IZBR - Backup Table Display ---------------------------------
 COMMAND ===>                                                          Scroll ===> CSR
 Data set name filter: [begin] SJMK
 Cols 1 to 14 of 24
     S                              Appl SubAppl Backup  Backup     Backup  Backup User Change    Chan
 CMD R Data set name                Id   Id      Method  Date       Time    Flags  Info Date      Time
 ___   PROD.ACCT.BAUDFLAT.G0001V00  LMNOP        DSS     2018/07/28 14:52:46            2018/06/28
 ___   PROD.ACCT.BAUDFLAT.G0002V00  LMNOP        DSS     2018/09/08 14:52:46            2018/06/28
 ___   PROD.ACCT.BAUDFLAT.G0003V00  LMNOP        DSS     2018/09/08 14:52:46            2018/06/28
 ___   PROD.ACCT.BAUDFLAT.G0004V00  LMNOP        DSS     2018/09/08 14:52:46            2018/06/28
 ___   PROD.ACCT.BAUDFLAT.G0005V00  LMNOP        DSS     2018/09/08 14:52:46            2018/06/28
 ___   PROD.ACCT.BKUPJCL. DSS       LMNOP        DSS     2018/06/28 14:52:12            2018/06/28
 ___   PROD.ACCT.RESTJCL. DSS       LMNOP        DSS     2018/06/28 14:52:12            2018/06/28
 ___   PROD.ACCT.RESTJCL. DSS2      LMNOP        DSS     2018/06/28 14:52:12            2018/06/28
 ___   PROD.ACCT.SORTED.HIST.LONG   LMNOP        ALLOCAT 2018/07/28 14:52:46            2018/06/28
 _r_   PROD.ACCT.VBCLIST            LMNOP        DSS     2018/08/28 14:52:46            2018/06/28
 ___   PROD.ACCT.VSAM.TEST02        LMNOP        ALLOCAT 2018/09/28 14:52:12            2018/06/28
```

Figure 132. Backup Table Display

The IBM Z Batch Resiliency restore JCL generator displays the information for the selected data set including the backup method, job number, application ID and sub-application ID.

6. If you want to restore the file to a different data set name, enter an alternate name as shown in this example:

```
                IZBR restore JCL generator
Data set name: PROD.ACCT.VBCLIST
Backup method: DSS     Job number: JOB03995
Application Id: LMNOP   SubAppl Id:
JCL data set IZBR.V1R1M0.RESTJCL.DSS3
Replace existing JCL Y
Create non-existing JCL library Y
Restore to data set _PROD.ACCT.VBCLIST2___________________
Replace existing data set Y
Press ENTER to save the changes
Press PF3   to terminate the dialog
```

Figure 133. IBM Z Batch Resiliency Restore JCL Generator

The restore JCL is displayed as shown in the following example:

```
VIEW        PROD.ACCT.RESTJCL.DSS3
Command ===> ____________________________________________________________
//*
//PS10.BRNVFILT DD *
 INCLUDE(PROD.ACCT.VBCLIST2)
/*
//PS10.SYSIN DD *
 RESTORE INDD(TAPE) TGTGDS(ACTIVE) SPHERE -
   TOL(ENQF) CAT DATA SET(FILTERDD(BRNVFILT)-
   REPLACE
/*
```

Figure 134. RESTJCL example

7. To restore the data set, do one of the following:
   - Issue the Submit command to run the JCL
   - Save the generated JCL and schedule the job to run at a later time

Restoring all critical data sets using RESTJCL

The RESTJCL utility generates the JCL that restores all critical data sets marked in the BKUPTBL as successfully backed up.

You can use this method of recovery as a disaster recovery exercise to restore all production application data sets.

Recovery job flow process

This diagram shows the critical file recovery process using RESTJCL.

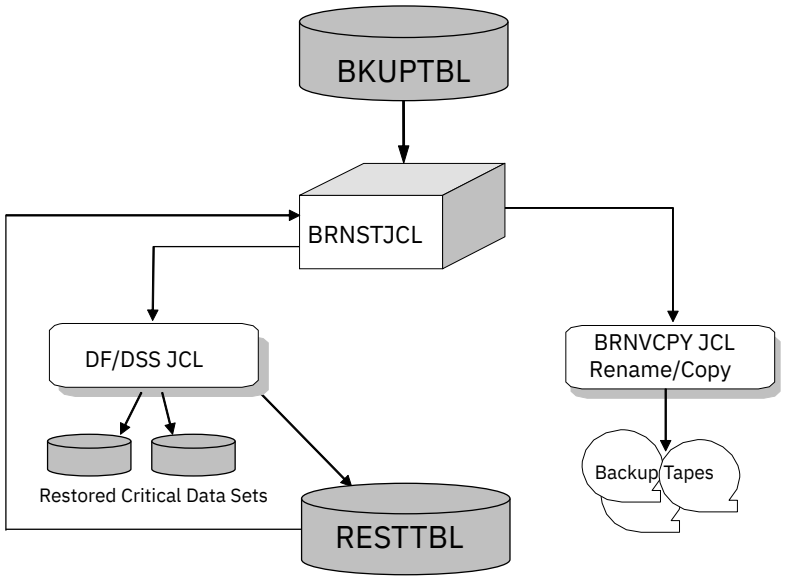


Figure 135. Recovery job flow process diagram

Generating the RESTJCL

The RESTJCL utility generates the JCL to restore all application critical files that have been successfully backed up, as marked in the BKUPTBL.

The files generated by RESTJCL are grouped by backup methodology, which is incorporated into the generated JCL. For example:

- DF/DSS restores are directed to the library *hlq*.RESTJCL.DSS
- FAVER restores are directed to *hlq*.RESTJCL.FAVER

All members generated in these libraries correspond to the APPLID that they were backed up with.

Tape restores are handled in a different manner, and actual I/O to restore the tapes is not required. To make a tape available, you can rename the tape data set name in the Tape Management Catalog and change the z/OS catalog. See Restoring data sets from BRNVCPY backups for more information.

Restoring specific applications or data sets using the XRESTJCL Exit

Use the XRESTJCL exit to select specific applications or data sets to be restored.

The USERMAC(XRESTJCL) member exit point is invoked after the critical data sets are evaluated, and prior to the generation of JCL to restore the files. Some sample exit routines follow.

In this example, no data sets backed up before Julian date *yyyyjjj* for DFDSS are recovered.

```
IF BKPMETH = 'DSS' AND BKPDATE LE 'yyyyjjj' THEN DROP = 1
```

In this example, remove any APPLIDs not required to be run as shown below:

```
IF APPLID = 'TEST' THEN DROP = 1
```

If there are any data sets marked for recovery in REPORTS(RESTJCL) that you do not want to recover (for example, redundant recoveries or database backups), add an exit code to remove specific data sets. For example, add an exit code such as the following, then rerun RESTJCL to generate the new JCL.

```
IF DSNAME = 'ADABAS.FILE' THEN DROP = 1
```

Deleting critical files ($DELETE$)

In each restore PDS, a member named $DELETE$ is generated that contains an IDCAMS parameter list used to delete all the critical files about to be restored. In many cases, you can run this job so that device allocation does not occur when the restores are run. If you determine the need to run this job, be aware that the DELETE keyword has been generated to FIREME and must be changed to properly execute.

CAUTION: Do not run $DELETE$ in a production environment.

This job is not designed to run in a production environment; running it there deletes critical data.

There is one option available for $DELETE$ in USERMAC member YDELETE.

Table 58. MODE option parameters

Option	Parameter	Description
MODE	SMART	IBM Z Batch Resiliency uses DCOLLECT volume information and uses SCRATCH for online volumes.
	DUMB	IBM Z Batch Resiliency generates all cards with NOSCRATCH options.

Uncataloging data sets (YDELETE)

In the $DELETE$ job, IBM Z Batch Resiliency invokes the JCL YDELETE procedure to uncatalog data sets for each methodology. The YDELETE procedure is used to simplify the process of evaluating the output of the IDCAMS output generated when the critical data sets are uncataloged or deleted.

In addition, when the USERMAC(YDELETE) is set for MODE=SMART, IDCAMS is re-invoked for data sets that failed to be deleted due to the volume being online and the data set not being present, (message IKJ56232I). All IDCAMS messages are evaluated and directed to one of four output files to aid in resolving any issues. The following are the IDCAMS report DDs:

GOOD
Listing of successful deletes.

NOTFOUND
Data sets that were not cataloged

NOTFOUND
Data sets that were not cataloged

NOTONVOL
Data sets not uncataloged with the SCRATCH option because the physical file was not present on the on-line volume.

OTHER
Summary of files not processed for other reasons.

IBM Z Batch Resiliency creates a second list of data sets to be processed with the NOSCRATCH option for any data set in the NOTONVOL list. These are data sets that had previously been attempted to be deleted with the SCRATCH option.

Building a data set previously restored using RESTTBL

IBM Z Batch Resiliency tracks applications that have been restored in the Restore Table (RESTTBL). You can use RESTTBL to build a JCL for a data set previously restored, by removing the appropriate record from the RESTTBL file.

When IBM Z Batch Resiliency recovers data sets, entries are written to RESTTBL. If you run RESTJCL again, the data sets that have already been restored are excluded from the JCL generation.

Important: If you are running RESTJCL for the first time at a DR site, ensure that there are no entries in the Restore Table. Otherwise, JCL are not generated for those files.

Creating jobs to allocate data sets (RESTALC)

When using backup utilities that do not include allocation data for data sets that are backed up (such as SORT, IDCAMS, and IEBCOPY), run RESTALC before the restore jobs are created by RESTJCL. RESTALC creates jobs (in *hlq*.RESTJCL.ALLOCAT) that allocate data sets before they are restored.

Before running RESTALC:

- Verify the required backup method (BKUPMETH) files are allocated. See Allocating the backup method JCL libraries for more information.
- Run BKUPALC to create ALLOCAT records in the BKUPTBL. See Adding data to the Backup Table for non-standard utilities for more information.

Modifying the action of RESTALC (XRESTALC)

Use the member USERMAC(XRESTALC) to modify or limit the action of RESTALC.

For example, you can remove all application IDs that include ‘TEST’ as shown in this example:

```
IF APPLID = 'TEST' THEN DROP = 1
```

The following figure shows an example of the JCL used to run CNTL(RESTALC):

```
//jobnam   JOB ...
//*-------------------------------------------------------------------
//* EXTRACT ALLOCAT METHODOLOGY RECORDS
//*-------------------------------------------------------------------
//PS010    EXEC PGM=SORT,REGION=64M
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SORTIN   DD  DSN=HLQ.BKUPTBL,DISP=SHR
//SORTOUT  DD DSN=&&SORTED,
//         UNIT=SYSDA,DCB=(LRECL=1741,RECFM=FB,BLKSIZE=0,DSORG=PS),
//         SPACE=(CYL,(15,15),RLSE),DISP=(NEW,PASS)
//SORTWK01 DD  UNIT=&WUNIT,SPACE=(CYL,(50,25))
//SORTWK02 DD  UNIT=&WUNIT,SPACE=(CYL,(50,25))
//SORTWK03 DD  UNIT=&WUNIT,SPACE=(CYL,(50,25))
//SYSIN    DD  DSN=HLQ.USERMAC(SORTRES1),DISP=SHR
//*-------------------------------------------------------------------
//PS020    EXEC PGM=IKJEFT01,PARM='RESTALC'
//STEPLIB  DD  DISP=SHR,DSN=HLQ.LIBRARY
//SYSUDUMP DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//SYSEXEC  DD  DSN=HLQ.SBRNSRCE,DISP=SHR
//WORK     DD  DSN=&&SORTED,DISP=(OLD)
//LAYOUT   DD DSN=HLQ.USERMAC(#BKUPALC),DISP=SHR
//XRESTALC DD DSN=HLQ.USERMAC(XRESTALC),DISP=SHR
//BKJCL    DD DSN=&&RESTJCL,
//         UNIT=SYSDA,
//         DCB=(LRECL=1741,RECFM=FB,BLKSIZE=0,DSORG=PS),
//         SPACE=(CYL,(15,15),RLSE),DISP=(NEW,PASS)
//*-------------------------------------------------------------

//PS030    EXEC PGM=MAKEJCL,PARM='HLQ.USERMAC'
//STEPLIB  DD DISP=SHR,DSN=HLQ.LIBRARY
//SYSEXEC  DD DISP=SHR,DSN=HLQ.SBRNSRCE
//TEMPLATE DD DISP=SHR,DSN=HLQ.USERMAC
//BKJCL    DD DISP=SHR,DSN=&&RESTJCL
//UFIELDS  DD DISP=SHR,DSN=HLQ.USERMAC(#MAKEALC)
//BKUPMETH DD DISP=SHR,DSN=HLQ.BKUPMETH
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//MAKEJCL  DD DUMMY
//XMAKEJCL DD DUMMY
```

Figure 136. JCL to run RESTALC

Backup and restore methodologies

This section describes the backup and restore methodologies supported by IBM Z Batch Resiliency. In cases where generated JCL invokes procedures, those procedure names are identified within the table for each methodology.

Restoring DASD data sets backed up by DF/DSS

Use DF/DSS to restore DASD data sets backed up with DF/DSS. The generated JCL executes a procedure that resides in the CNTL library. The members of the RESTJCL.DSS library correspond to the APPLIDs defined to this methodology.

The member $DELETE$ is generated to optionally delete all of the critical files associated with this methodology.

Note: Run $DELETE$ only in a non-production or recovery environment. See Deleting critical files ($DELETE$) for details.

The name of the output data set is generated by IBM Z Batch Resiliency and the Restore Table is automatically updated when the process completes.

```
DF/DSS Information
NAME IN BKUPMETH
DSS
Procedure in the CNTL Library (RESTORE)    YDSSREST
```

```
DD Statement in RESTJCL    dynamic allocation
RESTJCL Template in USERMAC    BDSS

Restore Updates RESTTBL
YES
```

The following is a sample of the JCL RESTJCL generated for DF/DSS:

```
//jobnam   JOB ...
//*-------------------------------------------------------------------
//* IZBR RESTORE JOB
//* (JOBNAM ABOVE MUST BE 6 CHARACTERS OR LESS)
//* DO NOT EXCEED 20 LINES IN THIS MEMBER
//*-------------------------------------------------------------------
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL.)
//*-------------------------------------------------------------------
//*
//*-------------------------------------------------------------------
//*RESTORE SEPARATOR FOR: IZBR.BKUPV.ALLDASD1.G0001V00
//*               BKPVOL: 620013
//*-------------------------------------------------------------------
//JS01    EXEC YDSSREST,REGION=4M,
//             APPLID=ALLDASD1
//TAPE     DD  DSN=IZBR.BKUPV.ALLDASD1.G0001V00,
//**           UNIT=3590-1,LABEL=(1,SL),
//             VOL=(,RETAIN,,50),
//**           SER=(620013)),
//             DISP=OLD
//SYSIN DD *
 RESTORE INDD(TAPE) TGTGDS(ACTIVE) -
   SPHERE REPLACE TOL(ENQF) CAT -
    DATA SET(FILTERDD(BRNVFILT))
//PS10.BRNVFILT DD *,LRECL=80
            INCLUDE(APPS1.PSOFTFAX.SPOOL.G3405V00 -
                    AP001.APCL01DD.DATA.G2390V00 -
                   )
```

Figure 137. RESTJCL for DF/DSS

Restoring online DASD data sets backed up by ABARS

You can use DFSMShsm ABARS to restore online DASD, migrated DASD, and tape data sets that have been backed up with ABARS. ABARS is a started task that is initiated by HSM.

The members of the RESTJCL.ABARS correspond to the APPLIDs designated for the aggregates. The member $DELETE$ is generated by RESTJCL to optionally delete all the critical files associated with this backup methodology.

Note: Run $DELETE$ only in a non-production or recovery environment. See Deleting critical files ($DELETE$) for details.

Table 59. RESTJCL.ABARS members	
ABARS information	
NAME IN BKUPMETH	ABARS
Procedure in the CNTL Library (RESTORE)	YABARS
DD Statement in RESTJCL	Dynamic allocation
RESTJCL Template in USERMAC	BABARS
Restore Updates RESTTBL	Using ABARBKUP

```
//YABARS PROC
//*
//* RUN BATCH PGM TO SEND A COMMAND TO HSM AND SET UP GLOBAL TABLES
//*  FOR ARCSKEXT
//*
//BRNTBARS EXEC PGM=BRNTBARS,
//     PARM='MSGLVL(4),SVCDUMP(N),TRACE(N),TYPERUN(RUN)'
//*
//SYSPRINT DD SYSOUT=*
//INCLUDE  DD DUMMY
//CMDIN    DD DUMMY
```

Figure 138. YBARS Procedure

The YABARS procedure shown in the figure is invoked when JCL is generated with the BRNDBARS option enabled in USERMAC(YABARS), for example, BRNDBARS=YES.

BRNTBARS parameters

The following table outlines the RESTBARS parameters

Table 60. RESTBARS parameters

Parameter	Value	Description
MSGLVL	0 through 9 (Default: 3)	Controls the number of messages to DD:SYSPRINT for both the batch requester and the ABARS exit. Zero is the minimal level of messages that are issued.
SVCDUMP	YES or NO	If set to YES, an SVC DUMP is issued if Batch Requestor abends. Note that if ABARS exit abends, the SVC dump is always issued.
TRACE	Yes or NO	Use only if instructed by Technical Support.
TYPERUN	RUN or NORUN	If NORUN is specified, the ARECOVER command is NOT sent to DFSMShsm and the return code is set to 888. Otherwise the ARECOVER command is sent to DFSMShsm.

Enabling the BRNTBARS and ARSKEXT exits

As distributed by IBM®, the DFSMShsm ABARS process does not allow selective restores from an ABARS aggregate. The IBM Z Batch Resiliency utility BRNTBARS is invoked by YABARS when BRNDBARS=YES is specified in USERMAC(YABARS). It is intended to simplify selective aggregate recovery by providing an ABARS skip exit to selectively recovery data. Unlike the ABARS restart data set, aggregate recovery is controlled by inclusion rather than exclusion.

There are two modules in the process, BRNTBARS and the ABARS exit itself, known as ARSKEXT (AbaRs SKip EXiT).

Two control files must be provided as input to BRNTBARS:

- File INCLUDE contains INCLUDE commands to specify data set names to be restored from the aggregate. If an * is in the first column the line is ignored as a comment. Data set name masking is not supported; the exact name of a data set must be provided. For example: INCLUDE TEST.APPL.JCL
- File CMDIN contains the ABARS ARECOVER command to be sent to DFSMShsm.

The ARECOVER command provided in the CMDIN input file is sent unchanged to DFSMShsm and the Batch Requestor waits for the command to complete. When the command completes, the return code from the ARECOVER command is used as the step return code. A report is generated containing the names of all of the restored data sets. The names of data sets specified in the INCLUDE input file but not found within the aggregate are written to another report. The name of the log data set generated by the ABARS exit is reported in the SYSPRINT file of the BRNTBARS job.

BRNTBARS takes advantage of the HSM exit ARCSKEXT using it to filter out data sets that are not to be restored from the aggregate. To enable the exit, the command `SETSYS EXITON(SK)` should be entered

into the HSM startup command set, ARCCMDxx contained in the active system parameter library (for example, SYS1.PARMLIB).

Each data set to be restored is validated against the contents of the aggregate. If it is found within the aggregate, the data set is approved for restoration. It is possible that an approved data set is not restored by ABARS since the restore process may fail for reasons external to IBM Z Batch Resiliency. However, the reason for any failure is reported in the ABARS log, the name of which is sent back to the associated Batch Requestor process.

BRNTBARS requirements

This section describes the requirements for BRNTBARS.

- BRNTBARS and ARSKEXT require APF authorization.
- Issue the command SETSYS EXITON(SK) to DFSMShsm for the exit to be called and only desired files to be restored. To ensure that the exit is always enabled, include the command in the ARCCMDxx startup HSM command set.
- The Batch Requestor ensures that the exit is enabled and reports the status of all exits. If the exit is not enabled, the Batch Requestor terminates with an appropriate message and a non-zero return code (4 if the exit is off, 8 if it has never been activated).
- Ensure the user or job that issues the ARECOVER command has DFSMShsm administrator privileges.
- In the ABARS procedure, add a STEPLIB that points to the IZBR.LIBRARY where the exit resides. The ABARS procedure that needs to be modified is specified in ABARSPROCNAME(xxxxxxxx) in member ARCCMDxx in the //HSMPARM data set specified in the DFSMShsm started task. Alternatively, the exit module can be placed in a library in the LINKLIST concatenation.

Restoring DASD data sets backed up by FDR/ABR

Use the FDR/ABR (Automatic Backup and Restore) component to restore DASD data sets that have been backed up with Fast Dump Restore (FDR).

The generated restore JCL executes a procedure that resides in the CNTL library. The members of RESTJCL.FDR library correspond to the APPLIDs defined to this backup methodology. The member $DELETE$ is generated by RESTJCL to optionally delete all the critical files associated with this backup methodology prior to their restoration.

Note: Run $DELETE$ only in a non-production or recovery environment. See Deleting critical files ($DELETE$) for details.

The Restore Table is automatically updated upon completion of the process.

Table 61. Information for FDR/ABR	
FDR/ABR information	
NAME IN BKUPMETH	FDR
Procedure in the CNTL Library (RESTORE)	YFDRREST
DD Statement in RESTJCL	Dynamic allocation
RESTJCL Template in USERMAC	BFDR
Restore Updates RESTTBL	No

The following is a sample of the JCL RESTJCL generated for FDR/ABR:

```
//jobnam JOB ...
//*
//*RESTORE SEPARATOR FOR: HLQ.GL.D040506.T101024.T0010
//*              ARCDSN : ARCDSN
//*              BKPVOL : VT0046
//*
//JS10    EXEC YFDRREST,REGION=4M,
//             APPLID=GL,
//             ADSN=ARCDSN,
//*            BDSN=HLQ.GL.D040506.T101024.T0010
//PS10.TAPE1 DD DSN= HLQ.GL.D040506.T101024.T0010
RESTORE TYPE=ARC,DYNTAPE,RECAT,ONLVOL
ALLOCATE ALLDSN
SELECT DSNAME=HLQ.GL.DATA,COPY=1
```

Figure 139. JCL RESTJCL for FDR/ABR

The following procedure is in CNTL(YFDRREST):

```
//YFDRREST PROC APPLID=,
//            RGN=4M,
//            WUNIT=SYSDA,
//            SOUT='*'
//*
//*====================================================================
//* THIS STEP WILL RESTORE ALL OF THE SPECIFIED DATA SETS TO
//* DASD VOLUMES.
//*====================================================================
//PS10     EXEC PGM=FDRDSF,REGION=&RGN
//SYSPRINT DD SYSOUT=&SOUT
//TAPE1    DD DUMMY
//SYSIN    DD DUMMY
//*====================================================================
//PS20     EXEC PGM=FDRABR,REGION=&RGN
//ARCHIVE  DD DISP=SHR,DSN=&ADSN
//SYSPRINA DD SYSOUT=&SOUT
//SYSPRINT DD DSN=&&SYSOUT,DISP=(NEW,PASS),
//            UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
//            DCB=(RECFM=VBA,DSORG=PS,LRECL=121,BLKSIZE=1214)
//ABRWORK     DD   UNIT=&WUNIT,SPACE=(CYL,(1,1))
//*====================================================================
//PS30     EXEC PGM=IKJEFT01,COND=EVEN,
//     PARM=(BRNVRFDR,'&APPLID')
//STEPLIB  DD DSN=AG.DEV430A.LIBRARY,DISP=SHR
//SYSTSIN  DD DUMMY
//DDIN     DD DSN=&&SYSOUT,DISP=(OLD,DELETE)
//DDOUT    DD DSN=&&BRNVTEMP,DISP=(NEW,DELETE),
//         UNIT=SYSDA,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=255,BLKSIZE=255,RECFM=F,DSORG=PS)
//RESTAUD  DD DSN=AG.DEV430A.RESTTBL,DISP=(SHR)
//REPORT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//ERRORS   DD SYSOUT=*
//*
//*====================================================================
//PS40     EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1   DD DSN=&&SYSOUT,DISP=(OLD,PASS)
//SYSUT2   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
```

Figure 140. YFDRREST Procedure

Restoring VSAM data sets backed up by FAVER

Use FAVER to restore VSAM data sets backed up with FAVER. The generated JCL executes a procedure that resides in the CNTL library.

The members of the RESTJCL.FAVER library correspond to the APPLIDs defined to this backup methodology. The member $DELETE$ (also in RESTJCL.FAVER) is generated by RESTJCL to optionally delete all the critical files associated with this backup methodology.

Note: Run $DELETE$ only in a non-production or recovery environment. See Deleting critical files ($DELETE$) for details.

Table 62. RESTJCL.FAVER members	
FAVER information	
NAME IN BKUPMETH	FAVER
Procedure in the CNTL Library (RESTORE)	YFAVER
DD Statement in RESTJCL	Dynamic allocation
RESTJCL Template in USERMAC	BFAVER
Restore Updates RESTTBL	Yes

The following is a sample of the JCL RESTJCL generated for FAVER:

```
//jobname JOB ...
//*
//* IZBR BACKUP JOB
//* (JOBNAME ABOVE MUST BE 6 CHARACTERS OR LESS)
//* SEE MEMBER BKUPJOB IN USERMAC
//*
//*RESTORE SEPARATOR FOR:DR.BKUP.IZBR.D020718.T1432.T001
//*              BKPVOL :112160
//PS10     EXEC PGM=GVRESTOR,REGION=4M
//SYSPRINT DD SYSOUT=*
//SNAPDD   DD SYSOUT=*
//TAPE0    DD DISP=SHR,DSN=DR.BKUP.IZBR.D020718.T1432.T001
//SYSIN DD *
 RESTORE
 CLUSTER
 CL=AG.IZBR.V1R1M0.BKUPTBL
```

Figure 141. RESTJCL generated for FAVER

Restoring DASD data sets backed up by CA Disk (DMS/OS)

Use the DMS/OS JCL to restore DASD data sets backed up with the CA Disk DMS proc.

The generated JCL executes a procedure that resides in the .CNTL library. The members of both BKUPJCL.DMS and RESTJCL.DMS correspond to the APPLIDs defined to this backup methodology.

The member $DELETE$ in RESTJCL.DMS is generated by RESTJCL to optionally delete all the critical files associated with this backup methodology.

Note: : Run $DELETE$ only in a non-production or recovery environment. See Deleting critical files ($DELETE$) for details.

Table 63. DMS information	
DMS information	
NAME IN BKUPMETH	DMS
Procedure in .CNTL Library (RESTORE)	YDMSREST
DD Statement in RESTJCL	Dynamic allocation
RESTJCL Template in USERMAC	BDMS
Restore Updates RESTTBL	No

Restoring data sets from BRNVCPY backups

When the IBM Z Batch Resiliency BRNVCPY utility is used to copy critical tapes to backup volumes, you can recover the backups from these tapes.

See Copying critical tapes to backup volumes for offsite storage for more information.

Restoring data sets for TMS (CA-1)

To restore critical data sets in a TMS environment, run two IDCAMS jobs to re-catalog the critical data sets and run two TMS utilities after recovering the TMS database.

Procedure

1. Run the CNTL(TMSUPVOL) member to rename the disaster recovery data sets on the backup tapes to their original data set names in the TMS master file.
2. Run the CNTL(TMSUPDSN) member to update the records for multi-file tape volumes in the TMS master file.
3. Run the CNTL(TMSUPC02) member to execute IDCAMS to uncatalog the original tape files.
4. Run the CNTL(TMSUPC01) member to execute IDCAMS to associate the critical tape files with their backup copies.

Results

The control cards required to perform these restores reside in the RESTJCL.BRNVCPY library that is generated by the job RESTJCL. The generated members are:

DEFINES
The control statements that are used to redefine the Critical Files to refer to the backup copies. Member TMSUPC01 in the CNTL library invokes this input stream.

DELETES
The control statements that are used to delete the original critical tapes. ***Do not execute these statements in a production environment as real critical files would be UNCATALOGED***. Member TMSUPC02 in the CNTL library invokes this input stream.

DSNBUPDT
The TMS control statements used to update the tape backup records for multi-file tape volumes to reflect the original data set name. Member TMSUPDSN in the CNTL library invokes this input stream.

VOLUPDT
The TMS control statements used to update the tape backup volume records to reflect the original data set name. Member TMSUPVOL in the CNTL library invokes this input stream.

COPY
The control statements used to recover tape data sets with data set names shorter than the standard 17-character length. Member YTAPECPY in the USERMAC library invokes this input stream.

Restoring data sets for CA Dynam/TLMS

To restore critical data sets in a CA Dynam/TLMS environment, run two IDCAMS jobs to re-catalog the critical data sets to refer to their backup copies and run one CA Dynam/TLMS utility after recovering the CA Dynam/TLMS database.

Procedure

1. Run the CNTL(TLMSVOLR) member to rename data sets from disaster recovery backup tapes to the original data set names in the CA Dynam/TLMS master file.
2. Run the CNTL(TMSUPC02) member to execute IDCAMS to uncatalog original tape files.

3. Run the CNTL(TMSUPC01) member to execute IDCAMS to redefine the critical files to the backup copies.

Results

The control cards required to perform these restores reside in the RESTJCL.BRNVCPY library that is generated by the job RESTJCL. The generated members are:

DEFINES
The control statements that are used to redefine the Critical Files to the backup copies. Member TMSUPC01 in the CNTL library invokes this input stream.

DELETES
The control statements that are used to delete the original critical tapes. ***These statements are not to be executed in a production environment as real critical files would be UNCATALOGED***. Member TMSUPC01 in the CNTL library invokes this input stream.

VOLUPDT
The CA Dynam/TLMS control statements used to update the tape backup volume records to reflect the original data set name. Member TLMSVOLR in the CNTL library invokes this input stream.

COPY
The control statements used to recover tape data sets with data set names shorter than the standard 17-character length. Member YTAPECPY in the USERMAC library invokes this input stream.

Restoring data sets for Control-M/Tape (CTT)

To restore critical data sets in the Control-M/Tape (previously known as Control-T) environment, run two IDCAMS jobs to recatalog the critical data sets to point to their backup copies and run one Control-M/Tape utility after recovering the Control-M/Tape database.

Procedure

1. Run the CNTL(CTTVOLR) member to rename data sets from disaster recovery backup tapes to original data sets names in Control-M/Tape master file.
2. Run the CNTL(TMSUPC02) member to execute IDCAMS to uncatalog the original tape files.
3. Run the CNTL(TMSUPC01) member to execute IDCAMS to associate the critical tape files with their backup copies.

Results

The control cards required to perform these restores reside in the RESTJCL.BRNVCPY library that is generated by the job RESTJCL. The generated members are:

DEFINES
The control statements that are used to redefine the critical files to the backup copies. Member TMSUPC01 in the CNTL library invokes this input stream.

DELETES
The control statements that are used to delete the original critical tapes. **These statements are not to be executed in a production environment as real critical files would be UNCATALOGED.** Member TMSUPC02 in the CNTL library invokes this input stream.

VOLUPDT
The CTT control statements used to update the tape backup volume records to reflect the original data set name. Member CTTVOLR in the CNTL library invokes this input stream.

COPY
The control statements used to recover tape data sets with data set names shorter than the standard 17-character length. Member YTAPECPY in the USERMAC library invokes this input stream.

Restoring data sets for RMM

To restore critical data sets in the RMM environment, run two IDCAMS jobs to re-catalog the critical data sets to refer to their backup copies and run three jobs after recovering the RMM database.

Procedure

1. Run the (RMMBKUP) member to empty the RMM journal so that all changes to the database works. Verify that all RMM data sets are correct in the JCL.
2. Run the CNTL(RMMVOLR) member to rename data sets from disaster recovery backup tapes to the original data set names in the RMM master file.
3. Run the (RMMVER) member to verify the integrity of the RMM master file.
4. Run the CNTL(TMSUPC02) job to execute IDCAMS to uncatalog the original tape files.
5. Run the CNTL(TMSUPC01) job to execute IDCAMS to associate the critical tape files with their backup copies.

Results

The control cards required to perform these restores reside in the RESTJCL.BRNVCPY library that is generated by job RESTJCL. The generated members are:

DEFINES
The control statements that are used to redefine the critical files to the backup copies. Member TMSUPC01 in the CNTL library invokes this input stream.

DELETES
The control statements that are used to delete the original critical tapes. **These statements are not to be executed in a production environment as real critical files would be UNCATALOGED**. Member TMSUPC02 in the CNTL library invokes this input stream.

VOLUPDT
The IBM Z Batch Resiliency RMM rename utility control statements used to update the tape backup volume records to reflect the original data set name. Member RMMVOLR in CNTL library invokes this input stream. This utility is invoked on the unloaded RMM database prior to recovery, and the output is used to restore the RMM database.

COPY
The control statements used to recover tape data sets with data set names shorter than the standard 17-character length. Member YTAPECPY in the USERMAC library invokes this input stream.

Restoring data sets for ASG-Zara

To restore critical data sets in the ASG-Zara™ environment, run two IDCAMS jobs to recatalog the critical data sets to point to their backup copies and run one ASG-Zara utility after recovering the ASG-Zara database.

Procedure

1. Run ZARAVOLR member from CNTL library to rename data sets from disaster recovery backup tapes to original data sets names in ASG-Zara master file.
2. Run TMSUPC02 from CNTL library to execute IDCAMS to uncatalog the original tape files.
3. Run TMSUPC01 from CNTL library to execute IDCAMS to associate the critical tape files with their backup copies.

Results

The control cards required to perform these restores reside in the RESTJCL.BRNVCPY library that is generated by the job RESTJCL. The generated members are:

DEFINES
The control statements that are used to redefine the critical files to the backup copies. Member TMSUPC01 in the CNTL library invokes this input stream.

DELETES
The control statements that are used to delete the original critical tapes. **These statements are not to be executed in a production environment as real critical files would be UNCATALOGED**. Member TMSUPC02 in the CNTL library invokes this input stream.

VOLUPDT
The ASG-Zara control statements used to update the tape backup volume records to reflect the original data set name. Member ASG-ZaraVOLR in the CNTL library invokes this input stream.

COPY
The control statements used to recover tape data sets with less than the standard 17-character length. Member YTAPECPY in the USERMAC library invokes this input stream.

Chapter 13. Using TimeLiner

TimeLiner is a feature of IBM Z Batch Resiliency that monitors production jobs across the SYSPLEX in real time and identifies data sets that were in use when a failure occurred. TimeLiner identifies potential corrupted data sets by date and time of the event, to help you determine the point-in-time recovery period and quickly return applications to full production.

TimeLiner provides an interface to the Dell EMC Data Protector for Z Systems zDP™ product SnapVX feature. SnapVX creates and manages multiple (up to 255) point-in-time snapshots of volumes. The *snapshot* is a pointer-based, point-in-time image of a single volume. A *snapset* is a single *snapshot* of each volume across a group of volumes. TimeLiner builds commands to link and restore from each *snapset*.

TimeLiner overview

The IBM Z Batch Resiliency TimeLiner feature collects data set information from selected batch jobs.

The type of data collected includes:

- Job name and step name
- Job and step start and end times
- Data set name, open and close times, and data disposition

Data is collected based on job and data set-related exits and SVCs, which are installed when TimeLiner is initiated. A job is monitored only if the job's name matches an entry, either a fully-qualified job name or a job name mask, in the application table (APPTABLE).

TimeLiner uses the following exits and SVC routines to collect job information and data:

- SMF exits – collects job and step initiation and termination events.
- SVC routines – collects data set information by these frontends: SVC 19 (data set open), 20 (data set close) and 22 (data set open type J).
- Job Initiation Exit (IEFUJI) – collects the start time and job name of the targeted job. The exit checks the intercepted job name and if a match with the mask or job name is found in the application table, then TimeLiner monitors this job.

Data is collected from several points and written to a Data-In-Virtual (DIV) data space, which is backed by a linear VSAM file. In addition, the data is also written to a log stream file which provides a mechanism to access historical data.

Using Log Stream information

When you install TimeLiner, you define a log stream which is used by TimeLiner. For more information on installing TimeLiner, see TimeLiner component setup. The staging data set, IXGLOGR.SYSPLEX.TIMELINR, is allocated while TimeLiner is running and is deleted at termination.

About the zDP Restore interface

The TimeLiner zDP interface provides information about Dell EMC SnapVX snapset technology through the use of IBM Z Batch Resiliency SMF data collection and analysis. The real-time SMF collection subtask for zDP captures required metadata, at the data set level, of the zDP processes. TimeLiner captures the snapset metadata based on the SMF record type generated by the zDP VDG definitions, which is added to *hlq*.CNTL(SMFP&*systemid*) member SMFP&SYSTEM.

TimeLiner uses knowledge of jobs that are running and what data sets are open, combined with IBM Z Batch Resiliency application metadata, to enable you to identify data at risk from an event and quickly recover from the most appropriate backup.

© Copyright IBM Corp. 2019, 2020

When a data set is selected to be restored, TimeLiner builds the control cards and recovers the data set with a new name and cleans up the LINK so that it can be done again with a different data set. Only one snapset VDG Group can be linked to a target (TGT) group.

Starting TimeLiner

Start TimeLiner as a started task to ensure continuous monitoring of your application jobs.

Before you begin

Before starting TimeLiner, confirm that the TimeLiner initial implementation tasks (found in IBM Z Batch Resiliency implementation) and configuration tasks (found in Configuration) are complete:

- Load libraries are AFP authorized:
  - DSN=*hlq*.SCKKLOAD
  - DSN=*hlq*.SBRNLOAD
  - DSN=*hlq*.SBRNTLOD
- Data sets required by TimeLiner are allocated:
  - DIV linear data set – *HLQ*.CNTL(TLNDEFDV)
  - VSAM UFIFILE – *hlq*.CNTL(UFIDEF)
  - ZDPAUDIT – *hlq*.CNTL(DEFZDPAU)
  - Log stream – *hlq*.CNTL(TLNDEFLG)
- APPTABLE in USERMAC is built and contains the jobs TimeLiner monitors.
- SMF record type and real-time SMF is configured for zDP.

Procedure

1. To start Time Liner as a Started Task, issue the following command:

   S <TLINERP>

   where <TLINERP> is the name of the TimeLiner started task procedure.

   The default started task is *hlq*.CNTL(**TLINERP**) which you have copied to a proclib as part of the installation procedure.

```
//TLINERP   EXEC PGM=BRNLCT00,PARM='MEMBER(TLNPRM00)'
//*
//*  TIMELINR STARTED TASK JCL
//*
//STEPLIB  DD  DISP=SHR,DSN=HLQ.SBRNTLOD
//         DD  DISP=SHR,DSN=HLQ.SCKKLOAD
//         DD  DISP=SHR,DSN=HLQ.SBRNLOAD
//RTDPRINT DD  SYSOUT=*
//APPTABLE DD  DISP=SHR,DSN=%USERMAC%(APPTABLE)
//PARMLIB  DD  DISP=SHR,DSN=%USERMAC%
```

Figure 142. TimeLiner Started Task

Table 64. TimeLiner start parameters

Program	Description
EXEC	PARM keyword specifies the PARMLIB member as MEMBER(**TLNPRM00**), which contains the startup parameters
STEPLIB	Defines the TimeLiner libraries
RTDPRINT	SYSOUT=*

Table 64. TimeLiner start parameters (continued)	
Program	**Description**
APPTABLE	The application table that defines the jobs to be managed
PARMLIB	Points to the PARMLIB library

2. To start TimeLiner as a batch job, submit HLQ.CNTL(TLINER).
3. To verify the address space is fully initialized, check the JESMSGLG log file of the TimeLiner job for the message:

   ```
   BRNL266I  TIMELINER INITIALIZATION COMPLETE
   ```

   The first time you activate the started task, you might see the following message in the BRNLCT00 data set:

   ```
   BRNL013I BRNLCT00  CALL TO 'CSRPQPL' NEAR LABEL 'MAIN1351' COMPLETED
       WITH COMPLETION CODE 28
   ```

   This message indicates the Data-In-Virtual (DIV) data space is not yet initialized. The started task needs to initialize the DIV before it is used in the ISPF Dialog.

   To check for other messages, look at the BRNLCT00 member in the TimeLiner job for more information.

Using the TimeLiner Job Record Viewer

The interface allows the user to navigate between jobs, steps and data sets, tracking either forwards or backwards in time.

About this task

Data sets in the interface links to other job steps that ran at other times. Below each input data set, the previous steps that wrote to the data set is shown. Below each output data set, the subsequent steps that read from the data set is shown. Each component type shows different navigation options.

Job Panel

The job panel shows the steps of the job in execution order. Steps are shown as a tree table in the following figure.

```
--------------------------------TimeLiner Job Record Viewer--------------------------------
System ID: BRA1
Job Name: MICK0001    Job Active  : No
Job Id  : JOB21745    Job Start   : 2020/09/09 14:05:12.52
Job CC  : 0           Job End     : 2020/09/09 14:05:12.67
Max CC  : 0

Cols 1 to 7 of 7
S  +  Type  Name      Active   Start/Open              End/Close               Details
L  -                  State    Date/Time               DateTime
      Step  DEL (1)   No       2020/09/09 14:05:12.52  2020/09/09 14:05:12.57  CC=0
    + Step  COPY (2)  No       2020/09/09 14:05:12.57  2020/09/09 14:05:12.63  CC=0
    + Step  STEP9 (3) No       2020/09/09 14:05:12.63  2020/09/09 14:05:12.67  CC=0
****************************************Bottom of data****************************************
```

Figure 143. Job panel showing steps of job in tree table

The + and - symbols indicate expandable and collapsible nodes. Enter line command + to expand the corresponding row, or line command - to collapse the corresponding row.

Expanding a step shows data sets used by the step. Expanding a data set shows linking steps from other jobs that accessed the data set. Linking steps have the format:

```
JOBNAME(JOBNUMBER)-STEPNAME(STEPNUMBER)
```

Entering line command **S** navigates to the selected component.

Step panel

The step panel shows the input and output data sets in open order. Data sets are shown in a tree table - as shown in the preceding figure. Entering line command + expands the selected row, using - collapses the row.

Expanding a data set shows linking steps from other jobs that accessed the data set. Entering line command **S** navigates to the selected component.

Data set panel

The data set panel shows linking steps for the data set across the total hours back time. The links are across hours back time, rather than in relation to when the data set was used. Entering line command **S** navigates to the selected component.

Hours back

The hours back filter option limits the amount of Job records read from the TimeLiner database.

Note: If the hours back filter is set to a smaller value, the interface runs faster.

Recovery

Entering an **R** line command against any data set in the interface shows a list of available backups for that data set. This list is generated from the backup database.

The Valid column indicates that the backup is valid for the particular job step; which means it hasn't been modified before the job step was run.

The backup highlighted in white indicates the backup is closest to when the job ran.

Recovering zDP snapsets

Using TimeLiner, you can determine the point-in-time zDP data set snapshot at the job level, which enables you to recover stable data then restart the batch job to return the application to production.

Before you begin

- ISPF for TimeLiner is configured
- TimeLiner is running as a started task
- Real-time SMF is running
- Dell EMC zDP is installed in your environment

Before you restore any zDP snapset, you need the following information:

- Date and time of when the corruption occurred
- Knowledge of the application flow and the input files for a particular job

About this task

Note: Refer to Dell EMC product documentation for information about TimeFinder® SnapVX and zDP™.

To restore selected snapsets, use the IBM Z Batch Resiliency TimeLiner zDP interface option. To see available TimeLiner commands for a panel, enter a slash (/).

To recover a zDP snapset:

Procedure

1. From the IBM Z Batch Resiliency primary panel, select option 8, TLN - TimeLiner Functions.
2. Select option 1, TimeLiner - Job and Dataset Timeline Viewer.
3. Select option 2, zDP – EMC zDP interface to view the contents of the ZDPAUDIT file.

4. Locate the snapset name for the date and time of the event, as shown in this example:

```
COMMAND ===>

Cols 1 to 5 of 16
S Status  Snapset
L Message Name
          C21VDG02. . . . . 183471954C00338… 2018/12/13  19:54:
          C21VDG02. . . . . 183471944C00337… 2018/12/13  19:44:
          C21VDG02. . . . . 183471934C00336… 2018/12/13  19:34:
          C21VDG02. . . . . 183471924C00335… 2018/12/13  19:24:
          C21VDG02. . . . . 183471914C00333… 2018/12/13  19:04:
          C21VDG02. . . . . 183471914C00332… 2018/12/13  18:54:
          C21VDG02. . . . . 183471914C00331… 2018/12/13  18:44:
          TSOVDG02. . . . . 183471843C00090… 2018/12/13  18:43:
```

Figure 144. TimeLiner Snapsets

Note: In the above example, the snapset name C21VDG02 identifies application data sets and TSOVDG02 identifies backups of SO data sets. The snapset names might be different for your site.

5. Enter the data set name and optional high-level qualifier (HILEVEL):

```
_________________________ TimeLiner zDP Custom Recovery _____________________
COMMAND ===>

Snapset       C21VDG02……183031256C01753..
Data Set  =>  PROD.TESTING.DDA.V5.FRD.V51
Target Group         => C21TGT02 (required)
Source Volume List      BRNVGRP0s

OPTIONAL FIELDS:
Enter HILEVEL    => newhlq
<or>  LLQ    =>

             Specify the value and press ENTER  to continue
                                    or CANCEL to exit
```

Figure 145. TimeLiner zDP Custom Recovery

6. After the data set is recovered and you confirm it is valid, restart the job.
7. To determine the downstream jobs, use the Cascade Impact Report as described in the next section.

Generate a cascade report from a selected job, step or data set

The interface allows the generation of a forward and reverse cascade report from any job, step, or data set.

About this task

The report shows how previous or subsequent Jobs and Steps have affected the selected component. This is similar to previous versions of the forward cascade report which showed subsequent jobs, steps, and data sets affected by the selected component.

The format of the report shows jobs, steps and data sets in a structured way, to help the user understand how they link together.

Procedure

1. Next to any job, step, or data set in the interface, type **P** for reverse cascade or **N** for forward cascade.

Note: In the interface, the functionality of using **N** or **P** to navigate reverse cascade or forward cascade is not enabled when the 'Show only DSNs requiring recovery option' is set to YES.

2. The report is a tree table, and data set rows can be expanded to show linking steps.
3. The selected cascade report is displayed. The selected component is highlighted in white.

4. Linking data sets are shown in red. Output DS indicates a later job or step read from this data set. Input DS indicates a previous job or step written to this data set.
5. The report shows components in order of start and open times. Forward cascade is displayed in *ascending* time, and reverse cascade in *descending* time.
6. The rows in the report support navigation (**S** line command) and data set recovery (**R** line command).
7. Other than the selected component, only steps that link to other steps are shown. Navigate to the job to see its full list of steps.

Generate a Safeguarded Copy report

The Safeguarded Copy report shows active jobs and data sets open for output at a specified time.

About this task

The Safeguarded Copy function provides backup copies to recover data in case of logical corruption or destruction of primary data. The Safeguarded Copy report lists all data sets open for output at a specified time. This is especially useful for determining whether any jobs or data sets captured by Safeguarded Copy were invalid because they were active at the time.

The JCL to run the report is located in the CNTL member DSACTIVE. The report uses the TimeLiner Data-In-Virtual (DIV) data space to lookup the job, step and data set history. The following JCL is displayed.

```
//IZBR110  JOB (),CLASS=A,MSGCLASS=X,REGION=256M,NOTIFY=&SYSUID
//*
//IZBRPROC JCLLIB ORDER=(HLQ.CNTL)
//*
//* PROGRAM:  BRNLACTI
//* FUNCTION: REPORT ON DATASETS OPEN FOR WRITING AT
//*           A POINT IN TIME
//*
//* OPTIONS
//*
//* DIVDSN(<DIVDSN>) - LOCATION OF TIMELINER DIV FILE
//* DTM(<TIMESTAMP>) - REPORT OPEN DATASETS AT THIS
//*                       TIME. YYYY/MM/DD HH:MM:SS.CS
//* DUR(<SECONDS>) -   DURATION IN SECONDS WHERE A DATASET
//*                       CAN BE OPEN (OPTIONAL)
//*
//BRNLACTI EXEC PGM=BRNLACTI
//STEPLIB  DD  DISP=SHR,DSN=HLQ.SBRNTLOD
//         DD  DISP=SHR,DSN=HLQ.SCKKLOAD
//         DD  DISP=SHR,DSN=HLQ.SBRNLOAD
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  DSN=&&REPORT1,DISP=(,PASS),UNIT=SYSDA,
//             SPACE=(CYL,(10,10)),
//             DCB=(RECFM=FBA,LRECL=133,DSORG=PS,BLKSIZE=0)
//TLNBAOPT DD    *
DIVDSN(HLQ.TLN.DIV)
DTM(2020/10/06 16:10:00.00)
DUR(60)
//*
//*
//*-------------------------------------------------------------------------
//* COPY REPORT(S) TO ONLINE LIBRARY
//*-------------------------------------------------------------------------
//COPYRPT1 EXEC COPYPRNT,NUMBER=1,MEMBER=DSACTIVE
```

Figure 146. DSACTIVE Safeguarded Copy JCL

The configuration options for the report are as follows:

DIVDSN
The TimeLiner DIV file

DTM
The Date and Time for reporting

Format
YYYY/MM/DD HH:MM:SS.CS

DUR
Duration in seconds where a data set can be open (OPTIONAL)

After running DSACTIVE the report is stored in the REPORTS database.

The Safeguarded Copy report, when complete, lists the Type, Name and other details for the jobs, steps and data sets that were active during the specified time - as shown in the following figure:

```
Datasets Report, Open for Writing

Open between 2020/10/06 16:10:00.00 and 2020/10/06 16:11:00.00

Type        Name                  Active   Start/Open             End/Close                Details
-------------------------------------------------------------------------------------------------------
Job         TIIII010(JOB34528)      No     2020/10/06 16:09:05.99  2020/10/06 16:10:06.09  CC=0
Step        STEP1(2)                No     2020/10/06 16:10:06.02  2020/10/06 16:10:06.04  CC=0
Output DS   IZBRA.TIIII.FILE010     No     2020/10/06 16:10:06.03  2020/10/06 16:10:06.03  DDName=SYSUT2, V
Step        STEP2(3)                No     2020/10/06 16:10:06.04  2020/10/06 16:10:06.06  CC=0
Output DS   IZBRA.TIIII.FILE011     No     2020/10/06 16:10:06.05  2020/10/06 16:10:06.05  DDName=SYSUT2, V
Job         TIIII011(JOB34529)      No     2020/10/06 16:10:06.10  2020/10/06 16:10:07.17  CC=0
Step        STEP1(1)                No     2020/10/06 16:10:06.10  2020/10/06 16:10:06.12  CC=0
Output DS   IZBRA.TIIII.FILE011     No     2020/10/06 16:10:06.11  2020/10/06 16:10:06.11  DDName=SYSUT2, V
Step        STEP2(2)                No     2020/10/06 16:10:06.12  2020/10/06 16:10:06.14  CC=0
Output DS   IZBRA.TIIII.FILE012     No     2020/10/06 16:10:06.13  2020/10/06 16:10:06.13  DDName=SYSUT2, V
Job         TIIII012(JOB34530)      No     2020/10/06 16:10:07.18  2020/10/06 16:10:07.27  CC=0
Step        STEP1(1)                No     2020/10/06 16:10:07.19  2020/10/06 16:10:07.21  CC=0
Output DS   IZBRA.TIIII.FILE012     No     2020/10/06 16:10:07.20  2020/10/06 16:10:07.20  DDName=SYSUT2, V
 ...
```

Figure 147. Safeguarded Copy report

TimeLiner maintenance tasks

TimeLiner generates a message log during operation. Timeliner can also be stopped to perform system maintenance.

TimeLiner messages

TimeLiner produces a number of messages as it monitors application jobs. Use the information provided here to determine the status of the TimeLiner system as it runs.

Table 65. TimeLiner messages

Message	Description
`BRNL265I JOB TIMELINER ELIGIBLE`	This message is displayed in the monitored job JESMSGLG and indicates the job, step and data set information that is being collected for the job. **Note:** A job is monitored only if the job's name matches an entry, either a fully qualified job name or a jobname mask, in the application table (APPTABLE).
`BRNL266I TIMELINER INITIALIZATION COMPLETE`	This message appears in the JESMSGLG of the TimeLiner job and indicates the address space is fully initialized.
`BRNL013I BRNLCT00 CALL TO 'CSRPQPL' NEAR LABEL 'MAIN1351' COMPLETED WITH COMPLETION CODE 28'`	If this message appears in the BRNLCT00 data set when you first start TimeLiner, it indicates the DIV data set is not yet initialized. You can ignore this message.

A complete table of TimeLiner messages can be found in "TimeLiner Messages" on page 380.

TimeLiner commands

TimeLiner has a number of commands available for the administration and query of TimeLiner processes.

To use, issue the MVS Modify command in the following format: F *TIMELINER,command*

For example:

```
F TIMELINER,DISPLAY ACTJOBS
```

where *TIMELINER* is the Started Task name TimeLiner is running under.

Table 66. TimeLiner commands

Command	Description
DIVSAVE	Writes the contents of the TimeLiner buffer to the DIV.
DISPLAY ACTJOBS	Displays all the Nametokens and shows all the currently monitored jobs, if any.
DISPLAY USECOUNTS	Displays the number of active jobs, jobs using current table and jobs using previous AppTable.
DISPLAY GROUP	Displays the PLEX group information.
DISPLAY GROUPTL	Displays the PLEX group information using TL data structures.
DISPLAY OJXLIST	Displays the job index in job ID order.
DISPLAY OJXCHAIN	Displays the job index in start time order.
RELOAD	Reloads the AppTable.
SWITCH	Posts the SYSPLEX ECB and checks the sysplex state.
DUMP	Takes a SVC dump and a transaction dump to dynamically allocated data sets. Note: The following needs to be specified in the parmlib member SVCDUMP=YES and TRANSACTIONDUMPS=YES. TimeLiner will take a snapshot dump and keep running.
MAP	Displays all the modules, their addresses and the eye catcher (including compiled date and time).
SET	Redrives the SET processing (the code that reads the TLNPRM00 member).

Stopping TimeLiner

You can stop or quiesce TimeLiner for a period of time for system maintenance, such as when you want to reload the SMF exit or SVC.

About this task

There are several ways to stop TimeLiner, depending on the situations described here.

Procedure

Stop TimeLiner using one of the following methods:

Method	Description
STOP,QUIESCE command	The **STOP,QUIESCE** command prevents TimeLiner from monitoring any new jobs and stops all exit intercepts. In a quiesced state, TimeLiner does the following: • Waits for any jobs that are currently being monitored to complete. • Processes the records that remain in the buffer until the buffer is empty. This ensures any inflight data is not lost. BRNL289I is issued after 10,000 records are processed until the buffer is empty and shutdown can continue. • Waits until any jobs that were currently examining the Application Table to finish examining the table.

Method	Description
	To view the jobs that are still being monitored, issue the DISPLAY,ACTJOBS command. The Job IDs of any currently monitored jobs are displayed. You can also issue **DISPLAY,USECOUNTS** to show how many jobs are: • Currently being monitored • Currently examining the Application Table
STOP command	If you do not want to wait until all monitored jobs have completed, you can issue /P TLINER or issue the STOP command to prevent TimeLiner from monitoring any new jobs. When TimeLiner is stopped: • TimeLiner becomes inactive immediately and all exit intercepts stop. It does not wait until all currently monitored jobs are completed before stopping. • Processes all records in the buffer until the buffer is empty. This ensures any inflight data is not lost. BRNL289I is issued after 10,000 are processed until the buffer is empty and shutdown can continue. • Waits until any jobs that were currently examining the application table to finish examining the table. You can also issue the **DISPLAY**,**ACTJOBS** and **DISPLAY,USECOUNTS** commands as described above.

In a SYSPLEX group	If you are running more than one TimeLiner STC in a SYSPLEX group, start by shutting down each secondary TimeLiner, and allow each to terminate completely.
	You can then shut down the primary TimeLiner.
	To determine which TimeLiner STCs are secondary or primary, issue the DISPLAY,GROUP command.
	The output is similar to the following:
	`BRNL225I TLINRNJK DISPLAY,GROUP COMMAND RECEIVED FROM SNJK$`
	`BRNL304I MEMBER: S0W1 JOBNAME: TLINRNJK STATE: ACTIVE STATUS: PRIMARY`
	`BRNL304I MEMBER: S0W2 JOBNAME: TLINRNJK STATE: ACTIVE STATUS: SECONDARY`
	You can also use the above information to determine which TimeLiner jobs, on each system, are primary or secondary.

Reloading the Application Table

If updates were made to your application table since the time you started TimeLiner, reload the updated table into the TimeLiner address space.

Procedure

To reload the Application table, issue the following command:

```
/F TLINER,RELOAD
```

The following message is displayed:

```
TLN225I TLINS300 RELOAD COMMAND RECEIVED FROM user
```

When the command completes, the following message is issued:

```
TLN266I TABLE INITIALIZATION COMPLETE
```

If the command is issued and already running, this message is issued:

```
TLN006I TABLE RELOAD ALREADY RUNNING
```

If the RELOAD is already running, wait for it to complete before issuing another RELOAD.

Reloading the SVC or SMF exit

Optionally, you can reload the SVC or SMF exits as instructed by Technical Support.

About this task

CAUTION: The SVC and SMF exits typically do not need to be reloaded. Perform these steps only when directed by a member of the Technical Support team.

Procedure

1. Stop TimeLiner.
2. Update the TimeLiner parameters to include **REINIT(YES)**.
3. Restart TimeLiner.
4. Reset the TimeLiner parameters to specify **REINIT(NO)**.

Appendix A. IBM Z Batch Resiliency Parameters

This section describes the IBM Z Batch Resiliency parameters contained in the USERMAC(VFIPARMS) member.

The following figure shows an example of the VFIPARMS member with default values.

```
AUDIT  = 1
RETPD1 = 2
CRITSW = 1
RETPD2 = 0
CONAME = your company name
DMS = FDR
TMS = RMM
POPTIONS=CENTER
RECOVERY=APPLICATION
REPORTS=ONLINE
RPTGENS=5
RPTRETPD=999
EXHPDM=NO
AUDTDSN = hlq.IZBR.V1R1M0.AUDITDB
HISTDSN = hlq.IZBR.V1R1M0.HISTORY
ARETPD1 = 2
ARETPD2 = 1
```

Figure 148. USERMAC(VFIPARMS) with default values

When editing a parameter value, include a blank space before and after the equal sign (=) as shown in this example:

```
RETPD1 = 2
```

IBM Z Batch Resiliency parameter description (VFIPARMS)

The VFIPARMS parameters and values are described in the following tables.

Table 67. VFIPARMS parameters and values

Parameter	Description
AUDIT = 1 or 0	Set to 1 to create an Audit database. If set to 0, no Audit database is created.
RETPD1 = *n*	Set the number of days, counting backward from the current date that SMF history data is retained. The default is two days. The length of time specified for the retention period, depends on how often you run the Collection and Critical File Identification Analysis process. The size of the History database depends on factors such as the number of production jobs run daily, and the number of files opened by these jobs.
CRITSW = 1 or 0	Set to 1, all files with a disposition of OLD or MOD are automatically considered critical files and are placed in the critical file list. Set to 0, files with a disposition of OLD or MOD are listed as exception files on the MAINTLST report and are not added to the critical file list.

© Copyright IBM Corp. 2019, 2020

Table 67. VFIPARMS parameters and values (continued)	
Parameter	**Description**
RETPD2 = *n*	Sets the number of days, counting backwards from the current date, for which resource requirements analysis data is retained. Resource usage data (SMFtype30 / IZBRsubtype3) is produced on a job step basis and creates the largest number of SMF records collected by IBM Z Batch Resiliency. As a best practice, keep the retention period for the resource data parameter (RETPD2=) to a minimum. Because sizing for recovery purposes is an analysis process which is done infrequently, it is a best practice to turn off resource usage data collection (RETPD2=0) until resource reporting is required.
CONAME = *Company name*	Contains the company name that appears on all IBM Z Batch Resiliency reports (16 characters maximum). Do not include a comma in this field.
DMS = DMS, FDR, DSS or ABARS	Specify DMS, FDR, DSS, or ABARS to indicate the default disk management system in use. Backup and restore JCL and control cards are created in the format for the indicated disk management system.
TMS = CA1, TLMS, CTT, RMM or ZARA	Specify CA1, RMM, CTT (for Control-M/Tape, previously known as Control-T), ZARA or TLMS to indicate the tape management system in use. Backup verification processing uses this setting to obtain information from the specified tape management system.
POPTIONS = CENTER or NOCENTER	Controls the justification on IBM Z Batch Resiliency reports. Specify 'CENTER' to center report data between margins. Specify 'NOCENTER' to left-justify reports.
RECOVERY = APPLICATION or GLOBAL	Specify the type of recovery used at your particular site. If APPLICATION is specified, each application defined in the APPTABLE is treated on an individual basis. If GLOBAL is specified, all defined applications are treated as one large application. When GLOBAL is used, recovery must begin with the first application that starts the daily cycle. Typically, the first job for the GLOBAL application is the Tape Management System Vault job. See “Real-Time critical file analysis” on page 69 for additional details on this parameter.
REPORTS = SYSOUT or ONLINE	Specify SYSOUT to route reports to the SYSOUT, or ONLINE to route reports to the IBM Z Batch Resiliency Reports library for on-line viewing.
RPTGENS =	Set the number of generations, most current generations, for which reports are kept. Default is 5 generations. Unlimited RPTGENS is specified as 0.
RPTRETPD =	Set the number of days, from the current date, for which reports are kept. Default is 999 days. Unlimited RPTRETPD days is specified as 0.
EXHPDM =	Set to YES if using StorageTek ExHPDM software; otherwise, NO.
AUDTDSN =	AUDIT data set name
HISTDSN =	HISTORY data set name

Table 67. VFIPARMS parameters and values (continued)	
Parameter	**Description**
ARETPD1 =	Set the number of days, counting backwards from the current date, for which Non-production SMF analysis data is retained in the Auditdb. The length of time specified for the Auditdb retention period depends on the planned frequency for running the Collection and Critical File Identification analysis process. The size of Auditdb depends on factors such as number of non-production jobs run daily, and the number of files opened by these jobs. It is a best practice to perform critical file analysis daily. To support this, ARETPD1= should be set to two days so data from application cycles that run past midnight are available.
ARETPD2 =	Set the number of days, counting backwards from the current date for which non-production SMF resource usage data is retained in the Auditdb. Resource usage data (SMFtype30 / IZBRsubtype3) is produced on a job step basis and creates the largest number of SMF records collected by IBM Z Batch Resiliency. Auditdb contains all non-production SMF records collected and can be very large. For this reason, you may want to keep the retention period for the resource data parameter to a minimum. (ARETPD2=0).

Optional VFIPARMS and values

Parameter	Description
INVLSRI =	INVENTRY file LSR pool index definition. This value defines the number of index buffers for the LSR pool. The default is 1000.
INVLSRD =	INVENTRY file LSR pool data definition. This value defines the number of data buffers for the LSR pool. The default is 2000.
GBLSTAT =	Global Status processing option is set to YES or NO. When YES is specified, program BRNVC010 is invoked at SMF interval processing. The default is NO.
KEYDHIST =	Keyed History processing option is set to YES or NO. When YES is specified, the optional Keyed History files are being updated automatically. The default is NO.
BRNVCPY =	**AUTO**: Use this value to indicate to ANALYSIS that when processing a tape data set to substitute the value BRNVCPY for the value specified for the disk management field (DMS=). **DEFAULT**. NO: Use the default DMS= value for all data sets.
SIMRCVRY =	Turn on Simulated Recovery at startup (YES or NO). The default is NO. See Virtual Recovery Simulatorfor more information about Simulated Recovery.

Appendix B. Data Sets and Libraries

This appendix lists each IBM Z Batch Resiliency data set and library by name and provides a corresponding explanation.

Table 68. Data sets and libraries

File name	Purpose
hlq.AUDITDB	The Audit database. This file contains a list of all files that were encountered but were not part of an application defined in USERMAC(APPTABLE). It can be used as an audit mechanism to determine if production files are being in non-production jobs.
hlq.AUDITL2	Contains discarded records filtered out of the AUDITDB file during the HISMAINT process.
hlq.AUDTDS	An intermediate file used by Real-Time Analysis to store audit data (that is, non-production files). This file is flushed every 10 minutes to the AUDITDB file.
hlq.BAUDFLAT	A backup of the BKUPAUD data set (see below).
hlq.BKUPAUD	This file tracks all changes to the backup table and can be used to recreate the backup table if it should become corrupted (contact Technical Support for help with this).
hlq.BKUPJCL	A PDS containing the ABARS Verify JCL if ABARS is a selected backup method. Otherwise, this file is not used.
hlq.BKUPJCL.BKUPMETH	Generated Backup JCL for a particular *backup method*. See the BKUPMETH file below.
hlq.BKUPMETH	A small data set containing a list of all supported backup methods and which of those are in use in this installation. This file also contains the name of the backup and restore data sets for the backup methods.
hlq.BKUPTBL	A VSAM KSDS used to track backups of critical DATASETS. This file is used in the generation of the restore JCL. It is needed at a disaster recovery site to generate RESTORE JCL.
hlq.BKUPFLAT	The unloaded BKUPTBL. This file is used to back up the backup table and by restoring it, reorganizes the BKUPTBL, a VSAM KSDS.
hlq.CHKPOINT.SYSxxxx	Checkpoint file used by SMF extractor. xxxx refers to the SYSID of the system from which the extractor is run.
hlq.CNTL	JCL to run IBM Z Batch Resiliency.
hlq.CRITICAL	The Critical File List. It is used for reporting only in the non-SAS releases.
hlq.DB2BRNCR	Critical Db2 files.
hlq.DB2BRNLG	Db2 Image Copy files.
hlq.DIST.LIBRARY	The installation control JCL for IBM Z Batch Resiliency.
hlq.DMS	This file is obsolete.
hlq.DOCLIB	IBM Z Batch Resiliency documentation library.
hlq.HISTORY	IBM Z Batch Resiliency History database. The keyed history feature database is built from this file.

© Copyright IBM Corp. 2019, 2020

Table 68. Data sets and libraries (continued)

File name	Purpose
hlq.HISTRYL2	Contains discarded records filtered out of the HISTORY file during the HISMAINT process.
hlq.HSMACT	Sequential data set used to track ABAR log data sets read by program ABARBKUP.
hlq.IMS.RECONLST	An unloading of the IMS RECON file.
hlq.IMS.RECONOUT	Processed RECONLST to isolate backup information for IMS.
hlq.IMS.RESPRT	List of IMS RESLIB files.
hlq.INVENTRY	The IBM Z Batch Resiliency Inventory database. This file contains data about all files defined to production applications.
hlq.MODEL	Model DSCB for IBM Z Batch Resiliency Analysis database and Backup Table Recovery Files.
hlq.ODTCKPT	A checkpoint file for the Real-Time Analysis process. This file and the RTACKPT file provide backup of the data in memory for the Real-Time Analysis process in case something causes it to abnormally terminate.
hlq.PRODDS	SMF data relating to production data set from the last collection.
hlq.REPORTS	IBM Z Batch Resiliency reports library.
hlq.RESTJCL	Data set name for PDS containing IBM Z Batch Resiliency-generated restore JCL (disk). This file is empty until CREAJCL has executed. (Version 3.3 compatibility).
hlq.RESTJCL.BKUPMETH	PDS data set to contain JCL / control statements to restore critical files successfully backed up in the backup table. BKUPMETHOD refers to the utility used to backup files.
hlq.RESTTBL	This table tracks files that have been restored and is necessary to be at a disaster recovery site. If, for some reason, any of the RESTJCL.bkupmeth libraries should be corrupted at the disaster recovery site, or it is necessary to restart the restore process, running the RESTJCL job taking this file as an input creates the JCL to restore the outstanding files.
hlq.RTACKPT	A checkpoint file for the Real-Time Analysis engine. This file and the ODTCKPT file provide backup of the data in memory for the Real-Time Analysis process in case something causes it to abnormally terminate.
hlq.RTCMDQ	This small data set is used to pass commands to the Real-Time Analysis function.
hlq.SBRNLOAD	IBM Z Batch Resiliency load module library.
hlq.SCANJCL	Data set generated by DC/OPTIMIZE 2000 process which contains concatenated data set information.
hlq.SMFEXT.ssss.SMF $yy.Tjjjhhmm	SMF extract file created by the SMF extractor. *ssss* refers to the System Name on which the data was created. *yy* refers to the year the file was created. *jjj* refers to the Julian Date the file was created. *hhmm* refers to the time the file was created (24 hrs).
hlq.SBRNSRCE	This PDS contains the SAS source code for the SAS IBM Z Batch Resiliency version.
hlq.SCKKLOAD	IBM Z Batch Resiliency load module library for SMF extractor which should run as a started task.
hlq.TAPEJCL	Data set name for PDS containing IBM Z Batch Resiliency-generated backup JCL (tape). This file is empty until TAPEJCL has executed.

Table 68. Data sets and libraries (continued)	
File name	**Purpose**
hlq.USERMAC	IBM Z Batch Resiliency user code library. User exits and IBM Z Batch Resiliency customizations are stored here.
hlq.SBRNEXEC	IBM Z Batch Resiliency online user interface execs.
hlq.SBRNMLIB	IBM Z Batch Resiliency online user interface messages.
hlq.SBRNPLIB	IBM Z Batch Resiliency online user interface panels.
hlq.SBRNSLIB	IBM Z Batch Resiliency online user interface skeleton library.
hlq.SBRNTLIB	IBM Z Batch Resiliency online user interface table library.
hlq.VTOCDUMP	Output from the VTOCDUMP program.

Appendix C. USERMAC Member Description

This appendix lists the members in the USERMAC library, along with their functions.

Table 69. USERMAC member descriptions

Module name	Description
$$CPYR	Proprietary statement
$DELETE$	Delete production data sets
$DELHSM$	Delete all HSM-migrated data sets
$RNLDIST	Code to copy HLQ.DIST.LIBRAR into BRNDISTL
#APPCNT	File layout for #APPCNTJ
#AUDITCK	REXX file layout for Audit report
#BACKRPT	REXX file layout for backup report
#BKUPALC	File layout for RESTALC report
#BKUPDUP	REXX file layout for BKUPDUP report
#BKUPTBL	REXX file layout for backup table
##CRIT	File layout for Critical File List
#CRITRMV	File layout for CRITRMV report
#CRITRPT	File layout for CRITRREPT report
##DB2RPT1	File layout for DB2BACK report
##DB2RPT2	File layout for DB2BACK report
##DB2RPT3	File layout for DB2BACK report
##DB2RPT4	File layout for DB2BACK report
##DLMRPR1	Parameters for DLM report
##DLMRPR2	Parameters for DLM report
##DLMRPR3	Parameters for DLM report
##DLMRPR4	Parameters for DLM report
##DLMRTM1	File layout for DLM report
##DLMRTM2	File layout for DLM report
##DLMRTM3	File layout for DLM report
##DLMRTM4	File layout for DLM report
#DORMANT	File layout for DORMANT report
#FREQRPT	Frequency report exit
#IMSRPT1	Parameters for IMS report
##INVBASE	File layout for Inventory Base Records
##INVCOMP	File layout for Inventory Component record
#INVDUPS	File layout for Inventory Duplicates report

© Copyright IBM Corp. 2019, 2020

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
##INVTBAS	File layout for Inventory Base record
##INVTCMP	File layout for Inventory Component record
#JOBRPTR	REXX exit for JOBRPTR report
##PULLLST	File layout for #PULLLST report
##PULL1	Pullist report file layout
##PULL2	Pullist report file layout
##RCRIT	Parameters for CRITRPT
##RCRITN	Parameters for CRITRPT used by CRITNEW report
##RECALAP	File layout for #RECALAP report
##RJCRPT1	File layout for RESTJCL
##RPULL1	Parameters for BRNLLIST report #1
##RPULL2	Parameters for BRNLLIST report #2
##RPULLA	Parameters for TAPE PULL LIST REPORT: SORTED BY APPLID
##RPULLDS	Parameters for TAPE PULL LIST Report: Sorted by VOLSER
##RPULLV	Parameters for TAPE PULL LIST Report: Sorted by BACKUP data set
##RVLTRPT	Parameters for TAPE BACKUP VAULT EXCEPTION Report: Sorted by BKPVOL
##VDUCPGR	File layout for #VLTRPT report
##VDULIST	Macro to build an options module for SMFDSSU
##VDUOPTG	Macro to build an options module for SMFDSSU
##VDUOPTM	Map SMFDSSU options that are generated
##VLTRPT	File layout for #VLTRPT report
##XDEP	Cross-dependency report file layout
#ZDMFBSZ	File layout for ZDMFBSZ report
AABARS	Backup control statement template for ABARS
ABARBKUP	Macro call for ABARLOG program
ABARCPY0	File layout for ABARCPY0 report
ABARPAT	Mask for ABARS logs to be read. Used by ABARBKUP
ABARVFY	Optional parameters for ABARVFY - SAS format
ACTM	File layout for ACTM report
ADPRT	File layout for ADPRT report
ADSS	JCL template for generating DFDSS backup JCL
ADXREF	File layout for ADXREF report
AFAVER	JCL template for generating FAVER backup JCL
AFDR	JCL template for generating FDR backup JCL

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
APPLDEF	Table used to accomplish application splitting
APPLDEFZ	File layout for APPLDEFZ report
APPTABLE	The application definition table
APPTDUP	File layout for APPTDUP report
ATAPECPY	JCL template for generating BRNVCPY backup JCL
AUDITCK	File layout for AUDITCK report
AZDMFJCX	File layout for AZDMFJCX report
BACKJCL	Run time options for BACKJCL
BACKJCLS	File layout for BACKJCLS report
BACKJSRT	File layout for BACKJSRT report
BACKMANT	Run time options for BACKMANT
BACKREPT	Parameters for BACKUP report
BACKUPR	Invoking exit for the BACKUPR utility
BACKWO	File layout for BACKWO report
BALLOCAT	Used to allocate data sets during installation
BDMS	JCL template for creating DMS restore JCL
BDSS	JCL template for creating DF/DSS restore JCL
BFAVER	JCL template for creating FAVER restore JCL
BFDR	JCL template for creating FDR restore JCL
BFDRFULL	JCL template for creating FDR full-volume restore JCL
BFULLDSN	JCL template for creating DF/DSS restore JCL
BFULLDSS	JCL template for creating DF/DSS restore JCL for the entire contents of a backup tape
BFULLFDR	JCL template for creating FDR full-volume restore JCL
BGINPARM	Input parameters for BGMAIN
BKUPDUPR	File layout for BKUPDUPR report
BKUPFULL	File layout for BKUPFULL report
BKUPHLQ	During Installation, this member defines the high-level qualifier for backup files created under the control of IBM Z Batch Resiliency. The value of the BKUP_HLQ parameter should be limited to 19 characters or less
BKUPJCL	File layout for BKUPJCL report
BKUPJCLA	File layout for BKUPJCLA report
BKUPJOB	Job card used by processes that generate backup JCL
BKUPTOT	File layout for BKUPTOT report
BKUPVALT	REXX exit to check remote offsite location indicator for offsite vault

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
BKUPVALX	REXX exit used by BRNIMSRC
BLDXCAT	File layout for BLDXCAT report
BTAPECPY	JCL template for creating BRNVCPY restore JCL
BTMSRENM	This module is compiled from the DIST.LIBRARY based on the Tape Management System choice
BTMSRMM	JCL template for generating RMM JCL to rename tape data sets
CATEGORY	File layout for CATEGORY report
CFEDITL1	File layout for CFEDITL1 report
CFEEDITL	File layout for CFEEDITL report
CFEEDRPP	File layout for CFEEDRPP report
CFEEDSRT	File layout for CFEEDSRT report
CRITREMV	File layout for CRITREMV report
CRITREPT	CRITREPT parameters
CRITRPTR	CRITREPT parameters
CRITRPTT	File layout for CRITRPTT report
CRITTIER	File layout for CRITTIER report
CROSSDEP	File layout for CROSSDEP report
CTMLIST	File layout for CTMLIST report
CYCDELR	File layout for CYCDELR report
DB2	File layout for DB2 report
DB2BACKR	File layout for DB2BACKR report
DB2PULL	File layout for DB2PULL report
DB2RPT01	File layout for DB2RPT01 report
DB2RPT02	File layout for DB2RPT02 report
DB2RPT03	File layout for DB2RPT03 report
DB2RPT04	File layout for DB2RPT04 report
DB2RPT05	File layout for DB2RPT05 report
DCOLFILT	File layout for DCOLFILT report
DCOLPARM	File layout for DCOLPARM report
DCOLVOL	File layout for DCOLVOL report
DCOPTPRM	DC/Optimizer 2000 parameter list used for dynamic allocation of work files
DCOPTS1	DC/Optimizer 2000 sort control statements – Do not change
DCOPTS2	DC/Optimizer 2000 sort control statements – Do not change
DCOPTS3	DC/Optimizer 2000 sort control statements – Do not change
DCOPTS31	DC/Optimizer 2000 sort control statements – Do not change

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
DCOPTS4	DC/Optimizer 2000 sort control statements – Do not change
DCOPTS5	DC/Optimizer 2000 sort control statements – Do not change
DORMANT	SAS macro to customize and invoke the DORMANT report
DSSCARD	Command statements for DSS used by XDSSBKUP process
DSSCARDR	Command statements for DSS used by YDSSBKUP process
DSSDUMPF	File layout for DSSDUMPF report
DSSDUMPP	File layout for DSSDUMPP report
ESPLIST	File layout for ESPLIST report
EXCLBKP	File layout for EXCLBKP report
EXCLDSN	The data set exclusion list
EXCLJOB	The JOB exclusion list
EXHPDM	Used by the StorageTek XDSSSOV process
FDRCARD	FDR command statement used by XFDRBKUP SAS program
FDRCARDF	File layout for FDRCARDF report
FDRCARDI	File layout for FDRCARDI report
FDRCARDV	File layout for FDRCARDV report
FREQRPTR	REXX code excerpt to provide parameters for the FREQRPTR report
GRPPARMS	Backup grouping parameters used throughout the Backup Grouping programs and the backup processes
IMSDBJCL	REXX code excerpt to add IMS databases explicitly defined in IMS startup JCL
IMSMDALB	REXX code excerpt to eliminate IMS databases that do not require image copies
IMSRESLB	REXX code excerpt to define the IMS DD file concatenation
IMSRPT01	File layout for IMSRPT01 report
IMSRPT02	File layout for IMSRPT02 report
INCLALC	Used by BKUPALC
INCLDB2	Used by INCLDB2
INCLDSN	Data sets to be included in the backup process whether they are critical or not
INCLMOVE	Used by MOVEJCL
INCLVOLS	Data set used by BKUPFULL to specify volumes to be backed up
INVDUPS	File layout for INVDUPS report
INVMAINT	Data input to INVMAINT job
INVREMOV	File layout for INVREMOV report
INVTYRPT	File layout for INVTYRPT report
JOBEXIT	User exit for job record modifications. Used by READHIST

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
JOBEXIT	User exit for job record modifications. Used by READHIST
JOBIRPT1	File layout for JOBIRPT1 report
JOBISRT1	File layout for JOBISRT1 report
JOBLRPT1	File layout for JOBLRPT1 report
JOBLSRT1	File layout for JOBLSRT1 report
JOBRPTR	REXX code excerpt to provide parameters for the JOBRPTR report
KEYDRBLD	File layout for KEYDRBLD report
KEYDUNLD	File layout for KEYDUNLD report
KEYJRBLD	File layout for KEYJRBLD report
KEYJUNLD	File layout for KEYJUNLD report
LEGFILE	File layout for LEGFILE report
LEGTABLE	Table used to define legacy data sets, used by SCANSMF and BRNVIUJV
LOUTHINP	File layout for LOUTHINP report
LOUTHOUT	File layout for LOUTHOUT report
MAINTLST	File layout for MAINTLST report
MIGBKPRM	File layout for MIGBKPRM report
MIGBKUPM	File layout for MIGBKUPM report
MJCLBFUL	File layout for MJCLBFUL report
MJCLCTMJ	File layout for MJCLCTMJ report
MJCLMIGJ	File layout for MJCLMIGJ report
MJCLREST	File layout for MJCLREST report
MOVEJCL	File layout for MOVEJCL report
PULLLIST	File layout for PULLLIST report
RECALAPP	File layout for RECALAPP report
RECALIST	File layout for RECALIST report
RESTJCLR	Parameters for the REXX RESTJCLR program
RESTJOB	The job card used for JCL generated by the RESTJCL program
REXXRCV	REXX code excerpt to check for the availability of the REXX/370 library
RJCRPT01	File layout for RJCRPT01 report
RTSRTDRT	IBM Z Batch Resiliency real-time SMF startup parameters, stop when out of SMS data sets
RTSRTD00	IBM Z Batch Resiliency real-time SMF startup parameters, keep running
RTVRSPRM	File layout for RTVRSPRM report
RTXPARMS	File layout for RTXPARMS report
SCANALC	File layout for SCANALC report

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
SCANFILT	REXX code excerpt consisting of scan filters to drop DD statements for specific utilities
SCANMETH	SCANMETH, a REXX code excerpt, associates program names to backup methods; the default is BKPMETH = PGMNAME(SMFPGM). It is used to associate an 8-character program name to 7 char method. For example, IEBGENER cannot be a BKUPMETH since it is 8 characters long
SCHDPARM	Parameter data for the Backup Scheduling process
SCHDREQ	Scheduling request table
SMFDSSUO	External options module for SMFDSSU
SMFFDRO	External options module for MFFDR
SNAPTABL	File layout for SNAPTABL report
SNPEXIT1	File layout for SNPEXIT1 report
SNPEXIT2	File layout for SNPEXIT2 report
SORTRES1	File layout for SORTRES1 report
SRTKEYD1	File layout for SRTKEYD1 report
SRTKEYD2	File layout for SRTKEYD2 report
SRTKEYJ1	File layout for SRTKEYJ1 report
SRTKEYJ2	File layout for SRTKEYJ2 report
SRTLEG01	File layout for SRTLEG01 report
SRTMNTLA	File layout for SRTMNTLA report
SRTMNTLD	File layout for SRTMNTLD report
STACEXIT	Stacked volume customization exit
IBM Z Batch Resiliency	IBM Z Batch Resiliency Installation parameters
TAPEJOB	JOB statement for BRNVCPY backups
TAPESCAN	File layout for TAPESCAN report
TCALTTBL	File layout for TCALTTBL report
TCDSSIN	Control statements for DSS used by XTAPECPY and XTAPESIM
TCPARMS	Parameter file for the BRNVCPY utility
TIMCHKR	File layout for TIMCHKR report
TLMSRENM	Null parameter set for the TLMSRENM program
TLNPRM00	File layout for TLNPRM00 report
TMCPASS	Tape Management System password, used by the YTAPECPY process
TMS5RENM	Null parameter set for the TLS5RENM program
UFBKFULL	File layout for UFBKFULL report
UFBKPJCL	File layout for UFBKPJCL report
UFCTMJBB	File layout for UFCTMJBB report

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
UFREST	File layout for UFREST report
UFRSTALC	File layout for UFRSTALC report
UMIGBKUP	File layout for UMIGBKUP report
VAULTLST	File layout for VAULTLST report
VB11EXIT	File layout for EXIT report
VFIFDSS	File layout for VFIFDSS report
VFIFXD	Sample FDR exit for user customizable SMFDSSU parameters
VFIFXZ	Dummy version of FDR Open Exit and Dataset Exit
VFIPARMS	Parameters for IBM Z Batch Resiliency used throughout the software
VFIRTD00	IBM Z Batch ResiliencyReal-Time Startup Parameters used by the RTSMF procedure and RTBATCH job
VFISTASK	IBM Z Batch Resiliency Started Task Parameters
VFITABLS	Defines the in-storage IBM Z Batch Resiliency tables
VLTSORT	File layout for VLTSORT report
VTOCDUMP	File layout for VTOCDUMP report
XABARVFY	File layout for XABARVFY report
XBACKMNT	File layout for XBACKMNT report
XBACKMN1	File layout for XBACKMN1 report
XBACKMN2	File layout for XBACKMN2 report
XBACKUPR	File layout for XBACKUPR report
XBACK01	REXX field layout of the backup file
XBKPMK01	Parameter to the BACKJCL
XBKUPDML	File layout for XBKUPDML report
XBKUPDM1	File layout for XBKUPDM1 report
XBKUPDUP	File layout for XBKUPDUP report
XBKUPFUL	File layout for XBKUPFUL report
XBKUPJCL	Sample exit code for BKUPJCL
XBKUP001	Sample exit code point 1 for BKUPJCL
XBUTABLE	Exit used to assign APPLIDs to pseudo-APPLIDS for the BUTABLE program
XCASJOB	File layout for XCASJOB report
XCASJOB1	File layout for XCASJOB1 report
XCA7JOB	SAS exit to customize the creation of an application table from CA7 using the CA7JOB
XCA7JOB1	File layout for JOB1 report
XCRITRPT	SAS exit to customize the Critical File report

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
XCTMJOB	SAS exit to customize the CTMJOBB utility
XCTMJOB1	SAS exit to customize the CTMJOBB utility
XCTMLIST	File layout for XCTMLIST report
XCTMREAD	File layout for XCTMREAD report
XDATAEXT	File layout for XDATAEXT report
XDB2READ	File layout for READ report
XDEPRPTR	Parameters for the REXX XDEPRPTR report
XDORMANT	File layout for XDORMANT report
XESPJOB	SAS exit to customize ESPJOB, the utility to create an application table from the ESP procedure libraries
XESPJOB1	File layout for XESPJOB1 report
XFULDSS1	File layout for XFULDSS1 report
XFULFDR1	File layout for XFULFDR1 report
XGROUP	REXX exit used by the Backup Grouping processes to associate a particular application to a particular backup group
XIMSBKRP	File layout for XIMSBKRP report
XJOBEXIT	File layout for XJOBEXIT report
XJOBINIT	SAS exit to customize the JOBINIT report which lists all jobs for all applications in run time order
XJOBLST1	File layout for XJOBLST1 report
XJOBRPTR	File layout for XJOBRPTR report
XJOBTRC	File layout for XJOBTRC report
XJOBTRC1	File layout for XJOBTRC1 report
XLEGACY	Sample SAS exit to customize the LEGACY report
XMAKEJCL	REXX code excerpt to supply parameters to the BACKJCL and RESTJCL utilities
XMIGBK01	File layout for XMIGBK01 report
XMKJCLB	File layout for XMKJCLB report
XMOVEXIT	File layout for XMOVEXIT report
XOPCINT1	Exit used by OPCJOB
XOPCINT2	Exit used by OPCJOB
XOPCJOB1	REXX User exit for OPC Job/Application selection
XRESTALC	File layout for XRESTALC report
XREST001	REXX exit code used to customize the RESTJCLR procedure which creates restore JCL
XSCANALC	File layout for XSCANALC report

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
XTAPESCN	File layout for XTAPESCN report
XTIERGRP	File layout for XTIERGRP report
XTWSJOB	File layout for XTWSJOB report
XTWSJOB1	File layout for XTWSJOB1 report
XZEKEJB	File layout for XZEKEJB report
XZEKEJB1	File layout for XZEKEJB1 report
ZDMFBSZ	File layout for ZDMFBSZ report
ZDMFRPT1	File layout for ZDMFRPT1 report
ZDMFSRT1	File layout for ZDMFSRT1 report
ZDMFSRT2	File layout for ZDMFSRT2 report
ZDMFSRT3	File layout for ZDMFSRT3 report
ZEKECMDS	File layout for ZEKECMDS report
SMFDSSUO	External options module for SMFDSSU
SMFFDRO	External options module for SMFFDR
STACEXIT	Stacked volume customization exit
SYMBOLS	IBM Z Batch Resiliency Installation parameters
TAPEJOB	JOB statement for BRNVCPY backups
TCDSSIN	Control statements for DSS used by XTAPECPY & XTAPESIM
TCPARMS	Parameter file for the BRNVCPY utility
TLMSRENM	Null parameter set for the TLMSRENM program
TMCPASS	Tape Management System password. Used by the YTAPECPY process
TMSFMT	Proc format for TMS location indicators
TMS5RENM	Null parameter set for the TLS5RENM program
VFIFXD	Sample FDR exit for user customizable SMFDSSU parameters
VFIFXZ	Dummy version of FDR Open Exit and Dataset Exit
VFIJ011S	Version 3 to Version 4.1 Inventory File Conversion program
VFIPARMS	Parameters for IBM Z Batch Resiliency used throughout the software
VFIRTD00	IBM Z Batch Resiliency Real-Time Startup Parameters used by the RTSMF procedure and RTBATCH job
VFISTASK	IBM Z Batch Resiliency Started Task Parameters
VFITABLS	Defines the in-storage IBM Z Batch Resiliency tables
XBACK01	REXX field layout of the backup file
XBKPMK01	Parameter to the BACKJCL
XBKUPJCL	Sample exit code for BKUPJCL
XBKUPMNT	Sample exit code for BKUPMNT

Table 69. USERMAC member descriptions (continued)	
Module name	**Description**
XBKUPMN1	Sample exit code for BKUPMNT
XBKUP001	Sample exit code point 1 for BKUPJCL
XBUTABLE	Exit used to assign APPLIDs to pseudo-APPLIDS for the BUTABLE program
XCA7JOB	SAS exit to customize the creation of an application table from CA7 using the CA7JOB
XCRITRPT	SAS exit to customize the Critical File report
XCTM0001	CTM Scheduler Parser
XCTMJOB	SAS exit to customize the CTMJOBB utility
XCTMJOB1	SAS exit to customize the CTMJOBB utility
XDEPRPTR	Parameters for the REXX XDEPRPTR report
XESPJOB	SAS exit to customize ESPJOB, the utility to create an application table from the ESP procedure libraries
XGROUP	REXX exit used by the Backup Grouping processes to associate a particular application to a particular backup group
XHISTRPT	Exit to customize the HISTRPT report which prints records in the History database
XJOBINIT	SAS exit to customize the JOBINIT report which lists all jobs for all applications in run time order
XLEGACY	Sample SAS exit to customize the LEGACY report
XMAKEJCL	REXX code excerpt to supply parameters to the BACKJCL and RESTJCL utilities
XOPCJOB1	REXX User exit for OPC Job/Application selection
XREST000	REXX exit code used to customize the RESTJCLR procedure which creates restore JCL
XREST001	REXX exit code used to customize the RESTJCLR procedure which creates restore JCL

Appendix D. IBM Z Batch Resiliency Defined Enqueues

The following table lists the defined enqueues for IBM Z Batch Resiliency.

Table 70. IBM Z Batch Resiliency defined enqueues

QNAME	RNAME	Scope	Component	Description
RTSMF001	RTCMD.PROCESSOR .ACTIVE.QUEUE	Systems	RT/SMF (Module CKKRC004)	RT/SMF Command Processor
RTSMF001	CMDQUEUE.UPDATE	Systems	RT/SMF (module CKKRC004)	RT/SMF Command Processor
RTSMF002	RTCMD.TPUT	Systems	RT/SMF (modules CKKRC006 and CKKRC008)	RT/SMF Command Processor Interface
BRNVYS00	LISTCAT	Systems	SMF Extractor	
BRNVYS00	BRNVP000	Systems		
BRNVYS00	SMF data set	Systems		
BRNVYS01	BKUPTBL data set	Systems	BKUPJCL	
BRNVYS01	BKUPTBL data set	Systems	RT/SMF (Module BRNVC007)	Real-Time Backup Table subtask
BRNVYS01	RESTTBL data set	Systems	RESTJCL	
BRNVYS02	Extractor $ data set	Systems	SMF Extractor	
BRNVYS02	Extractor $ data set	Systems	RT/SMF (Module CKKRC030)	RT/SMF SMF$ Dynamic Allocation Routine
BRNVPT01	REPORTS data set	Systems	REPORTS	
VSAMIO	RETRYOPEN	Systems	RT/SMF (Module CKKRC016)	Serializes the VSAM Verify operation if required at open
BRNVBG001	TMF (tape marker file) data set name	Systems	Backup Grouping	Used to enqueue the tape marker file on the individual backup tapes.
BRNVBG002	*hlq*.BGHUNT.CATALOGSEARCH data set	Systems	Backup Grouping	

© Copyright IBM Corp. 2019, 2020

Appendix E. BRNVCPY Alternate Return Code Table

Throughout execution, BRNVCPY issues processing messages.

The messages that BRNVCPY issues are designed to report on both positive and negative progress for each input copy request. All messages are categorized as one of: informational, warning, error, or terminating. Each category has a return code level default assigned. All but informational messages can be assigned individual alternate return codes using the BRNVCPY Alternate Return Code Table, TCALTTBL. Messages associated with the TCMSG= value can be found in the BRNVCPY Error Message section .

Why Change TCALTTBL?

Occasionally the level of return code may be too high or too low to adhere to your production control standards. If, for example, an individual copy failure for TCE026 VOLUME IN USE at Return Code 8 is not considered a significant error in your shop then you can lower this return code in the TCALTTBL as follows:

```
TCALTMSG TCMSG=TCE026,TCDFLT=8,TCALT=4
```

Must TCALTBL Be Changed?

No, if you are satisfied with the return codes associated with the error messages do nothing. The default alternate return code table, TCALTTBL, is included in the IBM Z Batch Resiliency load library.

What Are the BRNVCPY Default Return Codes?

There are four types of messages that can be generated by the BRNVCPY program: informational, warning, error, and terminating. Informational messages report start, input, and final program results. These require no user response. Warning messages delineate processing concerns such as an input parameter syntax error. This type of message does not prohibit input copy or end program processing but should be looked at. Warning messages get a return code of 4. Error messages include any processing error which causes an input copy request bypass such as an input data set allocation error. BRNVCPY generates a return code 8 and continues processing at the next input record. This type of message does need to be addressed and may require user intervention to resolve. A terminating error message is a serious error causing BRNVCPY processing to end with return code of 12. All BRNVCPY Messages can be found in DD=TRACE and DD=ERRORS. DD=ERRORS contains BRNVCPY processing report. DD=TRACE is a debugging tool for use by Technical Support.

Table 71. Default return codes

Message	Meaning	Return Code
TCI####:	Informational messages	N/A
TCW####:	Warning messages	4
TCE####:	Error messages	8
TCT####:	Terminating error messages	12

How to Change TCALTTBL

BRNVCPY TCALTTBL is made up of two parts: alternate return code table source, and assemble/link JCL.

About this task

To change TCALTTBL, first change the alternate return code table then assemble/linkedit the change into the IBM Z Batch Resiliency execution load library.

© Copyright IBM Corp. 2019, 2020

Note:

If you do not want to change any of the return codes, do nothing. The default table is included in the IBM Z Batch Resiliency load library.

Procedure

1. *hlq*.USERMAC(TCALTTBL) contains the alternate return code table.

   Change any of the return code values by appending ,TCALT=*##* to each TCMSG you want to change. Do not change the TCDFLT value. ## is the numeric return code value BRNVCPY uses if it encounters that TCMSG. The following figure shows a portion of the *hlq*.USERMAC(TCALTTBL) with alternate return codes used.

```
*=========================================================*
* GENERATE ALTERNATE RETURN CODE TABLE FOR BRNVCPY
* NOTE: 1) TCMSG MUST BE IN ALPHABETICAL ORDER
*    2) TO ASSIGN ALTERNATE RETURN CODE TO TCMSG USE TCALT *
*       OPTION, for example TCALTMSG TCMSG=TCT014,TCDFLT=12,TCALT=16
*    3) ASSEMBLE/LINKEDIT TABLE CHANGE VIA HLQ.CNTL(TCALTTBL)
*    4) TCMSG=TCI###  are not recorded in TCALTTBL.
*=========================================================*
         PRINT GEN
TC       TCALTMSG TYPE=INITIAL
         TCALTMSG TCMSG=TCE020,TCDFLT=8
         TCALTMSG TCMSG=TCE021,TCDFLT=8,TCALT=4
         TCALTMSG TCMSG=TCE022,TCDFLT=8
         TCALTMSG TCMSG=TCE023,TCDFLT=8
         TCALTMSG TCMSG=TCE024,TCDFLT=8
         TCALTMSG TCMSG=TCE025,TCDFLT=8
         TCALTMSG TCMSG=TCE026,TCDFLT=8,TCALT=4
             .
             .
             .
         TCALTMSG TCMSG=TCT010,TCDFLT=12,TCALT=8
         TCALTMSG TCMSG=TCT014,TCDFLT=12,TCALT=16
         TCALTMSG TCMSG=TCT017,TCDFLT=12
         TCALTMSG TCMSG=TCT019,TCDFLT=12,TCALT=8
         TCALTMSG TCMSG=TCT020,TCDFLT=12
         TCALTMSG TCMSG=TCT027,TCDFLT=12
         TCALTMSG TCMSG=TCT028,TCDFLT=12
         TCALTMSG TCMSG=TCW007,TCDFLT=4,TCALT=2
         TCALTMSG TCMSG=TCW008,TCDFLT=4
         TCALTMSG TCMSG=TCW010,TCDFLT=4
         TCALTMSG TCMSG=TCW011,TCDFLT=4
             .
             .
             .
         TCALTMSG TCMSG=TCW012,TCDFLT=4,TCALT=2
         TCALTMSG TYPE=FINAL
         END
```

   Figure 149. HLQ.USERMAC(TCALTTBL) with alternate return codes

2. *hlq*.CNTL(TCALTTBL) contains JCL to assemble/linkedit the table.
   a) Add a valid jobcard to this member.
   b) Substitute another DSN for SYSLMOD other than your production IBM Z Batch Resiliency load library.

      Only after the table is assembled and link/edited successfully should you move it into your production IBM Z Batch Resiliency.load library.

   c) Submit *hlq*.CNTL(TCALTTBL).

      Steps C and L should end with return code 0, found in DD=JESMSGLG of your *hlq*.CNTL(TCALTTBL) execution listing.

```
-STEPNAME PROCSTEP RC
-S1       C        00
-S1       L        00
```

   d) Move the new TCALTTBL load module into your IBM Z Batch Resiliency load library.

```
//TCALTTBL JOB
//*-------------------------------------------------------
//* FUNCTION: TCALTTBL ASSEMBLE/LINKEDIT
//* USED TO UPDATE BRNVCPY ALTERNATE RETURN CODE TABLE.
//* 1) UPDATE ALTERNATE RETURN CODE(S) IN
//* HLQ.USERMAC(TCALTTBL)
//* 2) INSTALL UPDATE VIA HLQ.CNTL(TCALTTBL)
//*-------------------------------------------------------
//S1     EXEC HLASMCL
//C.SYSLIB  DD DISP=SHR,DSN=HLQ.USERMAC
//C.SYSIN   DD DISP=SHR,DSN=HLQ.USERMAC(TCALTTBL)
//C.SYSUT1  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//C.SYSLIN  DD DSN=&&OBJ,SPACE=(3040,(40,40),,,ROUND),
//          DISP=(MOD,PASS),UNIT=SYSDA,
//          DCB=(BLKSIZE=3040,LRECL=80,RECFM=FBS,BUFNO=1)
//L.SYSLMOD DD DISP=SHR,DSN=HLQ..SBRNLOAD(TCALTTBL)
//L.SYSUT1  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

Appendix F. SMF.MANx Data Sets and System Logger

IBM Z Batch Resiliency can obtain SMF data from SMF.MANx VSAM data sets or from the System Logger.

Real-Time SMF Extraction

Real-Time SMF collection is accomplished via the IBM Z Batch Resiliency SMF Extractor. The IBM Z Batch Resiliency SMF Extractor reads SMF.MANx files and writes the required records to extract files.

The extracted records are written to an IBM Z Batch Resiliency managed file group called the extract file. The IBM Z Batch Resiliency collection process then uses the extract files to update the History and Inventory databases. Optionally, the SMF Extractor can automatically start the collection process based on an installation-defined interval. As IBM Z Batch Resiliency is shipped, the parameters to control the extractor reside in the CNTL library with the member name SMFP*systemid* where *systemid* corresponds to the SYSTEMID that IBM Z Batch Resiliency was set up with. The COLLECT process initiated by the extractor is COLLECTX in CNTL library.

The SMF Extractor can run as a started task or batch job. An operator interface is provided to control execution. The figure below illustrates the SMF Extractor job flow.

SMF Extractor Job Flow

You can use the SMF extractor to collect SMF records from SMF.MANx VSAM data sets or from the system logger as an alternative to enabling the SMFXU83 user exit.

The following figure illustrates SMF record job flow using SMF.MANx data sets or the system logger.

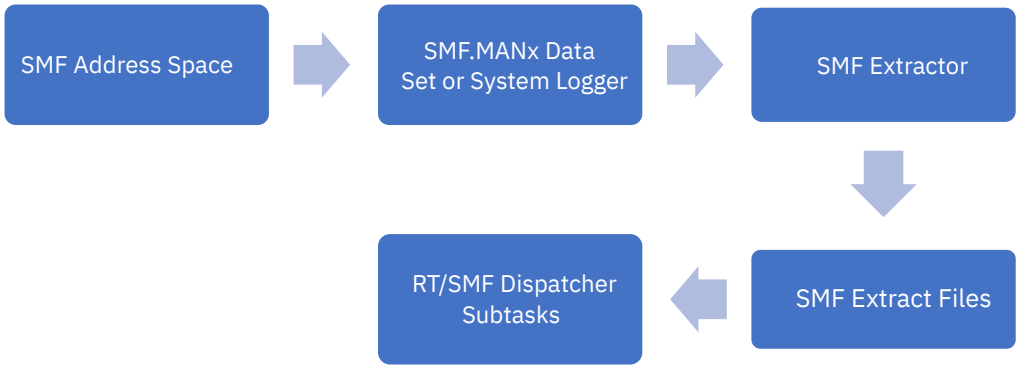


Figure 150. SMF.MANx data sets and system logger

The extracted records are written to an IBM Z Batch Resiliency managed file group called the extract file. The IBM Z Batch Resiliency collection process then uses the extract files to update the History and Inventory databases. Optionally, the SMF Extractor can automatically start the collection process based on an installation-defined interval. The parameters to control the extractor are in the CNTL library with the member name SMFPxxxx where xxxx corresponds to the system ID with which IBM Z Batch Resiliency was set up.

To configure IBM Z Batch Resiliency to use the System Logger, see SMFEXT - LOGR Support.

SMFEXT

Member SMFEXT in the CNTL library is the JCL required to run the SMF Extractor in Batch Mode. The value *systemid* is the system ID where the extractor is run. As IBM Z Batch Resiliency is installed, the system ID is generated from the Modify option of Installation.

© Copyright IBM Corp. 2019, 2020

```
//SMFEXT  EXEC SMFEXTP,
//             COLLECT=COLLECTX,          /* COLLECT JOB MEMBERNAME  */
//             SOUT='*',                  /* SYSOUT CLASS (REPORT)   */
//             SNAP='*',                  /* SYSOUT CLASS (SNAPDUMP) */
//             SYSTEM=xxxx                /* SMF SYSTEM ID           */
```

SMFEXT

Member SMFEXTP is the actual procedure that runs the SMF Extractor. This JCL resides in the CNTL library. It may be placed into a system procedure library to be initiated as a started task.

SMFP *systemid*

The parameters that control the SMF Extractors execution reside in the CNTL library. The *systemid* represents the system ID under which the extractor was setup. Refer to the section CKKSPARM.S Options later in this chapter for details of the parameter values.

CKKSWAIT

This program, when used with the IBM® DUMPXY procedure or the SMFREAD process, prevents it from clearing the SMF.MANx file before the IBM Z Batch Resiliency SMFEXT started task has completed reading it. See Installation and setup for the installation instructions.

Running Multiple SMF Extractors

If you wish to run the SMF extractor on multiple CPUs in your installation there are several considerations that must be taken.

- Each extractor must have an individual Checkpoint file.

  When IBM Z Batch Resiliency is installed, a checkpoint, data set is allocated for the system that the system was installed under. The member CKKIALLO in the CNTL library may be used to define any additional SMF Extractors. The following is a sample of CKKIALLO.

```
//*====================================================================
//* PROGRAM: CKKIALLO >**> VERSION 1.1.0
//*
//* FUNCTION: ALLOCATE A CHECKPOINT DATA SET FOR THE SMF EXTRACTER.
//* *** NOTE: CHANGE XXXX IN DSN TO REFLECT THE CPUID OF
//*           THE SYSTEM TO BE RUN ON AND REPEAT FOR EACH CPU.
//*
//*====================================================================
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1   DD  *
DUMMY
/*
//SYSUT2 DD DSN=HLQ.CHKPOINT.SYSXXXX,
//          UNIT=UNIT,VOL=SER=VOLSER,
//          DISP=(,CATLG),SPACE=(TRK,(1)),
//          DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
```

Figure 151. CKKIALLO example

- Each SMF Extractor must have its own set of parameters.

  The member SMFPxxxx in CNTL library may be used as a template to create member SMFPyyyy in the CNTL library but must be customized to reflect the system ID that it runs under. The parameter OUTDSN= must be modified to reflect its system ID. In systems also running the RTSMF started task, all systems should specify SMFREAD=NO. If, instead of RTSMF, analysis is performed by submitting the COLLECTX job, specify SMFREAD=YES. However, in multi-CPU environments, only one extractor should submit the jobs that initiate the COLLECTX process and therefore only one system should have the parameter SMFREAD=YES. All others should default to SMFREAD=NO.

- A new set of JCL must be created to run the extractor in batch mode.

  In this situation, the system ID must be changed to reflect the second system to run under.
- COLLECTX must be modified to read in the extract files created under the second and subsequent systems.

  Step RTSMF invokes the IBM Z Batch Resiliency program CKKRC001 and resolves the SMF input files created by the extractor.
- ADD/MODIFY the Production JCL that dumps and clears the SYSx.MAN data sets to incorporate the CKKSWAIT. job step.

  See the CKKSWAIT. topic above.

Example: Running Multiple SMF Extractors

IBM Z Batch Resiliency is installed on system MVSA. A second SMF Extractor is required to run on system MVSB which shares DASD with MVSA. The COLLECTX job is used for analysis instead of the RTSMF started task. It is determined that the IBM Z Batch Resiliency COLLECT process continues to run on MVSA.

Procedure

1. Create the Checkpoint file. Run the job CKKIALLO in the CNTL library and modify the following line to reflect MVSB:

   ```
   //SYSUT2 DD DSN=hlq.CHKPOINT.SYSMVSB,..
   ```

2. Create member SMFPMVSB in the CNTL library and copy member SMFPMVSA over. Modify the OUTDSN line to reflect MVSB:

   ```
   OUTDSN=hlq.SMFEXT.SYSMVSB
   ```

   Modify the SMFREAD line so that MVSB does not submit COLLECTX:

   ```
   SMFREAD=NO
   ```

3. If you are running the SMF Extractor as a batch process, create a new member to contain its JCL. Create member SMFEXTB and modify the SYSTEM symbolic to reflect MVSB:

   ```
   SYSTEM=MVSB...
   ```

4. If the SMF Extractor is to run as a started task, issue the command:

   ```
   S SMFEXT,SYSTEM=MVSB
   ```

5. Add the CKKSWAIT step into the production JCL that DUMPS/CLEARS SMF.

   ```
   //STEPXX EXEC PGM=CKKSWAIT.
   //STEPLIB DD DISP=SHR,DSN=hlq.SCKKLOAD.
   ```

SMFEXT - LOGR Support

The SMF extractor supports SMF generation using the System Logger in addition to the standard VSAM SMF generation. The SMF extractor automatically determines if the site is using VSAM or Logger SMF generation.

About this task

This documentation assumes your site is already set up to use the System Logger to record SMF records, and therefore; this documentation only addresses information related to adding support for IBM Z Batch Resiliency itself.

To utilize the Extractor System logger option, follow this check list:

Procedure

1. Specify SMFPRMxx parameters.

   To use SMF logging, specify the following parameters in SMFPRMxx:

   RECORDING(LOGSTREAM)
   Specifies that you want to write SMF data to Log streams rather than SMF data sets. There should be definitions for the rest of your system SMF recording.

   LSNAME

   Specifies a log stream in which you want to write particular SMF record types, specified on the TYPE sub parameter. Add a log stream definition for IBM Z Batch Resiliency to the system PARMLIB member that identifies SMF recording at your site and include the record types required by IBM Z Batch Resiliency. For example:

   ```
   LSNAME(IFASMF.IZBR.&SYSPLEX,
   TYPE(14,15,30,60,61,62,64,65))
   ```

   This definition defines an SMF log stream specifically for IBM Z Batch Resiliency using the &SYSPLEX symbolic. In this way, the same card can be used on each system.

   If the system is SVSCPLEX, the log stream generated is:

   ```
   IFASMF.IZBR.SVSCPLEX
   ```

2. Define an IBM Z Batch Resiliency log stream for each system.

```
//  JOB
//DFHILG1  EXEC PGM=IXCMIAPU
//*********************************************
//* DEFINE DASDONLY LOGSTREAM
//*********************************************
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DATA TYPE(LOGR) REPORT(YES)
/* DEFINE IBM Z Batch Resiliency LOG STREAM    */
  DELETE LOGSTREAM NAME(IFASMF.IZBR.SVSCPLEX)
  DEFINE LOGSTREAM NAME(IFASMF.IZBR.SVSCPLEX)
         DESCRIPTION(SMF_BRNVSTRM)
         DASDONLY(YES)
         MAXBUFSIZE(65532)
         STG_SIZE(1000)
         LS_SIZE(4500)
         LOWOFFLOAD(00)
         HIGHOFFLOAD(80)
```

Figure 152. Sample IBM Z Batch Resiliency log job

This job is just an example to define the IBM Z Batch Resiliency log stream to the system. This should be reviewed with your systems group for definitions that meet your installation's requirements. The

key element for IBM Z Batch Resiliency is the log stream name. In addition, review the utilities, definitions, and parameters in related IBM manuals,such as:

- Z/OS V1.Rx MVS System Management Facilities
- Z/OS V1.Rx MVS Setting Up a SYSPLEX

3. Update SMFPXXXX member in CNTL.

   There are 2 parameters related to LOGR support.

   - The required entry is SMFLOGR. This parameter defines the log stream for this system. Therefore, it must match the Log stream name in the SMF PARMLIB member and the log stream definition job (the job that executes PGM= IXCMIAPU). The log stream name must be, at most, 26 characters.

     Example: SMFLOGR=IFASMF.IZBR.SVSCPLEX

   - The other parameter is LOGRBUFL. The default is 64, representing 64K. This value defines in K(1024) the size of the log buffer. The related log stream definition is MAXBUFSIZE, and this value should be rounded up to the next K(xxxx) from the MAXBUFSIZE definition.

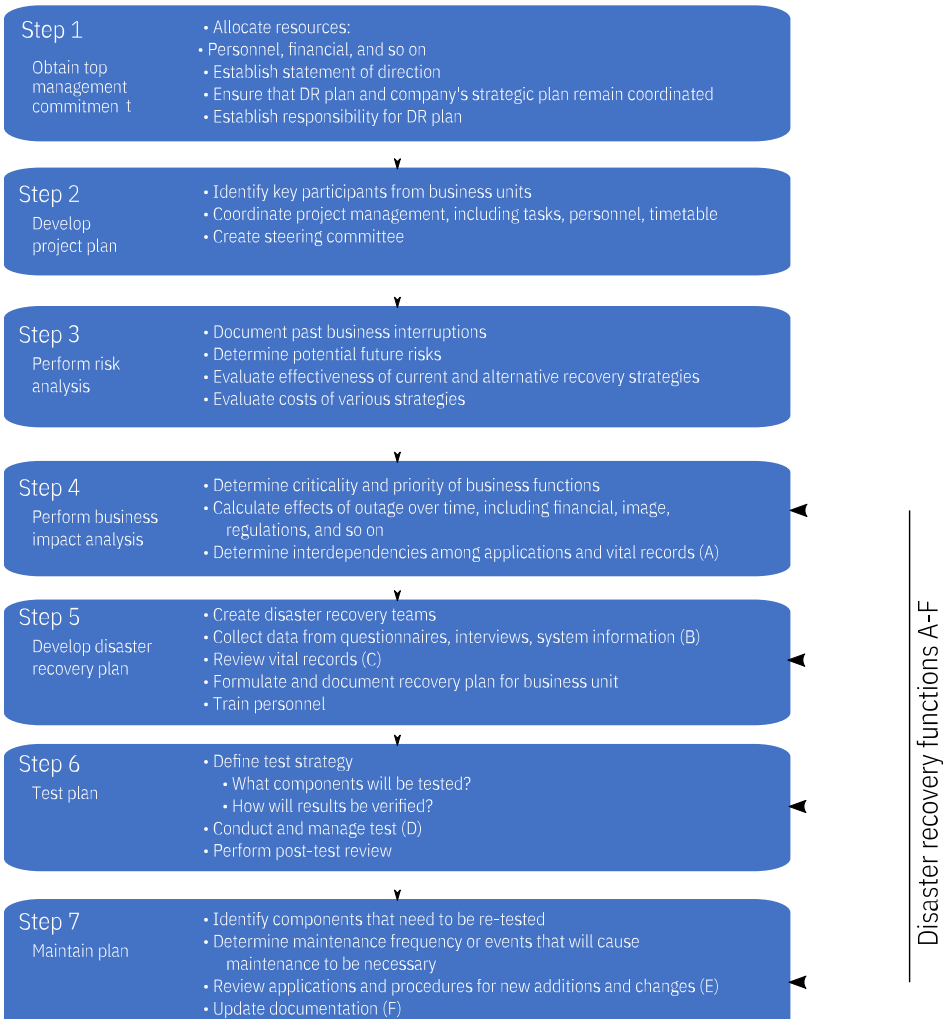
     Example: MAXBUFSIZE(65532) [implies use LOGRBUFL=64]

   Note: The SMFLOGR parameter must be specified in the SMFPxxxx member in the IBM Z Batch Resiliency CNTL library before your system is switched to using the SMF Logger; if not, you receive an IXG231I message in the system log and in the log for the IBM Z Batch Resiliency SMF extractor.

Appendix G. Disaster Recovery Planning Process Overview

To fully understand how IBM Z Batch Resiliency can help automate and implement key aspects of the disaster recovery plan, an overview of the disaster recovery planning process is necessary.

The following figure outlines the general disaster recovery planning process.



Note that in this diagram, steps A-F are directly related to IBM Z Batch Resiliency.

Step 1: Obtain Top Management Commitment

Implementing a disaster recovery plan requires a commitment in terms of time, money, hardware, software, and personnel. Without these resources, an effective disaster recovery plan cannot be realized.

Obtaining a commitment from top management is important in the early stages of a disaster recovery plan to reinforce the image of the plan as an ongoing, essential business function. Management should be involved in coordinating the plan and making certain its direction corresponds with the direction of the business as a whole. A direction statement is often useful during this phase to focus the planning effort on achieving a particular goal. For example: "We will recover service to our customers within 24 hours in the event of a major disaster."

© Copyright IBM Corp. 2019, 2020

Step 2: Develop Project Plan

The project planning phase draws upon Project Management disciplines to organize and coordinate the planning effort.

During project plan development, key members of the business departments should be present to ensure that all business units are adequately represented. This might include members of the Audit, Legal, Facilities Management, Human Resources, Finance, and Information Services departments. Representatives of these departments should be included to help define the project scope, tasks to be performed, and the personnel and time required to perform those tasks.

A steering committee should be created to acquire the necessary resources, confirm that procedures are followed, educate personnel, and ensure that the plan progresses according to the direction statement proposed in Step 1: Obtain Top Management Commitment.

Step 3: Perform Risk Analysis

The risk analysis phase determines which events could seriously disrupt the flow of normal business processing (that is, a disaster).

This phase determines and outlines the actions that must be taken to prevent or reduce the impact of such events. Use past incident reports to show where the business is vulnerable and to evaluate the effectiveness of past recovery procedures. When possible, discuss alternate methods for recovering a function to evaluate the cost effectiveness of the various recovery strategies. Document potential risks and assumptions about the recovery strategies. Assumptions might include such factors as:

- The ability to contact key personnel in the event of a disaster.
- The availability of transportation to the recovery site for responsible personnel.
- The availability of the necessary hardware, vital records, and services at the recovery site within a certain period of time.

If any element cannot be safely assumed, discuss a contingency strategy. Then prepare a document stating that the particular element may not be available under a given set of conditions. Also state that no part of the disaster recovery plan should depend on the availability of that particular element.

Step 4: Perform Business Impact Analysis

After the risks, recovery strategies, and assumptions are identified and documented, analyze their impact on the normal functioning of the business after the occurrence of a particular disaster or set of disasters.

During this phase, identify the company's essential products and services, the resources and information required to support those products and services, interdependencies of a given function, and the time frame in which a function must be restored to guarantee survival of the company. The critical business functions should then be prioritized based on such factors as:

- The financial impact on the corporation.
- The business and customers that will be lost (both present and future).
- The products and services that could be degraded.
- The possible effects on the corporate image.
- The regulatory requirements that exist.
- The competitive advantage this function provides.
- The possible effects on employee morale.
- The effect of this function on another more critical function.

In most data centers, the interdependence of business applications is extremely complex. Files created in one application may be used in another, and updates to vital files may be performed by many applications, some of which may initially seem non-critical.

IBM Z Batch Resiliency is an important auditing tool that identifies such cross-application dependencies by analyzing which applications are creating and using the vital files in your data center. If an application is initially deemed non-critical according to the financial or business-related measures cited above, IBM Z Batch Resiliency can audit the process to ensure that the application does not affect the vital files of any other critical application.

IBM Z Batch Resiliency also provides information to determine the order in which files must be restored to ensure data integrity among the applications that use them. For example, consider a system in which Application A updates the Accounts database on a periodic basis to show which accounts have been paid. Application B then uses this database to generate checks to pay the accounts. During the Business Impact Analysis phase, you determine that Application B is the more critical application because if certain accounts are not paid promptly, these accounts may withhold products or services. The absence of those products and services could prevent your company from operating.

You then design the backup/recovery plan so that:

- The Accounts database is backed up at the end of the Application B cycle.
- Files used by Application A are backed up at the end of the Application A cycle.
- Application B is restored and executed before Application A.

This procedure could cause a serious problem if Application B initially uses a copy of the database showing that all accounts have been paid. Later, Application A would run with an outdated copy of files that post these accounts as being unpaid. The next time Application B executes, the accounts would be paid again, resulting in accounts being paid twice.

Without an automated process in place to identify critical applications and report on critical files that are not backed up in a way that supports your data recovery method, your data center runs the risk of unknowingly excluding critical applications from the recovery process and, thus, compromising the data used by those applications.

Step 5: Develop Disaster Recovery Plan

This phase of plan development should address and document the disaster recovery plan, including the areas described here.

This phase of plan development should address and document the following areas:

- The overall objective of the recovery plan.
- The risks that the recovery plan will address.
- The recovery methods that will be used.
- Which business functions will be recovered.
- The priority in which business functions should be recovered.
- The time frame in which business functions must be recovered.

Formulate the detailed disaster recovery plan from the above information.

During this phase, you should also organize disaster recovery teams for each of the critical business units. The main function of these teams is to develop and coordinate the recovery plan for each individual business unit. The following outlines the responsibilities of those teams.

A. Develop Cooperation and Interest for the Recovery Plan Among Team Members Within the Company

Creation of a team environment greatly increases the chances of an effective recovery.

B. Collect Information from Questionnaires, Interviews, and System Data

The best source of information about a business unit is the people who work with it every day. Create questionnaires that can be used to formulate the detailed plan and prepare personnel for a more in-depth

interview. Interviews are a great source of information, as people often misunderstand a questionnaire or do not have time to fill it out. An interview allows the team leaders to formulate additional questions, explain the recovery plan, obtain user feedback, and build a greater sense of community.

When gathering information, avoid redundancy as much as possible. For example, obtain personnel information from existing documentation in the Human Resources department rather than via a questionnaire. Or, if an onsite recovery procedure is in place for an application, use it as a starting point for developing the offsite recovery plan.

System data is a valuable source of information for determining which jobs and programs comprise an application and which data sets and resources which the application uses. System data also has the advantage of being up-to-date and automatically maintained. You can use IBM Z Batch Resiliency to automatically collect system data on a continuous basis to keep your recovery plan current, avoid data redundancy, and avoid the effort (and possibility of error) involved in collecting data manually.

IBM Z Batch Resiliency stores and maintains data in files that you can easily transport to other platforms or in-house software and documentation files. Additionally, IBM Z Batch Resiliency has powerful reporting capabilities (in SAS and REXX versions), which can create custom reports or tailor existing reports and utilities according to site requirements.

C. Determine Resources Needed for Recovery

Resources include the following:

- Documentation, manuals, contact information
- Hardware, software
- Personnel

From the information obtained in "B. Collect Information from Questionnaires, Interviews, and System Data" on page 259, create a list of resources that must be available at the recovery site in order to meet your recovery objective. This might include recovery documentation, contact information for both employees and vendors, and manuals needed for required software. As a best practice, store this information in files that are backed up and restored as part of your vital records (Step D, following).

You should also compile hardware and software requirements. You can obtain this information from the data collected in "B. Collect Information from Questionnaires, Interviews, and System Data" on page 259. Also prepare a list of required personnel. If any of the people required for recovery have specialized skills, be sure to train other personnel in those areas in case the original person is unavailable.

D. Review Vital Records

Vital records are a critical part of the recovery process. While hardware, manuals, site facilities, and software may be repurchased or leased, vital records cannot. Information is the lifeblood of most modern corporations; without it, no amount of hardware or software will help a company survive a major disaster. The current recovery methods for vital records include:

- Restoring current backups of the original files.
- Retransmitting data from another site.
- Reprocessing old data.
- Using a duplicate copy of files created at an alternate location.

Deciding which recovery method to use is largely governed by:

- The amount of time needed to recover records to a certain point in time.
- The ability to retransmit data.
- Security issues.
- Cost.

To implement an effective vital records program, it is crucial to:

1. Identify and inventory vital records on an ongoing basis.
2. Provide appropriate backup and recovery procedures for those records.
3. Verify that recovery methods are in place and valid at the recovery site.
4. Provide an audit mechanism to ensure that 1, 2, and 3 are current.

Important: The main purpose of IBM Z Batch Resiliency is to effectively and accurately automate the four above-mentioned tasks.

IBM Z Batch Resiliency is a leader in the industry for identification of critical files. By analyzing system data and the way files are used by critical applications, IBM Z Batch Resiliency supplies disaster recovery managers with:

- A critical file list - an automatically maintained list of DASD and tape files which are essential for the normal execution of a critical application.
- A list of DASD and tape exception files (optional).
- An inventory of non-critical files for:
  - VSAM.
  - Files that were allocated before being opened.
  - Any files that are created as part of normal daily processing.

By using IBM Z Batch Resiliency to automate the identification of critical files on a daily basis, disaster recovery personnel are assured that the list of files they are to back up is complete, accurate, and free of errors inherent in manual identification processes.

IBM Z Batch Resiliency also provides a proven method for backing up and restoring these files at the recovery site. Whenever a new critical file is found, or at an interval determined by the user, IBM Z Batch Resiliency can automatically generate backup and restore JCL. By executing the backup JCL as part of a daily cycle (along with the identification process), IBM Z Batch Resiliency provides a level of security that is impossible with manual processes.

Once executed, the backup JCL posts the output of the backup job to a file that IBM Z Batch Resiliency can then use to verify backup synchronization and offsite rotation. By using this approach, a site is able to fully automate all of the above requirements without intruding on the operating system via system hooks or exits, and without the manual effort that would be required of application personnel to generate JCL and update a list of backup data sets.

E. Formulate the Detailed Plan and Identify Responsibilities of Active Participants

At this point in the recovery planning procedure, all resources to be recovered should be identified. Now you must formulate a list of tasks required for the recovery of those resources, along with the personnel responsible for execution of each task. Important items to address on this task list include:

- Contacting hot site and recovery personnel.
- Transportation, lodging, food, supplies, and special considerations (such as prescriptions, day care, and so on) for personnel at recovery site.
- Facilities and resource management at recovery site.
- Coordination of hardware, software, and vital records recovery.
- Telecommunications and support functions.
- Other functions that allow business units to operate in the manner specified by predetermined management goals.

F. Train All Participants

The recovery team must be able to execute their tasks as a coordinated effort. One of the main purposes of a recovery test is to have the team members rehearse their tasks so they eventually become reflexive rather than reactive. Without proper training, however, you can waste valuable test time and hinder the efforts of the disaster recovery plan. Cross-training for all tasks is also essential, as some personnel may

be unavailable when an actual disaster occurs. More importantly, personnel must be relieved after a 10 to 12-hour recovery shift, in order to remain effective.

Step 6: Test the Plan

You must test the disaster recovery plan to reveal weaknesses in the plan and to allow recovery personnel to 'practice'. Often, tests are considered a failure or success based only on the recovery of a set of business units. If a test is staged unrealistically, its success may disguise major faults in the plan that would prevent recovery during an actual disaster.

Devise a written test strategy before actual testing takes place. Items for consideration are:

- What potential disasters are being tested (for example, earthquake, flood, power outage)?
- What business units and components of these business units should be tested?
- Who will conduct and participate in the test?
- How will test results be evaluated?

Coordinate the testing process so that it does not interfere with normal business processing. If certain people are required to execute the business function in normal daily processing, then appoint an alternate for the test, or conduct the test during off hours. If possible, conduct a pre-test walk-through before the test to determine items or procedures that you may have overlooked during the planning process.

After the test, conduct a post-test review. The review should document procedures that were ineffective or could be improved.

Use IBM Z Batch Resiliency before the test to:

- Determine which files need to be backed up and made available at the recovery site for the business units being tested.
- Create a pull list of tapes that are required at the recovery site.
- Determine DASD and tape drive requirements at the recovery site.

Use IBM Z Batch Resiliency during the test to:

- Recover critical DASD and tape files.
- Recover critical application test files to the desired synchronization point.

Step 7: Maintain Plan

After the post-test review, address the portions of the plan that need to be altered and establish a time frame for retesting the components that failed. Maintain the plan to address any future revisions to business components.

If possible, include plan maintenance as part of corporate standards and change control procedures. In order for the plan to remain effective, incorporate any changes to business applications, such as new or modified hardware, software, and vital records, as quickly as possible. Determine the events that will require maintenance (such as new business acquisitions) and anticipate changes to the plan when possible. Keeping your plan updated is critically important to the success of your disaster recovery plan.

IBM Z Batch Resiliency is an invaluable tool for auditing critical applications and maintaining the vital records portion of your disaster recovery plan. Since IBM Z Batch Resiliency uses system data that is available in real time, IBM Z Batch Resiliency can immediately identify new critical files, new jobs that have executed for an existing application, updates to critical files, and if and when a backup has been produced for the critical files.

Additionally, IBM Z Batch Resiliency can automatically create JCL streams required to back up any newly found critical files in order to ensure their availability offsite. IBM Z Batch Resiliency lists and reports may also be downloaded and imported into your personal computer planning software, or used directly on the mainframe, to support and maintain the recovery plan.

Appendix H. IBM Z Batch Resiliency Messages

This section documents the IBM Z Batch Resiliency messages.

Message types are either informational, warning, or error. Message identifiers include a prefix, a number, and either I, W, or E to indicate the severity:

I
Informational message. Describes information or the status for normal conditions and operations. These messages can also provide supporting information accompanying warning or error messages.

W
Warning message. Alerts to a condition that might cause problems in the future. When a warning message is displayed, processing usually continues, but might not complete in a way that is expected. Examine for possible future errors and possible action.

E
Error message. Alerts to a problem that has occurred. Processing cannot continue. Error messages might be accompanied by additional supporting information for diagnosis, problem determination, and action.

ARCSKEXT – ABARS Backup and Restore Interface Messages

This section documents the messages originating from the IBM Z Batch Resiliency ABARS backup and restore processes.

VAP1000I **VAPLDT <timestamp> ARCSKEXT: DSNAME=SAXB.V410.VTC.LICLIB Verification request**

Explanation:
Issued by: ABARS exit. Reports the data set name sent to an exit for verification.

User response:
If ARECOVER processing is held, the *return_code* is X'12' (decimal 18) and the *reason_code* is either 4 or 8. To look up the DFSMShsm message that corresponds to *return_code* 18, locate ARC1618I message and its preceding message ARC1001I in the *IBM z/OS System Messages* manual. IBM Z Batch Resiliency issues message “VAP1618I” on page 263 before the call to DFSMShsm that might result in the ARC1618I message.

VAP1618I **VAPxxx <timestamp> ARECOVER function is disabled or held, ARC1618I**

Explanation:
Issued by: Batch Requestor. Reports that ARECOVER function might be disabled or held, as a result of the HOLD ARECOVER command. If the ARECOVER TSO command were to be issued, ARC1618I message would have been displayed by DFSMShsm if ARECOVER command were to end with RC=18.

User response:
Review the settings and correct. Review the RELEASE ARECOVER command options.

VAP2001E **VAPPSY <timestamp> Error in CMDIN, Data=<data in error>**

Explanation:
Issued by: Batch Requestor. Reports an error in ARECOVER command.

User response:
Correct the error and retry.

VAP2003E **VAPPIN <timestamp> Syntax error in DD:INCLUDE, Data=<data>**

Explanation:
Issued by: Batch Requestor. Reports syntax error in the DD:INCLUDE statement.

User response:
Correct the error and retry.

VAP2004E **DD:INCLUDE data set does not have valid data**

Explanation:
Issued by: Batch Requestor.

User response:
Correct the error and retry.

VAP2005E **VAPxxx <timestamp> *dd* does not exist**

© Copyright IBM Corp. 2019, 2020

Explanation

Issued by: Batch Requestor. Error processing either DD:INCLUDE or DD:CMDIN. The error message is either:

- VAPPIN <timestamp> DD:INCLUDE does not exist
- VAPPSY <timestamp> DD:CMDIN does not exist

User response:
Correct the error and retry.

VAP2006E **VAPxxx <timestamp> *dd* has unsupported dsorg**

Explanation

Issued by: Batch Requestor. Incorrect format for DD:INCLUDE or DD:CMDIN. The error message is either:

- VAPPIN <timestamp> DD:INCLUDE has unsupported dsorg
- VAPPSY <timestamp> DD:CMDIN has unsupported dsorg

User response:
Correct the error and retry.

VAP2007E **VAPxxx <timestamp> SYNAD exit taken for *dd***

Explanation

Issued by: Batch Requestor. I/O error processing for DD:INCLUDE or DD:CMDIN. The error message is either:

- VAPPIN <timestamp> SYNAD exit taken for DD:INCLUDE
- VAPPSY <timestamp> SYNAD exit taken for DD:CMDIN

User response:
Correct the error and retry.

VAP2008E **VAPxxx <timestamp> *dd* does not have LRECL=80**

Explanation

Issued by: Batch Requestor. Incorrect format for DD:INCLUDE or DD:CMDIN. The error message is either:

- VAPPIN <timestamp> DD:INCLUDE does not have LRECL=80
- VAPPSY <timestamp> DD:CMDIN does not have LRECL=80

User response:
Correct the error and retry.

VAP2009E **VAPxxx <timestamp> OPEN error for *dd***

Explanation

Issued by: Batch Requestor. The error message is either:

- VAPPIN <timestamp> OPEN error for DD:INCLUDE
- VAPPSY <timestamp> OPEN error for DD:CMDIN

User response:
Correct the error and retry.

VAP2010E **VAPPSY <timestamp> ARECOVER command in DD:CMDIN is longer than 1024 bytes**

Explanation:
Issued by: Batch Requestor. Limit exceeded for ARECOVER command size.

User response:
Correct the error and retry.

VAP2011W **VAPSAR <timestamp> RECOVERNEWNAMELEVEL() pairs have no matches in DD:INCLUDE**

Explanation:
Issued by: Batch Requestor. After analysis of ARECOVER command it is determined that one of the operands will never be used.

User response:
Execution continues.

VAP2012I **VAPSAR <timestamp> ARECOVER command - RC=0000005Bx**

Explanation:
Issued by: Batch Requestor. Reports *return_code* from ARECOVER command as set by ABARS. Job step *return_code* is set to this value.

User response:
None. Informational message.

VAP2013I **VAPSAR <timestamp> The following command will be sent to DFSMSHSM:**

Explanation:
Issued by: Batch Requestor. Command that is sent to DFSMShsm follows in the multi-line message VAP2014.

User response:
None. Informational message.

VAP2014I **VAPSAR <timestamp> *message_text***

Explanation

Issued by: Batch Requestor. Preceded by message VAP2013. Command to be sent to DFSMShsm in a multi-line message with *message_text* as follows:

- ARECOVER AGGREGATE(aaaaaa) EXECUTE RECOVE

- (CONT) RNEWNAMELEVEL(SAXB,SPJE,OLDL,NEWL) DATAS
- (CONT) ETCONFLICT(RENAMETARGET(AGPJE))

User response:
None. Informational message.

VAP2015I

Explanation:
Issued by: Batch Requestor. Reports the system level token values.

User response:
None. Informational message.

VAP2016I **VAPLDT <timestamp> GetToken: L=00000004x N=VFIASAXBA +0000000**

Explanation:
Issued by: ABARS exit. Reports the system level token name about to be retrieved.

User response:
None. Informational message.

VAP2017I **VAPLDT <timestamp> GotToken: L=00000004x N=VFIASAXBA +0000000 V=00000124 060E3000x**

Explanation:
Issued by: ABARS exit. Reports the system level name token name and its value.

User response:
None. Informational message.

VAP2018I **VAPSAR <timestamp> ARCSKEXT: Activity log DSN=mmmmmmmm.nnnnnnnn.oooooo**

Explanation:
Issued by: Batch Requestor. When ARECOVER command is finished, ABARS exits returns the name of its log data set to a Batch Requestor. This name is reported.

User response:
View the log.

VAP2019E **VAPPIP <timestamp> Error in JCL PARM, Data=<data>**

Explanation:
Issued by: Batch Requestor. Reports a JCL PARM error and displays the data in error.

User response:
Correct the error and retry.

VAP2020I **VAPSAR <timestamp> MSGLVL(3) SVCDUMP(N) TRACE(N) TYPERUN(R)**

Explanation:
Issued by: Batch Requestor. Reports the message level, trace level, SVC dump option, and typerun.

- MSGLVL: from 0 to 9
- SVCDUMP: either Y for YES or N for NO
- TYPERUN: either R for RUN or N for NORUN

User response:
None. Informational message.

VAP2021W **VAPPIP <timestamp> There were no valid keywords in parameter, defaults will be used**

Explanation:
Issued by: Batch Requestor. A JCL parameter is present but there are no valid keywords in it.

User response:
Execution continues. Review the parameter.

VAP2022I **VAPSAR <timestamp> TYPERUN(NORUN) is specified; ARECOVER command is not issued.**

Explanation:
Issued by: Batch Requestor.

User response:
None. Informational message.

VAP2023I **VAPTTR <timestamp> -> Dsname (INC) AG.WORK365.BKUPTBL was accepted for restore by ARCSKEXT**

Explanation:
Issued by: Batch Requestor. Reports that the data set name from the INCLUDE table that was accepted for processing by the ABARS exit.

User response:
None. Informational message.

VAP2024I **VAPAUT <timestamp> Abars/ Assist Version: <ver> Built: <date time>**

Explanation:
Issued by: Batch Requestor and Issued by: ABARS exit. Reports version and build time.

User response:
None. Informational message.

VAP2025I **VAPLDT <timestamp> REQ CTL: nnnnnnnn.mmmmmmmm.oooooooo GOT: pppppppp.qqqqqqqq.rrrrrrrr**

Explanation:
Issued by: ABARS exit. Reports the control data set name requested and allocated.

User response:
None. Informational message.

VAP2026I **VAPTTR <timestamp> => Dsname mmmmmmm.nnnnnnnn.ooooooo in include list but not within the aggregate**

Explanation:
Issued by: Batch Requestor. Reports the data set name from the INCLUDE table that is in INCLUDE list but not within the specified aggregate.

User response:
Revise the INCLUDE list and correct possible mistakes.

VAP2027I **VAPSAR <timestamp> Abars/ Assist Batch Requestor execution complete**

Explanation:
Issued by: Batch Requestor. Final message in the Batch Requestor log - execution completed.

User response:
None. Informational message.

VAP2028I **VAPPAL <timestamp> DD:ABARSLOG is defined as DUMMY; ARECOVER log is not processed**

Explanation
Issued by: Batch Requestor. The name of the ABARS log data set is passed to the Batch Requestor from the ARCSKEXT ABARS exit. If DD:ABARSLOG is allocated to DUMMY in the Batch Requestor JCL, then the ABARS log data set is not processed. Otherwise, the ABARS log data set is allocated and echoed to:

- DD:ABARSLOG is allocated to the job and used if possible.
- SYSOUT=*, if DD:ABARSLOG is not available

User response:
None. Informational message.

VAP2029W **VAPPAL <timestamp> DD:ABARSLOG processing abended <rc>x - <rs>x**

Explanation:
Issued by: Batch Requestor. While processing ABARS log data set, an abend occurred.

User response:
Review the return and reason codes and messages in the joblog. Ensure that DD:ABARSLOG is correctly defined.

VAP2030E **VAPPSY <timestamp> Keyword <data> in DD:CMDIN is not supported**

Explanation:
Issued by: Batch Requestor. Reports that a new for z/OS 1.8 keyword is found and that it is not supported. <data> could be either ONLYDATASET or ONLYDS

User response:
BRNTBARS does not support this keyword. If you wish to use it, use the standard ABARS HSEND ARECOVER command.

VAP2031E **VAPSAR <timestamp> BRNTBARS is not APF-authorized**

Explanation:
Issued by: Batch Requestor. Reports that the BRNTBARS module is loaded from a library that is not APF-authorized.

User response
The BRNTBARS module as well as the ARCSKEXT module must be installed in APF-authorized library.

- If a STEPLIB approach is used to enable BRNTBARS and ARCSKEXT, make sure that this library is either in the APF section of PROGxx SYS1.PARMLIB member, or it has been temporarily authorized by using the SETPROG APF command.
- If the LINKLIST approach is used to enable these modules, ensure that the library where these modules are located is in LINKLIST section of the PROGxx PARMLIB member, and, if LNKAUTH=LNKLST is not specified in IEASYSxx PARMLIB member, added it to the APF section of the PROGxx in PARMLIB.

VAP2032I **VAPSAR <timestamp> ABARSPROC=<name> MaxAbars=<n> ActAbars=<m>**

Explanation:
Issued by: Batch Requestor. Reports the ABARS proc name as defined in the configuration data set, maximum allowed ABARS tasks and the number of active ABARS tasks.

User response:
None. Informational message.

VAP2033I **VAPSAR <timestamp> ARECOVER processing may be held**

Explanation:
Issued by: Batch Requestor. Reports that after analysis of the DFSMShsm control blocks it appears that the ARECOVER processing may be held for all aggregates.

User response:
None. The requested command is still sent to the DFSMShsm, and if HOLD is still in effect, it fails with *return_code* 18 displayed by VAP2012 message.

VAP2034I **VAPSAR <timestamp> HOLD ARECOVER AGNAME(*name*) may be in effect**

Explanation:
Issued by: Batch Requestor. Reports that after examination of the HOLD queue, it appears that the HOLD ARECOVER AGNAME(*name*) may have been issued.

User response:
None. The requested command is still sent to the DFSMShsm, and if HOLD is still in effect, it fails with *return_code* 18 displayed by message VAP2012I.

VAP2035I **VAPSAR <timestamp> HOLD ARECOVER DSNAME(name) may be in effect**

Explanation:
Issued by: Batch Requestor. Reports that after examination of the HOLD queue, it appears that the HOLD ARECOVER DSNAME(*name*) may have been issued.

User response:
None. The requested command is still sent to the DFSMShsm and, if the HOLD is still in effect, it fails with return code 18 displayed by message VAP2012I.

VAP2036I **VAPSAR <timestamp> HOLD ABACKUP(*name*) may be in effect**

Explanation:
Issued by: Batch Requestor. Reports that after examination of the HOLD queue, it appears that HOLD ABACKUP(*name*) may have been issued.

User response:
None. Does not affect ARECOVER processing.

VAP3009I **VAPRAR <timestamp> - (2) Agr = pppppppp.nnnnnnnnnnn Key = qqqqqqqq.nnnnnnnnnnn Ctl = rrrrrrrr.ssssssss.tttttttt**

Explanation:
Issued by: Batch Requestor. Reports the matching aggregate record from DFSMShsm database used by the ARECOVER command as the Batch Requestor determined after parsing the ARECOVER command.

User response:
None. Informational message.

VAP3013I **VAPLDT <timestamp> ARCSKEXT: DSNAME=mmmmmmmm.nnnnnnn.ooooooo *message_text***

Explanation
Issued by: ABARS exit. Reports results of verification - accepted if the data set is in the INCLUDE table or rejected if not. The *message_text* is either:

- Accepted
- Rejected, not in INCLUDE table

User response:
None. Informational message.

VAP3016W **VAPLDT <timestamp> ARCSKEXT did not locate INCLUDE table**

Explanation
Issued by: ABARS exit. There are two possibilities:

- ABARS/Assist ARCSKEXT exit is active but the ARECOVER command is issued NOT by the Batch Requestor. In this case, necessary tables were not built and an entire aggregate is restored.
- Internal error.

User response
- Run the Batch Requestor with appropriate INCLUDE cards.
- Contact Technical Support for instructions. Be ready to rerun the job.

VAP3017W

Explanation:
Issued by: ABARS exit. Internal error - one of the control blocks was not located in the secondary ABARS address space.

User response:
Contact Technical Support for instructions. Be ready to rerun the job.

VAP3100E **VAPEST <t> *** ESTAE routine in control ***

Explanation:
The first in a set of consecutive messages from VAP31000 through to VAP31008.

System action:
Processing is terminated.

User response:
Review the log to determine if ARECOVER command was submitted for execution. Review the details in the set of messages through to VAP31008. Contact Technical Support.

VAP3101E **VAPEST <t> *** SDWA Address - 00005180**

Explanation:
The second in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3102E **VAPEST <t> *** System Abend Code: 000000C4 Reason: 00000011**

Explanation:
The third in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3103E **VAPEST <t> *** Active Load Module name: VAPSAR EPA: 00006A28**

Explanation:
The fourth in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3104E **VAPEST <t> *** PSW @ abend: 078D2000 80009A1C**

Explanation:
The fifth in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3105E **VAPEST <t> *** Which is VAPPSY +0000021Cx**

Explanation:
The sixth in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3106E **VAPEST <t> *** ESTAE routine is finished ***

Explanation:
The seventh in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3107E **VAPEST <t> *** SVC Dump is taken ***

Explanation:
The eighth in a set of consecutive messages from VAP31000 through to VAP31008.

User response:
See message VAP31000.

VAP3108E **VAPEST <t> *** *register_contents***

Explanation

The ninth and last in a set of consecutive messages from VAP31000 through to VAP31008. This is a multi-line message, displaying the *register_contents* over four lines, for example:

- GPR00-03: 0000AB98 00012FF8 0000002E FFFFE94B
- GPR04-07: 0000AAC0 06F00B00 008BEFF8 06F00AF5
- GPR08-11: 00010D64 0000000C 000059E0 0000A800
- GPR12-15: 00009800 000059E0 0000AB98 0000002E

User response:
See message VAP31000.

Backup Grouping Messages

This section documents the Backup Grouping messages.

BGALLO001I **hh:mm:ss DYNAMICALLY ALLOCATING BACKUP TAPE (VOLSER=*volser*)**

Explanation:
Backup Processing stage of Backup Grouping is dynamically allocating Backup Grouping tape to be used for subsequent backup of critical files.

User response:
None. Informational message.

BGALLO002I **hh:mm:ss DYNAMIC ALLOCATION SUCCESSFUL FOR BACKUP TAPE (VOLSER=*volser*)**

Explanation:
Backup Processing stage of Backup Grouping successfully allocated Backup Grouping tape which is used for subsequent backup of critical files.

User response:
None. Informational message.

BGALLO003I **hh:mm:ss DYNAMICALLY UNALLOCATING BACKUP TAPE (VOLSER=*volser*)**

Explanation:
Backup Processing stage of Backup Grouping is dynamically unallocating Backup Grouping tape which was used for backup of critical files.

User response:
None. Informational message.

BGALLO004I **hh:mm:ss DYNAMIC UNALLOCATION SUCCESSFUL FOR BACKUP TAPE (VOLSER=*volser*)**

Explanation:
Backup Processing stage of Backup Grouping successfully unallocated Backup Grouping tape which was used for backup of critical files.

User response:
None. Informational message.

BGALLO005E **hh:mm:ss DYNAMIC ALLOCATION FAILED FOR BACKUP TAPE, *dynamic_allocation_error_message* , INFORMATION CODE nnnn**

Explanation:
Error occurred during Backup processing stage while attempting to dynamically allocate a Backup Grouping tape. The message contains the Dynamic Allocation error code and information code.

System action:
Backup Grouping process cannot continue and ends abnormally.

User response:
Check the *IBM z/OS MVS Programming: Authorized Assembler Services Guide* for a description of Dynamic Allocation error and information codes. Correct the error and rerun the job.

BGALLO006W **hh:mm:ss DYNAMIC UNALLOCATION FAILED FOR BACKUP TAPE, DYNALLOC ERROR CODE nnnn, INFORMATION CODE nnnn**

Explanation:
Error occurred during Backup processing stage while attempting to dynamically unallocate a Backup Grouping tape. The message contains the Dynamic Allocation error code and information code.

System action:
This is only a warning message and Backup Grouping process continues.

User response:
Check the *IBM z/OS MVS Programming: Authorized Assembler Services Guide* for a description of Dynamic Allocation error and information codes.

BGALLO007E **hh:mm:ss *dynamic_allocation_error_message***

Explanation
Error occurred during Backup Grouping Initialization stage while attempting to dynamically allocate a Tape Marker File on a Backup Grouping tape. This message contains the Dynamic Allocation error and information code. The *dynamic_allocation_error_message* is a multi-line message with the following text:

- DYNAMIC ALLOCATION FAILED FOR TAPE MARKER FILE;
- DYNALLOC ERROR CODE nnnn, INFORMATION CODE nnnn
- DSN=*tmf.dsname*

System action:
Backup Grouping initialization process cannot continue and ends abnormally.

User response:
Check the *IBM z/OS MVS Programming: Authorized Assembler Services Guide* for a description of Dynamic Allocation error and information codes. Correct the error and rerun the job.

BGALLO008W **hh:mm:ss *dynamic_unallocation_warning_message***

Explanation
Error occurred during Backup Grouping Initialization stage while attempting to dynamically unallocate a Backup Grouping tape. The message contains the Dynamic Allocation error and information code. This is a warning message and Backup Grouping initialization ends normally. The *dynamic_unallocation_warning_message* is a multi-line message with the following text:

- DYNAMIC UNALLOCATION FAILED AFTER WRITING TAPE MARKER FILE;
- DYNALLOC ERROR CODE nnnn, INFORMATION CODE nnnn
- DSN=*tmf.dsname*

User response:
Check the *IBM z/OS MVS Programming: Authorized Assembler Services Guide* for a description of Dynamic Allocation error and information codes. Correct the error and rerun the job.

BGCRTT003E **hh:mm:ss ERROR ATTEMPTING TO WRITE TAPE MARKER FILE, ERROR CODE *cccc***

Explanation:
Error occurred while Backup Grouping was attempting to write a Tape Marker File (TMF) on Backup Grouping tape. Error code *cccc* further explains the error that occurred. Backup Grouping process cannot continue and ends abnormally.

User response:

Check the error code to determine the problem and correct it.

BGENQF001I hh:mm:ss OBTAINING ENQ FOR BACKUP GROUPING TAPE (VOLSER=*volser*)

Explanation:
Backup Grouping is obtaining an ENQUEUE for tape to prevent another job from using this tape while it is being written to.

User response:
None. Informational message.

BGENQF002I hh:mm:ss ENQ FOR BACKUP GROUPING TAPE WAS SUCCESSFUL (VOLSER=*volser*)

Explanation:
Backup Grouping was successful in obtaining an ENQUEUE for Backup Grouping tape.

User response:
None. Informational message.

BGENQF003I hh:mm:ss RELEASING ENQ FOR BACKUP GROUPING TAPE (VOLSER=*volser*)

Explanation:
Backup Grouping is releasing ENQUEUE for tape because the job is finished writing to the tape.

User response:
None. Informational message.

BGENQF004I hh:mm:ss DEQ FOR BACKUP GROUPING TAPE WAS SUCCESSFUL(VOLSER=*volser*)

Explanation:
Backup Grouping was successful in releasing ENQUEUE for Backup Grouping tape.

User response:
None. Informational message.

BGENQF005E hh:mm:ss ENQ FAILED FOR BACKUP GROUPING TAPE, ERROR CODE *cccc*, (VOLSER=*volser*, QNAME=qname, RNAME=rname)

Explanation:
Error occurred attempting to obtain ENQUEUE for Backup Grouping tape. Error code *cccc* further explains the error that occurred. Backup Grouping process cannot continue and ends abnormally.

User response:
Check the error code to determine the problem, correct it and rerun the job.

BGENQF006W hh:mm:ss DEQ FAILED FOR BACKUP GROUPING TAPE, Error code *cccc*, (VOLSER=*volser*, QNAME=qname, RNAME=rname)

Explanation:
Error occurred attempting to release ENQUEUE for Backup Grouping tape. Error code *cccc* further explains the error that occurred. This is a warning message and Backup Grouping process continues.

User response:
Check the error code to determine the problem and correct it.

BGENQF007E hh:mm:ss ENQ CHECK FAILED FOR BACKUP GROUPING TAPE, ERROR CODE *cccc*, (VOLSER=*volser*, *qname*, *rname*)

Explanation:
Error occurred while Backup Grouping was checking to see if the Backup Grouping tape was currently being used by another job. Error code *cccc* further explains the error that occurred. Backup Grouping process cannot continue and ends abnormally.

User response:
Check the error code to determine the problem, correct it and rerun job.

BGHUNT001I hh:mm:ss SEARCHING FOR NEXT AVAILABLE BACKUP GROUPING TAPE (GROUP=*grpname*)

Explanation:
Backup Processing stage of Backup Grouping is searching for next available Backup Grouping tape to use for subsequent backup of critical files.

User response:
None. Informational message.

BGHUNT002I hh:mm:ss TARGET BACKUP GROUPING TAPE FOUND (VOLSER=*volser*, *fileseq*)

Explanation:
Backup Processing stage of Backup Grouping successfully found a target Backup Grouping tape which is used for subsequent backup of critical files.

User response:
None. Informational message.

BGHUNT003I hh:mm:ss ALL DEFINED TAPES FOR GROUP ARE CURRENTLY IN USE (GROUP=*grpname*)

Explanation:
Backup Processing stage of Backup Grouping found that all the defined Backup Grouping tapes are currently in use by other jobs and are not available for this job. A WTO message BGMAIN012I is also issued by program BGMAIN and the job waits until a Backup Grouping tape becomes available.

User response:
None. Informational message.

BGHUNT004E hh:mm:ss ERROR OCCURRED SEARCHING FOR BACKUP GROUPING TAPE, *cccc*

Explanation:
An error occurred during the Backup Processing stage of Backup Grouping while searching for next available Backup Grouping tape to be used for subsequent backup of critical files. Error code *cccc* further explains the error.

User response:
Check the error code to determine the problem and correct it.

BGHUNT005I hh:mm:ss RESOLVING FSEQ THROUGH TAPE MANAGEMENT API(VOLSER=*volser*, *fileseq*)

Explanation:
Backup Processing stage of Backup Grouping found that the target *fileseq* already exists on the target tape and is therefore resolving new *fileseq* through the appropriate tape management API.

User response:
None. Informational message.

BGHUNT006W hh:mm:ss UNABLE TO DETERMINE TAPE MANAGEMENT SYSTEM FOR FSEQ VERIFICATION

Explanation:
Backup Processing stage of Backup Grouping could not determine the installation's tape management system for FSEQ verification. FSEQ verification does not occur. This is a warning message and Backup Grouping process continues.

User response:
Verify that the TMS= option in VFIPARMS data set is correct.

BGHUNT007W hh:mm:ss UNABLE TO READ VFIPARMS TO GET TAPE MANAGEMENT SYSTEM FOR FSEQ VERIFICATION

Explanation:
Backup Processing stage of Backup Grouping could not read VFIPARMS to determine the Installation's tape management system for FSEQ verification. FSEQ verification does not occur. This is a warning message and Backup Grouping process continues.

User response:
Verify that VFIPARMS DDname is allocated to the job and that the data set to which the DDname points to exists.

BGHUNT008W hh:mm:ss UNABLE READ RMM DATABASE FOR FSEQ VERIFICATION

Explanation:
In an RMM installation, Backup Processing stage of Backup Grouping could not read the RMM database to verify the tape FSEQ. FSEQ verification does not occur. This is a warning message and Backup Grouping process continues.

User response:
Verify that RMMFILE DDname is allocated to the job with the correct RMM database dsname.

BGHUNT010I hh:mm:ss INITIALIZING BACKUP GROUPING NEW DEPTH TAPE (GROUP=grpname, DEPTH=depth)

Explanation:
Backup Processing stage of Backup Grouping needed to initialize a new tape for a new depth level.

User response:
None. Informational message.

BGHUNT011I hh:mm:ss BACKUP GROUPING NEW DEPTH TAPE INITIALIZED SUCCESSFULLY (DSN=*tmf.dsname*, VOLSER=*volser*)

Explanation:
Backup Processing stage of Backup Grouping successfully initialized a new tape for a new depth level.

User response:
None. Informational message.

BGINIT001I hh:mm:ss BACKUP GROUPING TAPE INITIALIZATION BEGINS FOR ALL GROUPS DEFINED IN GRPPARMS

Explanation:
Tape Initialization stage of Backup Grouping is starting tape initialization for all groups that are defined in GRPPARMS member of USERMAC.

User response:
None. Informational message.

BGINIT002I hh:mm:ss BACKUP GROUPING TAPE INITIALIZATION BEGINS FOR GROUP *grpname*

Explanation:
Tape Initialization stage of Backup Grouping is starting tape initialization for a specific group *grpname* as defined in GRPPARMS member of USERMAC..

User response:
None. Informational message.

BGINIT003I **hh:mm:ss BACKUP GROUPING TAPE INITIALIZATION *comptype***

Explanation

comptype is one of the following:

- COMPLETED SUCCESSFULLY
- COMPLETED WITH WARNINGS
- FAILED WITH ERRORS

Specifies the status of how the Backup Grouping Tape Initialization ended.

User response:
If *comptype* is COMPLETED WITH WARNINGS or FAILED WITH ERRORS then check previous warning or error messages and correct the errors.

BGINIT004I **hh:mm:ss BGINIT ENDED WITH RETURN CODE nn**

Explanation:
Specifies the *return_code* that Backup Grouping Tape Initialization program BGINIT ended with.

User response:
None.

BGINIT005I **hh:mm:ss CREATING BACKUP GROUPING TAPE MARKER FILE(S)**

Explanation:
Tape Initialization stage of Backup Grouping is creating a Tape Marker File as the first file of a tape being initialized.

User response:
None.

BGLSTC001E **hh:mm:ss CATALOG SEARCH FAILED FOR TAPE MARKER FILES, *Lstcaterrmsg***

Explanation:
An error occurred while Backup Grouping attempted to search the catalog for Tape Marker Files. *lstcaterrmsg* further explains the error that occurred.

System action:
Backup Grouping process cannot continue and ends abnormally.

User response:
Check *lstcaterrmsg* error codes for further description of error and correct the error.

BGMAIN001I **hh:mm:ss IBM Z Batch Resiliency BACKUP GROUPING PROCESS BEGINS**

Explanation:
Backup Processing stage of Backup Grouping is starting.

User response:
None. Informational message.

BGMAIN002I **hh:mm:ss IBM Z Batch Resiliency BACKUP GROUPING PROCESS *comptype***

Explanation

comptype is one of the following:

- COMPLETED SUCCESSFULLY
- COMPLETED WITH WARNINGS
- FAILED WITH ERRORS

Specifies the status of how Backup Grouping process ended.

User response:
None if COMPLETED SUCCESSFULLY. However, if *comptype* is COMPLETED WITH WARNINGS or FAILED WITH ERRORS, then check previous warning or error messages.

BGMAIN003I **hh:mm:ss BGMAIN ENDED WITH RETURN CODE nn**

Explanation:
Specifies the return code that Backup Grouping Process program BGMAIN ended with.

User response:
None. Informational message.

BGMAIN004I **hh:mm:ss BGMAIN ABENDED WITH USER ABEND nnnn**

Explanation:
Backup Grouping Process program BGMAIN abended with a User Abend code nnnn.

User response:
Check the description of the User Abend code to correct the problem. See Backup Grouping Error Codes.

BGMAIN005I **hh:mm:ss BACKUP GROUPING OPTIONS READ FROM EXEC PARM**

Explanation:
Backup Grouping Process program BGMAIN is reading options from JCL EXEC statement. Message BGMAIN007I lists the options that were read from the EXEC statement.

User response:
None. Informational message.

BGMAIN006I **hh:mm:ss BACKUP GROUPING OPTIONS READ FROM INPUT FILE BRNVBGIN**

Explanation:
Backup Grouping Process program BGMAIN is reading options from input file pointed to by DDname

BRNVBGIN. Message BGMAIN007I lists the options that were read from the input file.

User response:
None. Informational message.

BGMAIN007I hh:mm:ss ==> options

Explanation:
Lists the options that were read by Backup Grouping Process program BGMAIN from the JCL EXEC statement and input file pointed to by DDname BRNVBGIN.

User response:
None. Informational message.

BGMAIN008W hh:mm:ss WARNING: DUPLICATE PARM OPTION FOUND, *dupoption*

Explanation:
Backup Grouping Process program BGMAIN found a duplicate option (*dupoption*)specified. If the option was specified both on the JCL EXEC statement and the input file pointed to by DDname BRNVBGIN, then the value on the JCL EXEC statement is used. This is a warning message and Backup Grouping process continues.

User response:
Verify that the option is not specified more than one time.

BGMAIN009W hh:mm:ss WARNING: INFILE PARM EXCEEDS MAX LENGTH 255

Explanation:
The total length of options specified in the input file pointed to by DDname BRNVBGIN exceeds 255 characters. The total length may not exceed 255 characters and only the first 255 characters are used. This is a warning message and Backup Grouping process continues.

User response:
Verify that the options specified in input file BRNVBGIN do not exceed 255 characters.

BGMAIN010E hh:mm:ss INPUT PARM ERROR – *errortype*

Explanation:
Backup Grouping Process program BGMAIN found an error in the options that were specified on either the JCL EXEC statement or the input file pointed to by DDname BRNVBGIN. *errortype* further explains the option in error. Program BGMAIN abends with User Abend.

User response:
Correct the option which is in error and rerun the job.

BGMAIN011E hh:mm:ss ERROR INVOKING IRXEXEC FOR *rexxrtn*

Explanation:
Backup Grouping Process program BGMAIN received an error attempting to invoke REXX exec routine *rexxrtn*. Backup Grouping process cannot continue and program BGMAIN abends with User Abend.

User response:
Verify that rexxrtn exists in the data set which is pointed to in the JCL by DDname SYSEXEC. If it does exist, check SYSTSPRNT output for any additional REXX messages which may describe the type of error that occurred.

BGMAIN012I hh:mm:ss ALL DEFINED TAPES FOR GROUP ARE CURRENTLY IN USE - WAITING FOR TAPE(S) TO FREE UP

Explanation:
Backup Processing stage of Backup Grouping found that all defined Backup Grouping tapes are currently in use by other jobs and are not available for this job. This WTO message is issued and the job waits until a backup grouping tape becomes available.

User response:
None. Informational message.

BGMAIN013I hh:mm:ss INVOKING PROGRAM *bkuppgm* FOR BACKUP PROCESSING TO TAPE (VOLSER=*volser*, *fileseq*)

Explanation:
Backup Grouping program BGMAIN is invoking data mover program *bkuppgm* to process backup of critical files to the Backup Grouping tape.

User response:
None. Informational message.

BGMAIN014I hh:mm:ss INVOKING PROGRAM *vfibpgm* TO UPDATE IBM Z Batch Resiliency BACKUP TABLE

Explanation:
Backup Grouping program BGMAIN is invoking IBM Z Batch Resiliency program *vfibpgm* to update the IBM Z Batch Resiliency Backup Table for all critical files which were successfully backed up.

User response:
None. Informational message.

BGMAIN015W hh:mm:ss BACKUP TABLE IS NOT BEING UPDATED - NO DATA SETS WERE BACKED UP

Explanation:
Backup Processing stage of Backup Grouping updates the IBM Z Batch Resiliency Backup Table because there were no data sets that were backed up. This is a warning message and Backup Grouping process continues.

User response:
None.

BGMAIN016I hh:mm:ss PROGRAM *pgmname* ENDED WITH RETURN CODE nnnn

Explanation:
Gives the *return_code* of the program that was invoked by Backup Grouping program BGMAIN.

User response:
None. Informational message.

BGMAIN017E hh:mm:ss BACKUP PROCESSING FAILED, CHECK MESSAGES FROM *bkuppgm*

Explanation:
Error occurred in the external data mover program invoked by BGMAIN to process the backup of the critical files. Backup Grouping process cannot continue and ends abnormally.

User response:
See messages written out by the external data mover program such as ADRDSSU for errors.

BGMAIN018E hh:mm:ss ERROR OBTAINING JFCB INFO FOR BACKUP DSN – *errtype*

Explanation:
An error occurred in the Backup Grouping program BGMAIN, attempting to obtain JFCB information on the output backup data set. Backup Grouping process cannot continue and ends abnormally.

User response:
Save output from job and contact Technical Support.

BGMAIN019E hh:mm:ss BRNVCPY ERROR, RELATIVE OUTPUT VOLUME NUMBER MUST BE THE SAME FOR ALL ELEMENTS

Explanation:
For BRNVCPY data mover utility, the relative output volume number must be the same for all the elements being copied.

User response:
Save output from the job and contact Technical Support.

BGUPDT001I hh:mm:ss *updating_TMF_entry*

Explanation
Backup Processing stage of Backup Grouping is updating the Tape Marker File (TMF) catalog entry with *volser* and file sequence information after successful backup of critical files was written. This is a multi-line message with the following *updating_TMF_entry* message text:

- UPDATING TAPE MARKER FILE ENTRY;
- (DSN=*tmf.dsname*, VOLSER=*volser*, FSEQ=*fileseq*)

User response:
None. Informational message.

BGUPDT002I hh:mm:ss TAPE MARKER FILE ENTRY NOT BEING UPDATED - NO DATA SETS WERE BACKED UP

Explanation:
Backup Processing stage of Backup Grouping updates Tape Marker File (TMF) catalog entry with volser and file sequence information because no data sets were backed up by appropriate data mover utility.

User response:
None. Informational message.

BGUPDT003I hh:mm:ss *updating_TMF_successful*

Explanation
Backup Grouping successfully updated the Tape Marker File (TMF) catalog entry with *volser* and file sequence information. This is a multi-line message with the following *updating_TMF_successful* message text:

- TAPE MARKER FILE ENTRY SUCCESSFULLY UPDATED,
- (DSN=*tmf.dsname*, VOLSER=*volser*, FSEQ=*fileseq*)

User response:
None. Informational message.

BGUPDT004E hh:mm:ss *updating_TMF_failed*

Explanation
Backup Grouping failed updating the Tape Marker File (TMF) catalog entry with *volser* and *fileseq*information. This is a multi-line message with the following *updating_TMF_failed* message text:

- TAPE MARKER FILE ENTRY UPDATE FAILED, BGUPDT004E
- (DSN=*tmf.dsname*, VOLSER=*volser*, FSEQ=*fileseq*)

User response:
See the following message BGUPDT005E for additional information on the reason of failure.

BGUPDT005E hh:mm:ss *updderrtype*

Explanation
This message is accompanied by message BGUPDT004E and specifies the reason for the Tape Marker entry update failure. *updderrtype* is one of the following:

- INVALID PARAMETER FOR PROGRAM BGUPDTMF

- NON-ZERO RETURN CODE FROM IDCAMS: IDCAMS RC=nn

User response:
If *upderrtype* is NON-ZERO RETURN CODE FROM IDCAMS, then check the return code from IDCAMS to determine the problem. If *upderrtype* is INVALID PARAMETER FOR PROGRAM BGUPDTMF, then save output from the job and contact Technical Support.

BGUTMF001I hh:mm:ss BACKUP GROUPING UPDATE TAPE MARKER FILE PROGRAM BEGINS

Explanation:
Program to manually update Tape Marker File catalog entry is starting.

User response:
None. Informational message.

BGUTMF002I hh:mm:ss INPUT PARAMETERS

Explanation
The input parameters specified for the program to manually update the Tape Marker File catalog entry. This is a multi-line message displaying the parameters on the next line.

BGUTMF002I hh:mm:ss INPUT PARAMETERS
BGUTMF002I hh:mm:ss parameters

User response:
None. Informational message.

BGUTMF003I hh:mm:ss BACKUP GROUPING UPDATE TAPE MARKER FILE PROGRAM *comptype*

Explanation
Specifies the status of how Backup Grouping Update Tape Marker File program ended. *comptype* is one of the following:

- COMPLETED SUCCESSFULLY
- COMPLETED WITH WARNINGS
- FAILED WITH ERRORS

User response:
None, if COMPLETED SUCCESSFULLY. However, if *comptype* is COMPLETED WITH WARNINGS or FAILED WITH ERRORS, then check the previous warning or error messages.

BGUTMF004I hh:mm:ss PROGRAM BGUTMF ENDED WITH RETURN CODE *return_code*

Explanation:
Specifies the return code from the Backup Grouping Update Tape Marker File program BGUTMF.

User response:
None. Informational message.

BGUTMF005E hh:mm:ss BGUTMF PARAMETER ERROR, MISSING VALUE FOR *option*

Explanation:
Input parameter for program BGUTMF is missing a required value for *option*.

User response:
Specify the missing value for the parameter and rerun the job.

BGUTMF006E hh:mm:ss BGUTMF PARAMETER ERROR, INVALID VALUE FOR option

Explanation:
Input parameter for program BGUTMF has an invalid value for option.

User response:
Correct the invalid value for the parameter and rerun the job.

BGUTMF007E hh:mm:ss TAPE MARKER FILE DATASET NOT CATALOGED; DSN=*tmf.dsname*

Explanation:
Program BGUTMF cannot update the Tape Marker File catalog entry because the Tape Marker File is not cataloged.

User response:
Correct the error and rerun the job.

Backup Grouping Error Codes

Table 72. Backup grouping error codes

Error Code	Reason
0208	Error occurred in program BGCRTTMF attempting to initialize tape for next depth, while searching for next available Backup Grouping tape. See previous messages in job for more information.

Table 72. Backup grouping error codes (continued)

Error Code	Reason
020C	Error occurred attempting to read GRPPARMS while searching for next available Backup Grouping tape.
0210	Backup Grouping found that the specified group was not defined in GRPPARMS while searching for next available Backup Grouping tape.
0214	Error occurred in function BGENQF while searching for next available Backup Grouping tape. The error occurred while checking if tape is currently being used. See previous messages in job for additional information.
0218	Error occurred in function BGENQF while searching for next available Backup Grouping tape. Enqueue failed attempting to initialize tape for next depth.
021C	No Tape Marker Files were found in catalog for specified group while searching for next available Backup Grouping tape. Backup Grouping tapes are not initialized for this group. BGINIT may not have been run yet for this group.
0220	No Tape Marker Files for depth 01 were found in catalog for specified group while searching for next available Backup Grouping tape. Execute BGINIT to re-initialize Backup Grouping tapes for this group.
0224	Error searching catalog while searching for Tape Marker Files while searching for next available Backup Grouping tape. See previous messages for CSI error codes.
0228	Error searching catalog for *volser* information of Tape Marker File for depth > 01 while searching for next available Backup Grouping tape. See previous messages for CSI error codes.
022C	Error while searching for next available Backup Grouping tape. All Tape Marker Files for depth 01 are greater than one day old.
0304	Unable to obtain ENQUEUE for tape resource. Tape is already in use by another job.
0308	Unable to obtain ENQUEUE for tape resource. Non zero return code from ENQ macro.
0504	Dynamic Unallocation failed for Tape Marker File. See associated Dynalloc Error and Information codes.
0508	Dynamic Allocation failed for Tape Marker File. See associated Dynalloc Error and Information codes.
050C	Error occurred reading GRPPARMS while attempting to create a Tape Marker File. GRPPARMS data set may not be allocated in JCL.
0510	Specified group not defined in GRPPARMS while attempting to create a Tape Marker File.
0514	Length of Tape Marker File dsname exceeds 44 characters. Verify the length of group name and High level qualifier in GRPPARMS.
0518	Unable to create Tape Marker File. Data set already exists in catalog.

Table 72. Backup grouping error codes (continued)	
Error Code	**Reason**
051C	Unable to create Tape Marker File. A nonzero return code from EXECIO while writing Tape Marker File.
1250	Missing parm while invoking program BGHUNT. Group name is a required parameter.
1251	Invalid parm while invoking program BGHUNT. Group name is greater than 7 characters.
1260	Logic error in program BGHUNT.
1351	Invalid parm (function code) while invoking function BGENQF.
1352	Invalid parm (*volser*) while invoking function BGENQF.
1353	Invalid parm (*qname*) while invoking function BGENQF.
1450	Missing parm while invoking program BGALLOC.
1451	Invalid/Missing parm1 while invoking program BGALLOC.
1452	Invalid/Missing parm2 while invoking program BGALLOC.
1453	Invalid user parm (Dynalloc option) while invoking program BGALLOC.
1454	Invalid user parm (*dsname*) while invoking program BGALLOC.

BRNYPRNT Messages

This section documents the BRNYPRNT messages. The message identifiers have the following format:

- BRNVCPTnnnI – Informational
- BRNVCPTnnnW – Warning
- BRNVCPTnnnE – Error

BRNVCPT100I REPORT PROCESSING COMPLETE

Explanation:
BRNYPRNT is able to process reports as either single or multi-version. For single version report processing it stores the current report in the report file. This is the informational message given for successful single report processing.

User response:
None. Informational message.

BRNVCPT101I MAIN REPORT COMPLETE

Explanation:
BRNYPRNT multi-version report processing is made up of three steps. First it stores the current report in the report file. Next COPYPRNT copies the report to a new name and archives it to the report file. And last it adds the transaction to the $INDEX report file member. This is the informational message given for successful step one multi-version report processing.

User response:
None. Informational message.

BRNVCPT102I ARCHIVE REPORT COMPLETE

Explanation:
BRNYPRNT multi-version report processing is made up of three steps. First it stores the current report in the report file. Second BRNYPRNT copies the report to a new name and archives it to the report file. And last it adds the transaction to the $INDEX report file member. This is the informational message given for successful step two multi-version report processing.

User response:
None. Informational message.

BRNVCPT103I $INDEX UPDATE COMPLETE

Explanation:
BRNYPRNT multi-version report processing is made up of three steps. First it stores the current report in the report file. Next COPYPRNT copies the report to a new name and archives it to the report file. And third it adds the transaction to the $INDEX report file member. This is the informational message given for successful step three multi-version report processing.

User response:
None. Informational message.

BRNVCPT106I ADD RECORD $INDEX TO REPORTS ($INDEX)

Explanation:
BRNYPRNT multi-version report processing checks for the existence of the $INDEX PDS member. If non-existent, BRNYPRNT creates member $INDEX in the reports PDS and adds a detail line in the $INDEX member documenting its creation. This is an informational message given for successful creation and initialization of PDS member $INDEX.

User response:
None. Informational message.

BRNVCPT107I MEMBER: *memname* DSNAME: *dataset_name*

Explanation:
This informational message is included with an error message to identify the failing data set (member) name.

User response:
Correct error and retry.

BRNVCPT108I <action>

Explanation
This informational message is included with an error message to identify what processing steps were bypassed. Possible actions include:

- ARCHIVE AND $INDEX >>NOT<< PROCESSED
- $INDEX >>NOT<< UPDATED

User response:
Correct error and retry.

BRNVCPT111I REPORTS($INDEX) INUSE BY JOB <jobname> OPERATOR CANCEL REQUESTED

Explanation:
To ensure exclusive use, BRNYPRNT attempts to ENQ the REPORTS($INDEX) member. This informational message is included when the operator has chosen to cancel a retry of the ENQ.

User response:
None. Informational message.

BRNVCPT112I REPORT NAME <reportname> TRUNCATED to 7 CHARACTERS FOR ISPF STATISTICS

Explanation
BRNYPRNT uses the original report name as the ISPF ID in the REPORTS PDS detail lines. This informational message is included only when truncation has occurred. For example:

```
Name     VV.MM Changed       Size Init Mod ID
ABARBKUP 01.04 02/06/20 20:30 358  358   0 ABARBKU
```

User response:
None.

BRNVCPT114W INCOMPATIBLE DCB DD=< ddname>, RETCODE = 0004

Explanation:
An error occurred when BRNYPRNT tried to process the DD= dsname file.

User response:
Check the program joblog to see if there are any errors listed. Check prior processing output to see if there was a problem passing data to BRNYPRNT. If there are any hardware errors, fix them then rerun the job. If the problem persists contact Technical Support.

BRNVCPT100E JMR PROCESSING ERROR

Explanation:
BRNYPRNT uses current jobname and jobtime in a new $INDEX detail record. There was a system error while extracting this information.

User response:
Contact Technical Support.

BRNVCPT101E CVDATE ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
BRNYPRNT formats a ten character date for use in a new $INDEX detail record. There was a system error while processing this information.

User response:
Contact Technical Support.

BRNVCPT102E OUTPUT DYNAMIC ALLOCATION ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
An error occurred when BRNYPRNT tried to dynamically allocate an output reports file data set member.

User response:
Ensure that no other job is using the identified data set (member) and that the data set is not out of space or

control blocks. Rerun the job. If the problem persists, contact Technical Support.

BRNVCPT103E INPUT DYNAMIC ALLOCATION ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
An error occurred when BRNYPRNT tried to dynamically allocate an input reports file data set member.

User response:
Ensure that no other job is using the identified data set (member) and that the data set is not out of space or control blocks. Rerun the job. If the problem persists, contact Technical Support.

BRNVCPT105E GETMAIN ERROR. RETURN CODE X"nnnnnnnn" $INDEX NOT UPDATED

Explanation:
An error occurred when BRNYPRNT tried to obtain main storage space for table processing.

User response:
Ensure that the region parameter on the job card is at least 1M. Fix this and rerun the job. If the problem persists, contact Technical Support.

BRNVCPT106E FREEMAIN ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
An error occurred when BRNYPRNT tried to free the main storage space for table processing.

User response:
Contact Technical Support.

BRNVCPT107E ENQUE RET=USE ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
An error occurred when BRNYPRNT tried to enqueue the reports ($INDEX) member.

User response:
Ensure that no other job is using that resource. Rerun the job. If this problem persists contact Technical Support.

BRNVCPT108E REPORTS FILE ERROR: <function>

Explanation:
An error occurred when BRNYPRNT tried to process the REPORTS file. <function> details what file function failed, that is, OPEN or BLDL.

User response:
Check the program joblog to see if there are any hardware errors. If there are any hardware errors, fix them then rerun the job. If the problem persists, contact Technical Support.

BRNVCPT109E ProdName WTOR ERROR!

Explanation:
An error occurred when BRNYPRNT tried to process a Write To Operator message with Reply (WTOR) message.

User response:
Contact Technical Support.

BRNVCPT111E GQSCAN ERROR. REASON CODE: <nnnn>

Explanation:
BRNYPRNT tries to get exclusive use of the REPORTS PDS via enqueue. If the PDS is already being used, a recheck loop occurs. During the loop the queue is rechecked for REPORTS PDS usage. If it is off usage, the condition codes of the recheck are processed before the enqueue is attempted. This error message occurs if the resource is not in use but not available.

User response:
Ensure that no non-DRVFI program is using the REPORTS PDS. Rerun the program. If the problem persists, contact Technical Support.

BRNVCPT112E CKKRS003 ERROR DDN/MEM = <ddname/membername>, RETURN CODE = <nnnn>

Explanation:
BRNYPRNT uses program CKKRS003 to write new REPORTS PDS members. This message is displayed if an error occurs during this process.

User response:
Contact Technical Support.

Cycle Management Messages

This section documents the Cycle Management messages.

VFI040000I mm/dd/yy hh:mm:ss - OAT OBTAINED FOR: XXXXXXXX AT: XXXXXXXX

Explanation:
This message indicates that the system has detected a record for the application listed and documents the storage where the OAT (Open Application Table) resides.

User response:
None. Informational message.

BRNVB04001E **mm/dd/yy hh:mm:ss - BRNVB040: XXXXXXXX PROGRAM LOAD ERROR**

Explanation:
This message indicates an error loading the identified program.

User response:
Ensure the library containing the listed program is in the STEPLIB and there is sufficient storage to run the job.

VFI040002I **mm/dd/yy hh:mm:ss - XXXXXXXX CYCLE: YYYY.DDD-00 - OPENED ON SYSTEM: XXXX BY: XXXXXXXX yyyy.ddd hh:mm:ss.th ( )**

Explanation:
This message indicates the application opened or closed on the identified system, the cycle and the job that caused the specified action.

User response:
None. Informational message.

BRNVB04003E **mm/dd/yy hh:mm:ss - SORT ERROR, XXXXXXXX PROGRAM TERMINATING**

Explanation:
This message indicates a sort error. Since there are multiple sorts in the process, the active process is identified.

User response:
Review the sort messages for the error, correct the error and rerun the job.

VFI040004I **mm/dd/yy hh:mm:ss - RTSTBLCB OBTAINED - ADDRESS = ' XXXXXXXX SUB-POOL/LENGTH = XXXXXXXX**

Explanation:
This message reports the acquired control block and the address.

User response:
None. Informational message.

BRNVB04005E **mm/dd/yy hh:mm:ss - INPUT PARM ERROR**

Explanation:
This message is listed when an error is found in the input parameters.

User response:
Review valid input parameters, correct error and rerun job.

BRNVB04006E **mm/dd/yy hh:mm:ss - CRITICAL FILE ERROR**

Explanation:
This message indicates an error building the internal Critical File Table.

User response:
Ensure that it is a valid Critical file, and rerun the job.

BRNVB04007E **mm/dd/yy hh:mm:ss - ABENDING XXXXXXXXXXXXXXXXXXX**

Explanation:
This message is issued when a severe error occurs. The reason is indicated in the message.

User response:
Correct the error and rerun the job.

In-Flight Processing Messages

The following messages may be issued by IBM Z Batch Resiliency In-Flight processing.

BRNVC02601I **mm/dd/yy hh:mm:ss - IN FLIGHT SUBTASK: XXXXXXXX WAITING ON SUBTASK: XXXXXXXX ODTCKPT MUST COMPLETE**

Explanation:
This informational message is issued at RTSMF shut down, indicating the ODTCKPT is still in progress. BRNVC026 must wait for the checkpoint on the OAT (DD ODTCKPT) to complete.

User response:
None. Informational message.

BRNVC02602I **mm/dd/yy hh:mm:ss - BRNVC026: RTRTIPRM ERROR, RTI$ COLLECTION DISABLED**

Explanation:
This message indicates RTI$ collection is disabled because of the parameter member USERMAC.(RTRTIPRM) is not defined or the content is invalid. This member identifies the high level qualifiers to use if RTIJB is defined in VFIPARMS. The job identified by RTIJB acts as a trigger to indicate a condition where the OAT in flight jobs and data sets should be captured. When this occurs the high level qualifier defined in USERMAC.(RTRTIPRM) is used to name to RTI$ data set. The captured data set can be used for DD ODTCKPT in CNTL(INFLTODT) to report in-flight data.

User response:

If you wish to use the trigger job function, define the job in USERMAC.(VFIPARMS), value RTIJB, and define the high level qualifier you need in USERMAC.(RTRTIPRM).

BRNVC02603I mm/dd/yy hh:mm:ss - BRNVC026: ALLOCATING RTI$ FILE:

Explanation:
This message documents the RTI$ file being created.

User response:
None. Informational message.

BRNVC02604E mm/dd/yy hh:mm:ss - BRNVC026: DYNAMIC ALLOCATION ERROR - R15(XXXXXXXX) S99ERROR(XXXX) S99INFO(XXXX) RTI$SNAP TERMINATED

Explanation:
This message indicates an error attempting to allocate the RTI$ snap file.

User response:
Evaluate the return and error codes in the IBM DYNALLOC documentation to identify and resolve the error.

VFI026005I mm/dd/yy hh:mm:ss - BRNVC026: IN RTI$SNAP STATS - APL ENTRIES 9999999 OJTE ENTRIES: 999999 OJDE ENTRIES: 999999

Explanation:
This message indicates the amount of data written for the RTI$ snap data set.

User response:
None. Informational message.

BRNVC02606I mm/dd/yy hh:mm:ss - INTERVAL TERM. - TASK XXXXXXXX PGM = BRNVC026 R/T = NNN COUNTS = NNN,NNN,NNN

Explanation:
This message is issued at a RTSMF check point. It indicates the SMF record types and number processed during the recent RTSMF period.

User response:
None. Informational message.

BRNVC02607E mm/dd/yy hh:mm:ss - BRNVC026: RTI$SNAP OPEN ERROR, COLLECTION DISABLED

Explanation:
This message indicates there was an error opening the RTI$ data set. Check the JES log for additional information.

User response:
Resolve the open error.

VFI045000I mm/dd/yy hh:mm:ss - OAT OBTAINED FOR: XXXXXXXX AT: XXXXXXXX

Explanation:
This message indicates that an OAT was obtained while processing the HISTORY file to identify in-flight jobs and data sets. The application and location are identified.

User response:
None. Informational message.

BRNVB04501E mm/dd/yy hh:mm:ss -BRNVB045: XXXXXXXX PROGRAM LOAD ERROR

Explanation:
This message indicates an error loading the identified program.

User response:
Determine why there was an error by checking the JES log for additional information. The most common error is the program does not exist in the STEPLIB concatenation for the job (Abend 806).

VFI045002I mm/dd/yy hh:mm:ss - XXXXXXXX CYCLE: YYYY.CDDD-00 - OPENED ON SYSTEM: XXXX BY: XXXXXXXX yyyy.ddd hh:mm:ss.th ( )

Explanation:
This message indicates that an application opened during the re-processing of the HISTORY file. The cycle, application, reason, and date/time are documented.

User response:
None. Informational message.

BRNVB04504I mm/dd/yy hh:mm:ss - RTSTBLCB OBTAINED - ADDRESS = XXXXXXXX, SUB-POOL/LENGTH =XXXXXXXX

Explanation:
This message indicates the RTSMF table control block was obtained.

User response:
None. Informational message.

BRNVB04505E mm/dd/yy hh:mm:ss - INPUT PARM ERROR

Explanation:
The parameters supplied to BRNVB045 are in error.

User response:
Correct the input, and rerun the job.

BRNVB04507E mm/dd/yy hh:mm:ss - ABENDING

Explanation:

This message indicates BRNVB045 is abending, and also the reason for the error.

User response:
Correct the error and rerun.

BRNVB04507I mm/dd/yy hh:mm:ss - BRNVB045: IN FLIGHT STATS - APL ENTRIES: 999999, OJTE ENTRIES: 999999 OJDE ENTRIES: 999999

Explanation:
This message documents the statistics related to in flight data.

User response:
None. Informational message.

INVMAINT Messages

There are three types of messages that can be generated by the INVMAINT program:

- Informational - These messages report input, delete, and final program results. These require no user response.
- Warning - These messages delineate processing concerns such as input edit errors. This type of message does not end program processing but generates a return code of 4 which might require user intervention to resolve.
- Error - An error message is a serious error that causes INVMAINT processing to end with a return code of 8.

The message identifiers have the following format:

Table 73. Message ID format

Message	Meaning	Return Code
IMNTInnn	Informational	N/A
IMNTWnnn	Warning	4
IMNTEnnn	Error	8

IMNTI001 PROCESSING BEGINS: mm/dd/yyyy hh:mm:ss

Explanation:
Signals the start of INVMAINT processing.

User response:
None. Informational message.

IMNTI002 EDITING INPUT: <display of input record CL80 >

Explanation:
Displays DD=SYSIN input records.

User response:
None. Informational message.

IMNTI003 REQUEST ACCEPTED: <display of record built for INVENTRY lookup>

Explanation:
Displays the record INVMAINT program built from the DD=SYSIN records. DSN, APPLID, and DSNAME fields may contain masking.

User response:
None. Informational message.

IMNTI004 REMOVE REQUEST <display of record built for INVENTRY delete search>

Explanation:
Displays the remove record INVMAINT program built from the DD=SYSIN records. DSN, APPLID, and DSNAME fields may contain masking.

User response:
None. Informational message.

IMNTI005 INVENTRY/FIXIST DELETE <show INVENTRY record that was deleted>

Explanation:
Displays the record deleted from the INVENTRY file. If masking has been used, multiple messages may be displayed for each matching INVENTRY record deleted.

User response:
None. Informational message.

IMNTI006 NO INVENTRY ELIGIBLE FOR DELETE

Explanation:

Reports that no INVENTRY records were eligible for delete. Return code = 4.

User response:
None. Informational message.

IMNTI007 NO INPUT ELIGIBLE FOR PROCESSING

Explanation:
Reports that no DD=SYSIN input records were valid for INVMAINT processing. Since no input was valid, no INVENTRY records were deleted. This message follows IMNTW03# messages and has a return code = 4.

User response:
Correct the cause of the IMNTW03# messages and rerun INVMAINT.

IMNTI008 INVENTRY/SIMULATE DELETE/ UPDATE B/C COUNT

Explanation:
INVMAINT issues the number of records processed message at program end. The message states that processing was in LIVE or SIMULATE mode and if the records processed were deleted or updated. A count is provided for the number of both "B", base, and "C", component, records processed.

User response:
None. Informational message.

IMNTI010 TOTAL START {B|C} COUNT: ##########

Explanation

Displays total number of INVENTRY Base "B" and Component "C" records at INVMAINT start. The start totals displayed depend on the INVMAINT request processed.

```
REQUEST  OPTION  TOTAL
-------  ------  -----
REMOVE   DSN     B,C
REMOVE   DSB     B
REMOVE   DSNAME  C
CATUPDT  DSNAME  C
```

Example: `REMOVE,DSN=...`

```
IMNTI010  TOTAL    START   BCOUNT: 0000000999
IMNTI010  TOTAL    START   CCOUNT: 0000001036
IMNTI010                           ------------
IMNTI010                           000000002035
```

Example: `REMOVE,DSB=...`

```
IMNTI010  TOTAL    START   BCOUNT: 0000000999
IMNTI010                           ------------
IMNTI010                           000000000999
```

Example:
`CATUPDT,DSNAME=...`
`REMOVE,DSNAME=...`

```
IMNTI010  TOTAL    START   CCOUNT: 0000001036
IMNTI010                           ------------
IMNTI010                           000000001036
```

Example:
`REMOVE,DSB=...`
`REMOVE,DSNAME=...`

```
IMNTI010  TOTAL    START   BCOUNT: 0000000999
IMNTI010  TOTAL    START   CCOUNT: 0000001036
IMNTI010                           ------------
IMNTI010                           000000002035
```

User response:
None. Informational message.

IMNTI011 {TOTAL|SIMULATE} UPDATE CCOUNT: ##########

Explanation

Displays total number of INVENTRY Component "C" records updated by CATUPDT request. CATUPDT is a DSNAME= request which processes component "C" records only. The CATUPDT,SIMULATE request does not change the INVENTRY file but reports on what would happen for an actual (non-simulated) CATUPDT request.

Example: `CATUPDT,DSNAME=...`

```
IMNTI011  TOTAL    UPDATE  CCOUNT: 0000000079
IMNTI011                           ------------
IMNTI011                           000000000079
```

Example: `CATUPDT,SIMULATE,DSNAME=...`

```
IMNTI011  SIMULATE UPDATE  CCOUNT: 0000000079
IMNTI011                           ------------
IMNTI011                           000000000079
```

User response:
None. Informational message.

IMNTI012 {TOTAL|SIMULATE} DELETE {B|C} COUNT: ##########

Explanation

Displays total number of INVENTRY Base "B" and Component "C" records removed by the INVMAINT REMOVE requests. Totals depend on the INVMAINT REMOVE NAME option processed. The REMOVE,SIMULATE request does not change the

INVENTRY file but reports on what would happen for an actual (non-simulated) REMOVE request.

```
REQUEST  OPTION  TOTAL
-------  ------  -----
REMOVE   DSN     B,C
REMOVE   DSB     B
REMOVE   DSNAME  C
```

Example: REMOVE,DSN=...

```
IMNTI012  TOTAL    DELETE  BCOUNT: 0000000215
IMNTI012  TOTAL    DELETE  CCOUNT: 0000000216
IMNTI012                           ------------
IMNTI012                           000000000431
```

Example: REMOVE,SIMULATE,DSB=...

```
IMNTI012  SIMULATE DELETE  BCOUNT: 0000000215
IMNTI012                           ------------
IMNTI012                           000000000215
```

Example: REMOVE,DSNAME=...

```
IMNTI012  TOTAL    DELETE  CCOUNT: 0000000216
IMNTI012                           ------------
IMNTI012                           000000000216
```

User response:
None. Informational message.

IMNTI013 {TOTAL|SIMULATE} FINISH {B|C} COUNT: ##########

Explanation

Displays total number of INVENTRY Base "B" and Component "C" records at INVMAINT end. Final totals depend only on the INVMAINT REMOVE requests processed. CATUPDT requests update INVENTRY records only and so do not change ending totals. The SIMULATE option on a request does not change the INVENTRY file but reports on what would happen for an actual (non-simulated) request.

```
REQUEST  OPTION  TOTAL
-------  ------  -----
REMOVE   DSN     B,C
REMOVE   DSB     B
REMOVE   DSNAME  C
CATUPDT  DSNAME  C
```

Example: REMOVE,SIMULATE,DSN=...

```
IMNTI013  SIMULATE FINISH  BCOUNT: 0000000784
IMNTI013  SIMULATE FINISH  CCOUNT: 0000000820
IMNTI013                           ------------
IMNTI013                           000000001604
```

Example: REMOVE,DSB=...

```
IMNTI013  TOTAL    FINISH  BCOUNT: 0000000784
IMNTI013                           ------------
IMNTI013                           000000000784
```

Example: REMOVE,DSNAME=...

```
IMNTI013  TOTAL    FINISH  CCOUNT: 0000000820
IMNTI013                           ------------
IMNTI013                           000000000820
```

Example:
REMOVE,SIMULATE,DSN=...
REMOVE,SIMULATE,DSB=...

```
IMNTI013  SIMULATE FINISH  BCOUNT: 0000000784
IMNTI013  SIMULATE FINISH  CCOUNT: 0000000820
IMNTI013                           ------------
IMNTI013                           000000001604
```

User response:
None. Informational message.

IMNTI999 PROCESS COMPLETE – RETURN CODE: {final return code}

Explanation:
Signals the end of INVMAINT processing and displays the final (highest) return code.

User response:
None. Informational message.

IMNTW007 DD=SYSIN INPUT FILE IS EMPTY

Explanation:
Reports that there were no records in the DD=SYSIN input file for INVMAINT to process.

System action:
INVMAINT ends with return code = 4 with no INVENTRY records processed.

User response:
Correct the cause of the empty DD=SYSIN and rerun INVMAINT.

IMNTW030 REMOVE BYPASSED – INVALID STATEMENT

Explanation:
INVMAINT has detected an invalid statement in the current DD=SYSIN input record being edited. The current DD=SYSIN record can be seen in informational message IMNTI002.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW031 REMOVE BYPASSED – DELAGE BLANK

Explanation:
INVMAINT has detected an error in the DELAGE= argument as displayed in message IMNTI002. DELAGE=#### must be between 1 and 9999. Omit the parameter if not using it.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW032 REMOVE BYPASSED – DELAGE NOT NUMERIC

Explanation:
INVMAINT has detected an error in the DELAGE= argument as displayed in the message IMNTI002. DELAGE=#### must be between 1 and 9999. Omit the parameter if not using it.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW033 REMOVE BYPASSED – LRAGE NOT NUMERIC

Explanation:
INVMAINT has detected an error in the LRAGE= argument as displayed in the message IMNTI002. LRAGE=#### must be between 1 and 9999. Omit the parameter if not using it.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW034 REMOVE BYPASSED – APPLID > CL8

Explanation:
INVMAINT has detected an error in the APPLID= argument as displayed in the message IMNTI002. APPLID=@@@ is CL8 and can be a complete applid from 1 to 8 characters. Masking is permitted.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW035 REMOVE BYPASSED – DSN > CL44

Explanation:
INVMAINT has detected an error in the DSN= argument as displayed in the message IMNTI002. DSN=!!! is CL44 and can be a complete DSN from 1 to 44 characters or a filter. INVMAINT accepts a filter as an argument for the parameter and validates it according to DFSMS rules.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW036 REMOVE BYPASSED – DSNAME > CL44

Explanation:
INVMAINT has detected an error in the DSNAME= argument as displayed in the message IMNTI002. DSNAME=&&& is CL44 and can be a complete DSNAME from 1 to 44 characters or a mask. INVMAINT accepts a filter as an argument for the parameter and validates it according to DFSMS rules.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW037 REMOVE BYPASSED – DSN OR DSNAME NOT BOTH

Explanation:
INVMAINT has detected an error in the REMOVE statement as displayed in the message IMNTI002. One of either DSN=!!! or DSNAME=&&& is permitted, but not both.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW038 REMOVE BYPASSED – BRNXLIST ERROR

Explanation:
INVMAINT has detected an error in the BRNXLIST option as displayed in the message IMNTI002. Option BRNXLIST can be followed by a comma if more options follow or blanks if no more options follow. Make sure the input line following the last option has all blanks in it. LRECL=80.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:

Correct the invalid statement and rerun INVMAINT.

IMNTW039 REMOVE BYPASSED – MSG OPTION NEEDS Y or N

Explanation:
INVMAINT has detected an error in the MSG= argument as displayed in the message IMNTI002. Valid response is Y or N. Omitting the MSG= option defaults to MSG=Y.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW040 REMOVE BYPASSED – DSN or DSNAME REQUIRED

Explanation:
INVMAINT has detected an error in the REMOVE/ CATUPDT statement as displayed in the message IMNTI002. One of DSN=!!! or DSNAME=&&& is required. INVMAINT accepts a filter as an argument for the parameter and validatews it according to DFSMS rules.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW041 REMOVE BYPASSED – SIMULATE ERROR

Explanation:
INVMAINT has detected an error in the REMOVE / CATUPDT statement as displayed in the message IMNTI002. The correct syntax is the word SIMULATE and is an option within a REMOVE or CATUPDT DD=SYSIN input record.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW042 REMOVE BYPASSED – CATUPDT ERROR

Explanation:
INVMAINT has detected an error in the CATUPDT statement as displayed in the message IMNTI002. The correct syntax is the word CATUPDT and is the first word on a CATUPDT DD=SYSIN input record. Valid CATUPDT statement minimally needs a DSNAME= option.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW043 REMOVE BYPASSED – CATUPDT REQUIRES DSNAME=

Explanation:
INVMAINT has detected an error in the CATUPDT statement as displayed in the message IMNTI002. The correct syntax is the word CATUPDT and is the first word on a CATUPDT DD=SYSIN input record. Valid CATUPDT statement minimally needs a DSNAME= option. INVMAINT accepts a filter as an argument for the parameter and validates it according to DFSMS rules. DSNAME=** is acceptable.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW044 REMOVE BYPASSED – CATUPDT / BRNXLIST INVALID

Explanation:
INVMAINT has detected an error in the CATUPDT statement as displayed in the message IMNTI002. BRNXLIST is not a valid option for CATUPDT DD=SYSIN input record. BRNXLIST is only valid for a REMOVE DD=SYSIN input record.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW045 REMOVE BYPASSED – KEYWORD MISSING / INVALID

Explanation:
INVMAINT has detected an error in the REMOVE or CATUPDT statement as displayed in the message IMNTI002. This message usually means that the REMOVE or CATUPDT keyword is missing from the DD=SYSIN input record.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW046 REMOVE BYPASSED – DELAGE REQUIRES DSNAME=

Explanation:

INVMAINT has detected an error in the REMOVE statement as displayed in the message IMNTI002. DSNAME is required when DELAGE is used.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW047 REQUEST BYPASSED - MIX SIMULATE + NON, INVALID

Explanation:
INVMAINT has detected an error in the REMOVE statement as displayed in the message IMNTI002. DSNAME is required when DELAGE is used.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Correct the invalid statement and rerun INVMAINT.

IMNTW050 NEGFILE PROCESS – *message_text*

Explanation

Multi-line message with the following *message_text*:

NO NEGFILE DD IN JCL OR USING DD DUMMY
MUST USE SLOWER CATALOG SEARCH
FOR BETTER PERFORMANCE
NEED TO CREATE NEGFILE – SEE MANUAL

During REMOVE BRNXLIST or CATUPDT processing, INVMAINT has detected that the DD statement for NEGFILE has not been provided in the execution JCL.

System action:
Processing continues but catalog searching is done via the IBM LOCATE processing which is significantly slower than the DRVFI NEGFILE processing.

User response:
For improved performance in INVMAINT processing, see Chapter 10, “IBM Z Batch Resiliency Utilities,” on page 159.

IMNTW054 INVENTRY PPROCESS - NEGFILE EMPTY, IMPLIES ALL INVRECS CATLG

Explanation:
While processing a REMOVE BRNXLIST request, INVMAINT has detected that the NEGFILE is empty. This file contains a list of all data set names referred to in the INVENTRY file that are uncatalogued. NEGFILE EMPTY implies that all INVENTRY records are valid and none are in need of BRNXLIST delete.

System action:
INVMAINT bypasses the current DD=SYSIN input record and generates return code = 4.

User response:
Make sure that the NEGFILE is actually empty and was not incorrectly created. For more information, see Chapter 10, “IBM Z Batch Resiliency Utilities,” on page 159.

IMNTE010 PROCESSING ERROR – JMRDATE NOT FOUND

Explanation:
INVMAINT has detected a systems error while processing the current date.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE011 PROCESSING ERROR – DD=SYSIN OPEN ERROR

Explanation:
INVMAINT has detected a systems error while processing the DD=SYSIN open.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=SYSIN. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE014 PROCESSING ERROR – C21VS INIT WORKAREA ERROR

Explanation:
INVMAINT has detected a systems error while processing the VSAM work area initialization.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE015 PROCESSING ERROR – SORT ERROR

Explanation:
INVMAINT has detected a systems error while sorting the QIOAREA.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE020 PROCESSING ERROR – QIOAREA #0/1 CLOSE ERROR

Explanation:
INVMAINT has detected a systems error while opening the QIOAREA #0/1.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE021 INVENTRY PROCESS – OPEN (UPDATE) ERROR

Explanation:
INVMAINT has detected a system error while opening the INVENTRY file.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC messages associated with the INVENTRY file. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE022 INVENTRY PROCESS - CLOSE ERROR

Explanation:
INVMAINT has detected a systems error while closing DD=INVENTRY.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=INVENTRY. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE024 INVENTRY PROCESS - POINT ERROR

Explanation:
INVMAINT has detected a systems error while processing a VSAM point command against DD=INVENTRY.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=INVENTRY. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE025 INVENTRY PROCESS – GETNEXT ERROR

Explanation:
INVMAINT has detected a systems error while processing a VSAM GETNEXT command against DD=INVENTRY.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=INVENTRY. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE026 INVENTRY PROCESS – DELETE ERROR

Explanation:
INVMAINT has detected a systems error while processing a VSAM delete command against DD=INVENTRY.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=INVENTRY. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE027 INVENTRY PROCESS – PUT ERROR

Explanation:
INVMAINT has detected a systems error while processing a VSAM put command against DD=INVENTRY.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=INVENTRY. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE028 INVENTRY PROCESS – OPEN (INPUT) ERROR

Explanation:
INVMAINT has detected a system error while opening DD=INVENTRY.

System action:
INVMAINT terminates processing and generates a return code of 8.

User response:
Check the job log to see if there are any IEC messages associated with the INVENTRY file. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE029 NEGFILE PROCESS – GETKEY ERROR

Explanation:
INVMAINT has detected a systems error while processing a VSAM GETKEY command against DD=NEGFILE.

System action:
INVMAINT terminates processing and reports the return code.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=INVENTRY. If so, correct the error and rerun INVMAINT. If the problem reoccurs, contact Technical Support.

IMNTE059 C21VS R0 DISPLAY - R14(xxxxxxxx) R15(xxxxxxxx) R0(xxxxxxxx)

Explanation:
This error message provides additional supporting information for message IMNTE026.

System action:
INVMAINT terminates processing and generates an abend 3203.

User response:
Contact Technical Support with your execution listing and dump.

BRNVB05901I mm/dd/yy hh:mm:ss - BRNVB059: BATCH INVMAINT STARTED

Explanation:
This message indicates the start of the INVMAINT process.

User response:
None.

BRNVB05902I mm/dd/yy hh:mm:ss - BRNVB059: BATCH INVMAINT COMPLETE

Explanation:
This message indicates the end of the INVMAINT process.

User response:
None.

BRNVL02001I 08/31/13 12:10:56 - INVMAINT TABLE LOADER STARTED

Explanation:
Issued by BRNVL020 to indicate the INVMAINT table load process started.

User response:
None.

BRNVL02002I mm/dd/yy hh:mm:ss - INVMAINT LOADED - 2 INPUT, 2 OUTPUT, 0 ERROR

Explanation:
Issued by BRNVL020 at the end of the INVMAINT table load process. It reports the number of input records processed, the number output, and the number of errors.

System action:
If there are any errors, the error is reported and the INVMAINT process terminates.

User response:
If an error is indicated, correct the invalid INVMAINT request card and rerun.

BRNVL02003E mm/dd/yy hh:mm:ss – xxxxxxxx LOAD FAILURE, REASON xxxxxxxxxxxxxxxxxxxxxxxx, R15 = xxxxxxxx

Explanation

This message is issued to indicate a QIO error:

EMRSN008 0008 GET INPUT ERR0R
EMRSN012 0012 QIO OPNE ERROR
EMRSN016 0016 QIO PUT ERROR
EMRSN024 0024 OPEN INPUT ERROR

User response:
Correct the error and rerun the job.

BRNVL00205E mm/dd/yy hh:mm:ss - REQUEST RTNCD=xxxx, REC:

Explanation:
This message is issued when an invalid INVMAINT card is processed. The incorrect card is listed.

User response:
Correct the card and rerun the job.

BRNVC05901E mm/dd/yy hh:mm:ss - INVMAINT TABLE LOAD ERROR, INVENTRY MAINTENANCE TERMINATED

Explanation:
This message is issued when there is an INVMAINT table load error.

User response:
Correct the error and rerun.

BRNVC05902E mm/dd/yy hh:mm:ss - QIO PUT ERROR, RC=xxxxxxxx, INVENTRY MAINTENANCE TERMINATED

Explanation:
This message issued on a QIO error.

User response:
Correct the error and rerun.

BRNVC05903E mm/dd/yy hh:mm:ss - BRNVC059: PARM ERROR- xxxxxxxxxxxxxxxxxxxx INVMAINT FUNCTION TERMINATED

Explanation:
This message is issued on a parameter error. The error is listed.

User response:
Correct the error and rerun.

BRNVC05904E mm/dd/yy hh:mm:ss - BRNVC059: DYNALLOC ERROR - R15(xx) S99ERR(xxxx) DSN:

Explanation:
This message is issued on a dynamic allocation error. There are two files subject to dynamic allocation: NEGFILE and INVMRPT. The DYNALLOC return and reason codes are listed along with the data set name.

User response:
Correct DYNALLOC error and rerun.

BRNVC05905E mm/dd/yy hh:mm:ss - BRNVC059: ERROR xxxxxxxxxxxxxxxxxxxxxxxx INVENTRY MAINTENANCE FUNCTION TERMINATED

Explanation:
This message indicates a processing error. The specific error is reported.

User response:
Correct the error and rerun.

BRNVC05506I mm/dd/yy hh:mm:ss - BRNVC059: ALLOCATING XXXXXXXX FILE:

Explanation:
This message indicates the documented (XXXXXXXX) file allocated and reports the DSN.

User response:
None. Informational message.

BRNVC05907I mm/dd/yy hh:mm:ss - BRNVC059: INVENTRY MAINTENANCE REQUEST xxxxxxx COMPLETE.

Explanation:
This indicates the completion of the INVMAINT request. The request ID is reported.

User response:
None. Informational message.

BRNVC05908E mm/dd/yy hh:mm:ss - INVMAINT CHECK POINT FILE ERROR, INVENTRY MAINTENANCE TERMINATED

Explanation:
This message indicates an error with the INVCKPT file.

User response:
Correct the error and rerun.

BRNVC05909I mm/dd/yy hh:mm:ss - BRNVC059: INVENTRY MAINTENANCE STATUS:INTERVAL

Explanation:
This message is issued throughout the RTA life. It indicates the state of the INVMAINT process.

User response:
None. Informational message.

BRNVC05910I mm/dd/yy hh:mm:ss - BRNVC059: ALLOCATING BKPINV FILE:

Explanation:
This message indicates the BKPINV file allocation and the DSN.

User response

None.

The command processor is used to initiate an INVMAINT process. There are two commands, INVMSTRT and INVMSTOP:

- INVMSTRT - This command starts the INVMAINT process. It sets the INVMAINT bit on and the process starts at the next RTA interval.
- INVMSTOP - This command turns off the INVMAINT bit, and if a process is executing, the process terminates.

INVMAINT can be run via batch. Sample JCL follows:

```
//SAMPBAT JOB
//STEP2 EXEC PGM=BRNVB059
//STEPLIB  DD DSN=<Prod Loadlib>,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//INVENTRY DD DISP=SHR,DSN=<INVENTRY>
//PARMLIB  DD DSN=<HLQ USERMAC.>,DISP=SHR
//INVCKPT  DD DSN=<HLQ INVCKPT,DISP=SHR
//RTDPRINT DD SYSOUT=*
```

The INVMAINT process produces report files used as input to the report program, BRNVB062. Sample JCL follows:

```
//SAMPRPT JOB
//STEP1 EXEC PGM=BRNVB062
```

```
//PARM='INVNAME=xxxxxxxx,INVPFX=xxxxxxx'
//STEPLIB  DD DSN=<Prod Load Library>,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//REPORT   DD SYSOUT=*
```

The INVNAME and INVPFX must match the values specified when the INVMAINT process was requested, either via RTA or batch.

BRNVC06414I mm/dd/yy hh:mm:ss - BRNVC064: CATALOG *icf.catalog.name* ALLOC ERROR RSN(*rrrr*), SKIP

Explanation:
Issued by BRNVC064 when attempting dynamic allocation of catalog ICF.CATALOG.NAME fails with reason code *rrrr*.

System action:
Processing of the identified catalog is skipped.

User response
For further details on DYNALLOC reason codes, refer to the IBM manual:

z/OS MVS Programming: Authorized Assembler Services Guide

.

BRNVSPLT Messages

There are two types of messages that can be generated by the BRNVSPLT program: informational and error. Informational messages report startup and processing status and final program results. These require no user response. An error message is a serious error that causes BRNVSPLT processing to end with a return code of 8. The message formats are:

Table 74. INVSPLT messages

Message	Meaning	Return Code
BRNVSPLT##I	Informational	N/A
BRNVSPLT##E	Error	8

INVSPLT01I mm/dd/yyyy hh:mm:ss - *message_text*

Explanation
This multi-line message reports the number of various file calls against the INVENTRY file during BRNVSPLT processing. The *message_text* is:

```
GET/KEY GET/SEQ INSERTS UPDATES POINTS
DELETES
```

INVENTRY CLOSE:

```
nnn  nnn  nnn  nnn  nnn  nnn
```

User response:
None. Informational message.

INVSPLT02I mm/dd/yyyy hh:mm:ss - BRNVSPLT COMPLETE

Explanation:
Signals the completion of BRNVSPLT processing.

User response:
None. Informational message.

INVSPLT03I mm/dd/yyyy hh:mm:ss - SUBTASK: xxxxxxxx {ATTACHED | SHUTDOWN ISSUED}

Explanation:
Indicates that the named subtask has been attached or shut down.

User response:
None. Informational message.

INVSPLT04I mm/dd/yyyy hh:mm:ss - STATUS - RECID(x) DSN(xxxxxxxx) APPL(xxxxxxxx)

Explanation:
An informational message issued every 100,000 records during BRNVSPLT processing.

User response:
None. Informational message.

INVSPLT05I mm/dd/yyyy hh:mm:ss - *message_text*

Explanation
Issued at the completion of BRNVSPLT processing indicating the number of base and component INVENTRY records read and the number of NEGFILE records written. Consecutive informational messages contain the following *message_text*:

```
TOTALS: B RECORDS READ: nnn B RECORD
LOCATES: nnn
```

TOTALS: C RECORDS READ: nnn C RECORD LOCATES: nnn

TOTALS: NEGFILE RECORDS: nnn

User response:
None. Informational message.

INVSPLT06I **mm/dd/yyyy hh:mm:ss - {PROG | PARM} DEFINED NUMBER OF SUBTASKS: nn**

Explanation:
An informational message indicating the number of BRNVSPLT subtasks as defined by the program or the parameter file.

User response:
None. Informational message.

INVSPLT10I **mm/dd/yyyy hh:mm:ss - NORMAL SUB-TASK TERMINATION - TCB: xxxxxxxx - COMPLETION CODE: 00000000 (INVTCBx :INVSTSK)**

Explanation:
An informational message indicating that subtask INVTCBx has terminated normally.

User response:
None. Informational message.

INVSPLT10E **mm/dd/yyyy hh:mm:ss - ***EXC SUB-TASK TERMINATION - TCB: xxxxxxxx - COMPLETION CODE: xxxxxxxx (INVTCBx :INVSTSK)**

Explanation:
An error message indicating that subtask INVTCBx has terminated abnormally.

User response:
Rerun. If the problem persists, contact Technical Support.

BRNADHSM Messages

This section documents the BRNADHSM messages.

READHSM01I **CAN NOT OPEN xxxxxxxxx FILE (DD yyyyyyyy). RETRY IN 20 SEC.**

Explanation:
BRNADHSM tried to open xxxxxxxxx file (DD *yyyyyyyy*) but was unsuccessful. xxxxxxxxx= BCDS or BKUPTBL. *yyyyyyyy*= BRNADHSM or BKUPTBL. The program waits and retries the open in 20 seconds. This can happen up to 5 times until the file is available. If the file remains unavailable, the program is terminated.

User response:
The program retries every twenty seconds for up to 5 times until terminated. If the program is terminated, contact Technical Support.

READHSM02I **RETURN CODE = "*nnnnnnnn*"**

Explanation:
This informational message is included at the end of program processing to identify the return code or reason code.

User response:
None. Informational message.

READHSM03I **PROGRAM COMPLETED!**

Explanation:
BRNADHSM processing is completed.

User response:
None. Informational message.

READHSM01W **VOLUME *vvvvvv* DEVICE TYPE NOT FOUND, DEFAULT *uuuuuu* USED**

Explanation:
BRNADHSM inquiries HEX21 table records to extract backup volume device type. Volume *vvvvvv* was not found, default device type was used.

User response:
Ensure that the BCDS file (DD BRNADHSM) is properly monitoring backups. Rerun the job. If the problem persists, contact Technical Support

READHSM02W **FREEMAIN ERROR.**

Explanation:
An error occurred when BRNADHSM tried to free the main storage space for table processing.

User response:
Contact Technical Support.

READHSM00E **CAN NOT OPEN REPORT FILE.**

Explanation:
BRNADHSM requires a data set for report output.

User response:
Ensure that no other job is using the identified data set and that the data set is not out of space or control blocks. Rerun the job. If the problem persists, contact Technical Support.

READHSM01E **CAN NOT OPEN BCDS FILE.**

Explanation:
BRNADHSM tried to open xxxxxxxxx file (DD yyyyyyyy) but was unsuccessful. xxxxxxxxx= BCDS or BKUPTBL. yyyyyyyy= BRNADHSM or BKUPTBL.

User response:

Ensure that no other program is using this resource. Rerun the job. If the problem persists contact Technical Support.

READHSM02E GETMAIN ERROR.

Explanation:
An error occurred when BRNADHSM tried to obtain main storage space for table processing.

User response:
Ensure that the region parameter on the job card is at least 1M. Fix this and rerun the job. If the problem persists, contact Technical Support.

READHSM03E CAN NOT OPEN DDOUT FILE.

Explanation:
BRNADHSM requires a data set for detail output.

User response:
Ensure that no other job is using the identified data set and that the data set is not out of space or control blocks. Rerun the job. If the problem persists contact Technical Support.

READHSM04E TESTCB MACRO FOUND WRONG INFO FOR HEX21 REC'

Explanation:
An error occurred after BRNADHSM was unable to point to a HEX21 key in the BCDS file.

User response:
Ensure that the BCDS file (DD BRNADHSM) is properly monitoring backups. Rerun the job. If the problem persists, contact Technical Support.

READHSM05E NO HEX21 RECORDS FOUND IN BCDS

Explanation:
BRNADHSM did not find any HEX21 records in the BCDS file.

User response:
Ensure that the BCDS file (DD BRNADHSM) is properly monitoring backups. Rerun the job. If the problem persists, contact Technical Support.

READHSM06E END OF BCDS FILE DD=BRNADHSM

Explanation:
BRNADHSM did not find any HEX21 records in the BCDS file.

User response:
Ensure that the BCDS file (DD BRNADHSM) is properly monitoring backups. Rerun the job. If the problem persists contact Technical Support.

READHSM07E ERROR SORTING BRNADHSM DD

Explanation:
An error occurred when BRNADHSM tried to sort file BRNADHSM DD for HEX29 records.

User response:
Check the MESSAGES DD for any sort warning or error messages, ensure that the BCDS file (DD BRNADHSM) is properly logging backups. Rerun the job. If the problem persists, contact Technical Support.

READHSM08E CAN NOT OPEN DDSORT FILE.

Explanation:
BRNADHSM tried unsuccessfully to open sort file (DD DDSORT).

User response:
Check the MESSAGES DD for any sort warning or error messages, ensure that no other job is using the identified data set and that the data set is not out of space or control blocks. Rerun the job. If the problem persists, contact Technical Support.

Real-Time Dispatcher Messages

This section documents the RTS messages.

CKKR003001I mm/dd/yyyy hh:mm:ss – DYNALLOC (CKKRC003) STARTED, TCB: xxxxxxxx

Explanation

Issued by CKKRC003 (RTD SMF Processor) where:

xxxxxxxx - TCB address for CKKRC030 (dynamic SMFDATA allocation)

User response:
None. Informational message.

CKKR003002I mm/dd/yyyy hh:mm:ss - SUB-TASK TERMINATION - TCB: xxxxxxxx COMPLETION CODE: *yyyyyyyy* (DYNALLOC:CKKRC003)

Explanation

Issued by CKKRC003 at the end of the CKKRC030 sub task where:

xxxxxxxx - TCB address
yyyyyyyy - sub task completion code

When the subtask terminates abnormally <===*EXC is appended to the text for easy identification.

User response:

Rerun task. If abnormal termination continues, contact Technical Support.

CKKR003003I mm/dd/yyyy hh:mm:ss - CHECKPOINT ERROR – FIND

Explanation:
Issued by CKKRC003 on a QIO CHECK POINT FIND request.

User response:
None. Informational message.

CKKR003004I mm/dd/yyyy hh:mm:ss - CHECKPOINT ERROR – GET

Explanation:
Issued by CKKRC003 on a QIO CHECK POINT GET request.

User response:
None. Informational message.

CKKR003005I mm/dd/yyyy hh:mm:ss – CHECK POINT ERROR – UPDATE

Explanation:
Issued by CKKRC003 on a QIO CHECK POINT UPDATE request.

User response:
None. Informational message.

CKKR003006I mm/dd/yyyy hh:mm:ss - SMF D/S CLOSE - xxxxx RECORDS FROM: yyyy.ddd hh:mm:ss.th TO: yyyy.ddd hh:mm:ss.th

Explanation:
Issued by CKKRC003 after processing an SMF data set providing information regarding the number of SMF records and the time frame.

User response:
None. Informational message.

CKKR003007I mm/dd/yyyy hh:mm:ss - CHECKPOINT INFO - LRECL = xxxx RECORDS = xx,xxx

Explanation:
Issued by CKKRC003 after processing an SMF data set providing information regarding the LRECL of the checkpoint file and the number of records in the check point file.

User response:
None. Informational message.

CKKR003009E mm/dd/yyyy hh:mm:ss - SMF MANAGER (CKKRS009) ERROR xxxxxxxxxxxxxxxxxxx

Explanation:
Issued by CKKRC003 when an error condition is returned by CKKRS009, the SMF read manager. The error is listed in the message. Additionally, CKKRS009 issues messages.

User response:
Evaluate the messages, correct, and if necessary, contact Technical Support

CKKR003010I mm/dd/yy hh:mm:ss - WAITING FOR TERMINATION OF SUB-TASK: xxxxxxxx

Explanation:
This message is issued by CKKRC003 during the address space shutdown. The identified RTA subtask (xxxxxxxx) has not completed its end of job processing.

User response:
None needed, unless a problem is suspected in the identified subtask that shows that the subtask does not complete. If that is the case, the region should be canceled.

CKKR008001I COMMAND PASSED BY CALLER:

Explanation:
Issued by CKKRC008. Identifies the command issued.

User response:
None. Informational message.

CKKR008002E NO COMMAND WAS PASSED, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008 when no command was passed to the processor.

User response:
Retry and if necessary, contact Technical Support.

CKKR008002I COMMAND PROCESSOR BUSY, WAITING FOR THE COMMAND PROCESSOR TO BECOME AVAILABLE

Explanation:
Issued by CKKRC008. The RTCMDQ already has a command to process. The processor waits for it to be available.

User response:
None. Informational message.

CKKR008003E INVALID COMMAND WAS PASSED, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The command issued is invalid.

User response:
Correct and retry.

CKKR008003I WAITING COMMAND command RETURN

Explanation:

Issued by CKKRC008 while waiting for the command result to return from RTSMF. This message is displayed both on DD=RTDPRINT and operator console when the wait time exceeds 30 minutes.

User response:
None. Informational message.

CKKR008005E COMMAND PROCESSOR STILL BUSY, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008 after waiting 15 seconds for RTCMDQ to be available.

User response:
Retry. Contact Technical Support if necessary.

CKKR008005I COMMAND QUEUE DS (TEST) IN USE BY ANOTHER JOB/USER, WAITING FOR DATASET TO BECOME AVAILABLE

Explanation:
Issued by CKKRC008 when another process is updating the command queue data set. CKKRC008 waits for it to be available.

User response:
None. Informational message.

CKKR008006E COMMAND QUEUE DATASET STILL IN USE, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The command queue data set is still in use, CKKRC008 ends.

User response:
Retry.

CKKR008006I COMMAND QUEUE DS (HAVE) IN USE BY ANOTHER JOB/USER, WAITING FOR DATASET TO BECOME AVAILABLE

Explanation:
Issued by CKKRC008 when another process is updating the command queue data set. CKKRC008 waits for it to be available.

User response:
None. Informational message.

CKKR008007E ENQUEUE FAILED FOR COMMAND QUEUE DATASET, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program was unable to have an enqueue on the command queue data set. CKKRC008 ends.

User response:
Retry.

CKKR008007I COMMAND HAS BEEN QUEUED FOR R/T SMF PROCSSING

Explanation:
Issued by CKKRC008. The command is placed in the command queue data set (RTCMDQ) for processing.

User response:
None. Informational message.

CKKR008008E OPEN FAILED FOR COMMAND QUEUE DATASET, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program was unable to open the command queue data set. CKKRC008 ends.

User response:
Retry. If the problem reoccurs, check job executions DD=JESMSGLG for any IEC type messages. Document the error and contact Technical Support.

CKKR008008I COMMAND COMPLETE: RETURN CODE= ####

Explanation:
Issued by CKKRC008 when the processed command is complete.

User response:
None. Informational message.

CKKR008009E INVALID POINTER IN COMMAND QUEUE CONTROL RECORD, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program found an invalid record pointer in the command queue data set. CKKRC008 ends.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008010E UNABLE TO READ COMMAND QUEUE CONTROL RECORD, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program was unable to read the command queue data set control record. CKKRC008 ends.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008011E UNABLE TO READ COMMAND QUEUE DATASET, PROGRAM TERMINATING

Explanation:

Issued by CKKRC008. The program was unable to read the command queue data set record as indicated by the control record. CKKRC008 ends.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008012E COMMAND QUEUE IS FULL, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program was unable to add another command request to the command queue data set. Maximum command request number is 100. CKKRC008 ends.

User response:
Retry.

CKKR008013E UNABLE TO UPDATE COMMAND QUEUE CONTROL RECORD, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program was unable to read the command queue control record. CKKRC008 ends.

User response:
Retry. If the problem reoccurs, document the error and call Technical Support.

CKKR008014E UNABLE TO UPDATE COMMAND QUEUE DATASET, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program was unable to read the command queue data set record as indicated by the control record. CKKRC008 ends.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008015E R/T SMF COMMAND PROCESSOR NOT ACTIVE, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. The program found that RTSMF was not running.

User response:
Start RTSMF and retry CKKRC008.

CKKR008016E INVALID CONTROL RECORD IN COMMAND QUEUE, PROGRAM TERMINATING (%%%%%%%%)

Explanation:
Issued by CKKRC008. The program found that one of the command queue control record pointers was invalid. The symbol (%%%%%%%%) indicates the program routine where the invalid pointer was found.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008017E SOCKET API MACRO FAILURE, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. A TCPIP socket macro call failed.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008016E RT/SMF QUIESING. REQUEST IS BYPASSED, PROGRAM TERMINATING

Explanation:
Issued by CKKRC008. RT/SMF is in the process of terminating. This request is not processed.

User response:
Retry when RTSMF is up.

CKKR008020I COMMAND HAS BEEN SENT TO RT/SMF COMMAND PROCESSOR

Explanation:
Issued by CKKRC008. RT/SMF is now processing the requested command.

User response:
None. Informational message.

CKKR008021I WAITING FOR ECB TO BE POSTED: %%%%%%%%

Explanation:
Issued by CKKRC008. RT/SMF is waiting for TCPIP to post the listed ECB.

User response:
None. Informational message.

CKKR008022I CONNECTION TO RT/SMF COMMAND PROCESSOR ESTABLISHED; HOST ID: % PORT: #

Explanation:
Issued by CKKRC008. Connection to RT/SMF, hostid= % and port=#, is established.

User response:
None. Informational message.

CKKR008023E UNABLE TO ESTABLISH CONNECTION TO RT/SMF COMMAND PROCESSOR: ezatype FAILED (ERRNO=#)

Explanation:
Issued by CKKRC008 to indicated that a TCPIP call failed with ERRNO=#.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR008025E INVALID HOSTID SPECIFIED BY CALLER; HOST ID: % PORT: #

Explanation:
Issued by CKKRC008. Program found an invalid value for hostid in the DD=HOSTIDDD record.

User response:
Correct and retry.

CKKR008099I PROGRAM TERMINATING ABNORMALLY

Explanation:
Issued by CKKRC008. Final notification that program ended in error.

User response:
See the prior error message in DD=RTDPRINT for the actual cause of termination.

CKKR030001I mm/dd/yyyy hh:mm:ss-CRITICAL ERROR-UNABLE TO LOCATE >CKPT< ENTRY IN TABLE DEFINITION PARMS

Explanation:
Issued by CKKRC003 (RTD dynamic allocation sub task) indicating VFITABLS does not contain a check point entry.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR030010I mm/dd/yyyy hh:mm:ss - NO DSN READY - CSIFLG = XX - DSN = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

Explanation:
Issued by CKKRC003 when the call to IGGCSI00 returns an error or there are no data sets that match the SMF$ mask (CSIFLG=00). The error flag is reported in CSIFLG = XX, and the DSN is listed.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR030002I mm/dd/yyyy hh:mm:ss - CRITICAL ERROR - UNABLE TO OPEN CHECKPOINT DATASET - DDNAME = xxxxxxxxx

Explanation:
Issued by CKKRC003 (RTD dynamic allocation sub task). The DDNAME (xxxxxxxxx) failed to open.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR030003I mm/dd/yyyy hh:mm:ss - CHECKPOINT INFO - LRECL = xxxx RECORDS = xx,xxx

Explanation:
Issued by CKKRC003 after processing an SMF data set providing information regarding the LRECL of the checkpoint file and the number of records in the check point file.

User response:
None. Informational message.

CKKR030004I mm/dd/yyyy hh:mm:ss - CRITICAL ERROR - UNABLE TO LOCATE SMFALLOC SUB-TASK CONTROLS (RTSSBENT)

Explanation:
Issued by CKKRC003 when unable to locate the sub task control table RTSSBENT.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR030005I mm/dd/yyyy hh:mm:ss - CATALOG ERROR - CSIC FLG = xx – DSN = x

Explanation:
Issued by CKKRC003 on a CSI error.

User response:
Retry. If the problem reoccurs, document the error and contact Technical Support.

CKKR030006I mm/dd/yyyy hh:mm:ss - DYNAMIC ALLOC - DSN = x

Explanation:
Issued by CKKRC003 when an SMF$ (identified as DSN = X) data set is allocated.

User response:
None. Informational message.

CKKR030007I mm/dd/yyyy hh:mm:ss - FIRST CATALOG ACCESS FOR SYSTEM=xxxx, SCRATCH=Y, SMFDSN=z

Explanation:
Issued by CKKRC030 when the first SMF$ (identified as DSN = Z) data set is allocated. This information is taken from the SMFPARMS member.

User response:
None. Informational message.

CKKR030008I mm/dd/yyyy hh:mm:ss - STOPAFT THRESHOLD REACHED, ADDRESS SPACE WILL BEGIN TO QUIESE

Explanation:
Issued by CKKRC030 at end of file on the last SMF$ data set in this processing window.

System action:
Based on the value specified in VFIPARMS (default is 1) CKKRC003 waits for any newly created SMF$ data sets before starting the termination process.

User response:
None. Informational message.

CKKR030009I mm/dd/yyyy hh:mm:ss - STOPAFT PROCESSING IN EFFECT: xxx ATTEMPT(S) LEFT

Explanation:
Issued by CKKRC030 at end of file on the last SMF$ data set in this processing window to indicate how many STOPAFT attempts remain.

User response:
None. Informational message.

CKKR030011I mm/dd/yyyy hh:mm:ss - DUPLICATE DSN FOUND: DSN=xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx, VOL=vvvvvv, DATE=cyyddd

Explanation:
Issued by CKKRC030 when a duplicate entry is found in the RTACKPT file. This should only occur when SCRATCH=NO is specified in the SMFPARMS member is USERMAC.

User response:
None. Informational message.

CKKR030012I mm/dd/yyyy hh:mm:ss - REPROCESS DSN=

Explanation:
Issued by CKKRC030 when an SMF$ data set is being reprocessed. An entry exists in the RTACKPT data set, but the data set was not completely processed. The system resumes processing this data set.

User response:
None. Informational message.

CKKR030013I mm/dd/yyyy hh:mm:ss - DD(xxxxxxxxx) NOT AVAILABLE DSN=xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx.xxxxxxxxx

Explanation:
Issued by CKKRC030 every 5 minutes when a data set matching the defined SMF$ mask is not found. The DD name is also listed.

System action:
After 15 minutes, a request to remove the DD or continue is issued.

User response:
See message CKKR030013.

CKKR030017I mm/dd/yyyy hh:mm:ss - DD(xxxxxxxx) REMOVED FROM CURRENT PROCESS

Explanation:
Issued by CKKRC030 when a request to remove a system from processing is processed. The identified DD name is listed.

User response:
A request to restore the system can be issued through the command processor.

CKKR003008I mm/dd/yyyy hh:mm:ss - SMF$ DATASET SCRATCHED, ALREADY PROCESSED

Explanation:
Issued by CKKRC030 when an SMF$ data set is scratched because it was already processed by RT/SMF.

User response:
None. Informational message.

CKKR03014A DD(SMFDATAx) NOT AVAILABLE, REPLY: CONTINUE OR REMOVE

Explanation

WTOR issued by CKKRC030 when an SMF$ data set for the identified system is not available.

- A reply of CONTINUE tells the system to keep looking for the SMF$ matching data set.
- A reply of REMOVE removes the identified system from further processing.

Regardless of the reply the system continues to look for a matching data set. Use the command processor to restore a removed data set to processing. The command processor can be invoked via the TSO interface or via the batch interface using CKKRC008.

The format of the command is:

```
RESTORE SMFDATAx.
```

The SMFDATA suffix indicates the system you wish to reinstate for processing, for example, SMFDATA (system 1) or SMFDATA2 (system 2). A sample CKKRC008 JCL stream follows:

```
//<jobcard>
//STEP1 EXEC PGM=CKKRC008,
// PARM='RESTORE SMFDATA2' <= restores system 2
//STEPLIB  DD DSN=<DRVFI library>,DISP=SHR
//QUEUE    DD DSN=<DRVFI command queue
dataset(RTACMDQ)>,DISP=SHR
//RTDPRINT DD SYSOUT=*
```

You cannot remove the last processing system; that is, you must keep one system active.

User response:
See SMF Extractor Parameters for definitions of SMFPARMS for processing system identification.

CKKR9990001I mm/dd/yyyy hh:mm:ss - SUB-TASK TERM. - TASK = xxxxxxxxx PGM = xxxxxxxxx R/T = nnn COUNTS = nnn,nnn,nnn nnn,nnn,nnn

Explanation:
Issued by CKKRT999 at end of processing to document SMF record types and counts.

User response:
None. Informational message.

CKKRB00101I mm/dd/yy hh:mm:ss – CKKRB001: RTS SYSTEM LOGGER INITIALIZED

Explanation:
This message is issued when the RTS System Logger starts.

User response:
None. Informational message.

CKKRB00102I mm/dd/yy hh:mm:ss – CKKRB001: IXGCONN,

Explanation:
LOG STREAM: xxxxxxxxxxxxxxxxxxxxxxxxx This message is issued when the log stream is connected to the System Logger. The user RTS log stream is reported.

User response:
None. Informational message.

CKKRB00103I mm/dd/yy hh:mm:ss CKKRB001: LOGR ERROR, REQ: xxxxxxxx, - RC: xxxxxxxx, REASON CODE: xxxxxxxxxxxx

Explanation:
This message is issued when a System Logger request fails. The request, return code and reason code are reported. Review z/OS Vxx.x MVS Assembler Services Reference IAR-XCT for information on the logger command and direction for return code/reason code interpretation.

User response:
Review error codes for correction.

CKKRC00190I RT/SMF READY FOR PROCESSING

Explanation:
Issued by CKKRC001 to indicate RT/SMF is ready for processing; all subtasks used with this execution are attached and ready for input.

User response:
None. Informational message.

CKKRC00191I RT/SMF TERMINATING

Explanation:
Issued by CKKRC001 to indicate RT/SMF is finished and the started task or job is ending.

User response:
None. Informational message.

CKKRC0021I mm/dd/yyyy hh:mm:ss – RTDPRINT DCB OBTAINED - ADDRESS = YYYY, SUB-POOL/ LENGTH = zzllllll

Explanation

Issued by CKKRC002 (RTD Print manager) where:

- yyyyyyyy - DCB address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRC00401I mm/dd/yyyy hh:mm:ss - COMMAND PROCESSOR STARTED - WAIT TIME: nnn.nn SECONDS

Explanation:
Reports that command processor has started and waits for nnn.nn seconds between loops to check for work.

User response:
None. Informational message.

CKKRC00402I mm/dd/yyyy hh:mm:ss - COMMAND PROCESSOR TERMINATING

Explanation:
Issued by CKKRC004 when the command processor is terminating.

User response:
None. Informational message.

CKKRC00403I mm/dd/yyyy hh:mm:ss - UID: user_id, CMDQRID=que_recid, CMD: cmd_name, cmd_text

Explanation:
Issued by CKKRC004 when a command is processed. Listing the requester's user ID, record number assigned in the DD=CMDQUEUE data set, and the command.

User response:
None. Informational message.

CKKRC00404I mm/dd/yyyy hh:mm:ss - COMMAND: PROGRAM: ATTACHED, TCB:

Explanation:
Issued by CKKRC004 when the requested command requires attaching a command processor subtask; for example BRNVC014. The command, program and TCB address are listed.

User response:
None. Informational message.

CKKRC00405I mm/dd/yyyy hh:mm:ss - COMMAND: cmd_name, TERMINATED – COMPLETION CODE: ####

Explanation:
Issued by CKKRC004 when an attached command completes.

User response:
None. Informational message.

CKKRC00406I mm/dd/yyyy hh:mm:ss - WAITING FOR TERMINATION OF ATTACHED COMMAND:

Explanation:
Issued by CKKRC004 when a request is made that requires the attach subtask processor and there is still an active request. The command processor waits 15 seconds and retry the command.

User response:
None. Informational message.

CKKRC00407I mm/dd/yy hh:mm:ss - COMMAND: cmd_name, CMDQRID=que_recid - PROCESSOR ACTIVE, RETRY IN PROGRESS

Explanation:
Issued by CKKRC004 when the command processor is busy. Command is assigned a command queue record number and retried when the command processor is available.

User response:
None. Informational message.

CKKRC00408I mm/dd/yy hh:mm:ss - WAITING FOR COMMAND QUEUE TO BECOME AVAILABLE, CURRENTLY IN USE

Explanation:
Issued by CKKRC004 when the required command processor queue RTCMDQ is unavailable for ENQ because it is in use by another user. CKKRC004 retries the ENQ at 2 second intervals until available. The message is repeated every 10 minutes until the ENQ is obtained.

User response:
None. Informational message.

CKKRC00409I mm/dd/yy hh:mm:ss - COMMAND: cmd_name, CMDQRID=que_recid FLAGGED AS process

Explanation

Issued by CKKRC004 to track command queue processing status. Processes include:

- CMDQDEL - Command queue record deleted.
- CMDQDONE - Request processing program completed.
- CMDQINIT - Clean up command queue record
- CMDQNEW - New command queue request
- CMDQPEND - Wait on request processing program to complete.
- CMDQREDO - Retry request.
- CMDQSTOP - Stop in progress

User response:
None. Informational message.

CKKRC00410I cmd_name- COMMAND IN PROCESS

Explanation:
Issued by CKKRC004 to identify the command currently processing.

User response:
None. Informational message.

CKKRC00411I cmd_name- COMMAND PROCESSOR TERMINATING

Explanation:
Issued by CKKRC004 when the command processor is terminating.

User response:
None. Informational message.

CKKRC00412E cmd_name- INVALID COMMAND

Explanation:
Issued by CKKRC004 when an invalid command is requested. The invalid command is listed.

User response:
Correct request and retry.

CKKRC00413E cmd_name- FAILED, TBL_NAME NOT LOCATED

Explanation:
Issued by CKKRC004 when the table associated with the command is not defined for example REFRESH TABLEX and TABLEX does not exist

User response:
Correct and re-issue command.

CKKRC00413E * mm/dd/yy hh:mm:ss - DIAGNOSE - FAILED, XXXXXXXX NOT LOCATED

Explanation:
This message is issued by CKKRC004 to indicate an error initiating a DIAGNOSE function. The subtask specified was incorrect or not found.

User response:
Correct and re-issue command.

CKKRC00414I TBL_NAME - REFRESH INDICATOR SET

Explanation:
Issued by CKKRC004 when a table refresh is requested. The refresh table flag is set in the table control block. At the next interval restart, the table is reloaded.

User response:
None. Informational message.

CKKRC00414I mm/dd/yy hh:mm:ss - XXXXXXXX - DIAGNOSTIC SET ON

Explanation:
This message is issued by CKKRC004 to indicate initiation of the DIAGNOSE function. The subtask is identified.

User response:
None. Informational message.

CKKRC00414I mm/dd/yy hh:mm:ss - XXXXXXXX - DIAGNOSTIC SET OFF

Explanation:
This message is issued by CKKRC004 when the DIAGNOSE off command is requested.

User response:
None. Informational message.

CKKRC00415W cmd_name- BYPASSED, TBL_NAME REFRESH IS ACTIVE

Explanation:
Issued by CKKRC004 when a table refresh command cannot process because the table is already reloading.

User response:
None. Informational message.

CKKRC00416I cmd_name- PROCESSOR ACTIVE, RETRY IN PROGRESS

Explanation:
Issued by CKKRC004 when the command subtask attach processor is active on the same requested command. Only one of the exact same request can be processed at a time. The command processor retries the request when the command is available.

User response:
None. Informational message.

CKKRC00418I cmd_name- CKKRMFCB NOT LOADED, BYPASSED

Explanation:
Issued by CKKRC004 to reactive a removed SMF$ system. The CKKRMFCB control block does not exist. The command is bypassed.

User response:
None. Informational message.

CKKRC00419I cmd_name- SMF DD NAME SPECIFIED INVALID ddn_name NOT LOCATED

Explanation:
Issued by CKKRC004 when a remove or restore SMF$ data set command cannot be processed because the requested (DDN-NAME) is not in use on the system.

User response:
Check DD name and retry.

CKKRC00420I cmd_name- SMF DD ddn_name NOT IN 'REMOVED' STATUS

Explanation:
Issued by CKKRC004 when a request to restore an SMF$ dd fails because the requested (DDN-NAME) is not active on the system.

User response:
Re-enter with correct DD name.

CKKRC00421I cmd_name- SMF DD RESTORE COMMAND COMPLETE, ddn_name WILL NOW PROCESS

Explanation:
Issued by CKKRC004 when a restore command for a removed SMF$ system is processed. RT/SMF now includes the added system during processing.

User response:
None. Informational message.

CKKRC00422E mm/dd/yyyy hh:mm:ss – COMMAND: {*startvrs* | *stopvrs* – BYPASSED {*startvrs_reasons* | *stopvrs_reasons*}

Explanation

startvrs_reasons: {RT/VRS NOT IN SYSTEM | RT/VRS ALREADY ACTIVE}

stopvrs_reasons: {RT/VRS NOT IN SYSTEM | RT/VRS NOT ACTIVE}

Issued by CKKRC004 when the command processor *startvrs* or *stopvrs* cannot be completed as requested.

User response:
Retry. If problem reoccurs, contact Technical Support.

CKKRC00422E mm/dd/yy hh:mm:ss - ENQUEUE FAILED FOR COMMAND QUEUE DATASET

Explanation:
CKKRC004 requires exclusive ENQ to the command processor queue RTCMDQ in order to update the command queue status. This message signals that CKKRC004 was able to get ENQ TYPE=TEST but unable to get ENQ TYPE=HAVE. CKKRC004 retries the ENQ process until obtaining the ENQ TYPE=HAVE.

User response:
None. If retry does not complete, contact Technical Support.

CKKRC00423I mm/dd/yy hh:mm:ss - RT/VRS ACTIVATE REQUEST ACCEPTED

Explanation:
Issued by CKKRC004 to indicate that user command STARTVRS has successfully started Simulated Recovery Processing.

User response:
None. Informational message.

CKKRC00423E mm/dd/yy hh:mm:ss – ENQUE FAILED FOR COMMAND QUEUE DATASET

Explanation:
This message is issued by CKKRC004 when the acquisition of the RTCMDQ file failed. Retry is done.

User response:
None. If retry fails, contact Technical Support.

CKKRC00424I mm/dd/yy hh:mm:ss - RT/VRS TERMINATION SCHEDULED

Explanation:
Issued by CKKRC004 to indicate user command STOPVRS has successfully ended Simulated Recovery Processing.

User response:
None. Informational message.

CKKRC00424E mm/dd/yy hh:mm:ss – UNABLE TO READ COMMAND QUEUE

Explanation:
This message is issued by CKKRC004 when the read of the RTCMDQ file fails. Retry is done.

User response:
If the retry fails, contact Technical Support.

CKKRC00425E mm/dd/yy hh:mm:ss – UNABLE TO WRITE TO COMMAND QUEUE

Explanation:
This message is issued by CKKRC004 when the write to the RTCMDQ file fails. Program processing ends.

User response:
Retry. If the error reoccurs, contact Technical Support.

CKKRC00426E mm/dd/yy hh:mm:ss – UNABLE TO PROCESS COMMANDS – INVALID POINTER TO FIRST COMMAND

Explanation:
This message is issued by CKKRC004 when the command queue control record contains an invalid pointer to the first queued request. Program processing ends.

User response:
Document the error and contact Technical Support.

CKKRC00427E mm/dd/yy hh:mm:ss – UNABLE TO PROCESS COMMANDS – INVALID POINTER TO LAST COMMAND

Explanation:
This message is issued by CKKRC004 when the command queue control record contains an invalid pointer to the last queued request. Program processing ends.

User response:
Document the error and contact Technical Support.

CKKRC00432E mm/dd/yy hh:mm:ss - COMMAND PROCESSOR INITIALIZATION FAILED: *init_fail_reason*

Explanation

At RTSMF startup, CKKRC004 initializes the command queue DD=RTCMDQ. The *init_fail reason*) explains the failure. Each of the reasons cause RTSMF to end with ABEND X'0401'.

init_fail_reason can be any of the following:

COMMAND QUEUE DATASET NOT ALLOCATED
Signals that required hilevel.RTCMDQ data set has not been predefined.

OPEN FAILED FOR COMMAND QUEUE DATASET
Signals that a required hilevel.RTCMDQ data set has not opened successfully.

UNABLE TO READ COMMAND QUEUE CONTROL RECORD
Signals that the initial attempt to read the command queue control record failed.

READ FAILED FOR COMMAND QUEUE DATASET
Indicates that after first control record initialization, the attempt to read the first command queue user record has failed.

User response:
Check your jobs DD=JESMSGL for any additional IEC type messages to determine the problem with your RTCMDQ data set. Adjust the RTCMDQ file as

indicated. Rerun RTSMF. If the problem reoccurs, document the error and contact Technical Support.

CKKRC00433E mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR EXCEEDED ERROR LIMIT

Explanation:
Issued by CKKRC004 (Command Processor) after maximum number of errors (100) is exceeded. Processor abends with return code 0402.

User response:
Check the RTSMF execution DD=RTDPRINT for the 100 error (CKKRC004##E) messages. Adjust your processing as the messages indicate. Rerun RTSMF. If the problem reoccurs, document the error and contact Technical Support.

CKKRC00490I mm/dd/yyyy hh:mm:ss – UP QNAME/RNAME: RTSMF001/ UP = > hilevel..RTCMDQ

Explanation:
Issued by CKKRC004 (Command Processor) at sub-task initialization. Specifies command queue qname/ rname.

User response:
None. Informational message.

CKKRC00499I mm/dd/yy hh:mm:ss - STOP IN PROGRESS, ACCEPTING NO NEW REQUESTS

Explanation:
Issued by CKKRC004 during orderly shutdown. No new requests are processed, current requests are completing.

User response:
None. Informational message.

CKKRC04S01I mm/dd/yyyy hh:mm:ss – RTSMF SYSID=ssss, RUNNING APF {authorized|unauthorized}

Explanation:
Issued by CKKR004S (RTD Command Processor) at initialization.

User response:
None. Informational message.

CKKRC04S02I mm/dd/yyyy hh:mm:ss - COMMAND PROCESSOR TERMINATING

Explanation:
Issued by CKKR004S (RTD Command Processor) at termination.

User response:
None. Informational message.

CKKRC04S03I mm/dd/yyyy hh:mm:ss - COMMAND PROCESSOR STARTED – WAIT TIME: ssss SECONDS

Explanation:
Issued by CKKR004S (RTD Command Processor) at initialization.

User response:
None. Informational message.

CKKRC04S06I mm/dd/yyyy hh:mm:ss – WAITING FOR TERMINATION OF ATTACHED COMMAND: cccccc

Explanation:
Issued by CKKR004S when a STOP request is made that requires the attach subtask processor and there is still an active request. The command processor waits 15 seconds and retry the command.

User response:
None. Informational message.

CKKRC04S11I mm/dd/yyyy hh:mm:ss – cccccccc COMMAND PROCESSOR TERMINATING

Explanation:
Issued by CKKR004S (RTD Command Processor) at termination

User response:
None. Informational message.

CKKRC04S30I mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR SERVER ACCEPTING REQUESTS; HOST ID: xxx.xxx.xxx PORT: #####

Explanation:
Issued by CKKR004S (RTD Command Processor) at initialization.

User response:
None. Informational message.

CKKRC04S31I mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR WAITING FOR ECB TO BE POSTED: ACCEPT

Explanation:
Issued by CKKR004S (RTD Command Processor) when listener socket is in the process of accepting a client request.

User response:
None. Informational message.

CKKRC04S33I mm/dd/yyyy hh:mm:ss – CLIENT PROCESS REQUESTING SERVICE; CLIENT ID: xxx.xxx.xxx PORT: #####

Explanation:

Issued by CKKR004S (RTD Command Processor) after the socket has been accepted from the client but before it has been given to an available CKKR004V child processor.

User response:
None. Informational message.

CKKRC04S35I mm/dd/yyyy hh:mm:ss – NO CHILD SUBTASK IS AVAILABLE, REQUEST BYPASSED

Explanation:
Issued by CKKR004S (RTD Command Processor) when all available BRNVC004V child processors are in use. Resubmit request.

User response:
None. Informational message.

CKKRC04S37I mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR SOCKET CLOSE REQUESTS. CLIENT ID: xxx.xxx.xxx PORT: #####

Explanation:
Issued by CKKR004S (RTD Command Processor) after a successful give of the client socket to the CKKR004V child processor. The child takes the socket and communicates directly to the client under a different socket. The server now closes the original socket.

User response:
None. Informational message.

CKKRC04S40I mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR REATTACHING CHILD; TCB: ########

Explanation:
Issued by CKKR004S (RTD Command Processor) after a CKKR004V child processor ends unexpectedly. Server attaches a new CKKR004V child processor with TCB=########. Maximum number of total re-attaches per Command Processor execution is the number of child processors * 2. Command Processor ends if maximum re-attach is exceeded.

User response:
None. Informational message.

CKKRC04S90I mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR QUIESING ATTACHED SUBTASKS

Explanation:
Issued by CKKR004S (RTD Command Processor) during STOP command processing after each subtask has been notified of termination processing. Each subtask is allowed to complete processing before shut down is complete.

User response:
None. Informational message.

CKKRC04S31E mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR LOGON TABLE {locate | load} FAILED

Explanation:
Issued by CKKR004S (RTD Command Processor) at sub task initialization. Program BRNVL019 used to create and load the USERMAC.(RTSAUTH) Logon authorization table has failed. Command Processor abends with return code 0401.

User response:
Make sure that USERMAC.(RTSAUTH) logon table has been created with RACF usernames and system-ids .

CKKRC04S32E mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR INITIALIZATION FAILED: SOCKET CALL {ezatype} FAILED (ERRNO=#####) | SOCKET API MACRO FAILURE FOR {ezatype} }

Explanation:
Issued by CKKR004S (RTD Command Processor) at ezatype socket failure.

User response:
Document the error and contact Technical Support.

CKKRC04S33E mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR EXCEEDED ERROR LIMIT

Explanation:
Issued by CKKR004S (RTD Command Processor) after maximum number of errors (100) is exceeded. Command Processor abends with return code 0402.

User response:
None. Informational error message

CKKRC04S34E mm/dd/yyyy hh:mm:ss – COMMAND PROCESSOR SELECT SOCKET(#####) UNEXPECTED {function} ERROR

Explanation:
Issued by CKKR004S (RTD Command Processor) after socket select processing had an unexpected error in function: Exception, Number, Read, or Write.

User response:
Document the error and contact Technical Support.

CKKRC03201I mm/dd/yy hh:mm:ss - CKKRC032: RTS SYSTEM LOGGER INITIALIZED

Explanation:
This message is issued when the RTS System Logger starts.

User response:

None. Informational message.

CKKRC03202I mm/dd/yy hh:mm:ss - CKKRC032: IXGCONN, LOG STREAM: xxxxxxxxxxxxxxxxxxxxxxxxx

Explanation:
This message is issued when the log stream is connected to the System Logger. The user RTS log stream is reported.

User response:
None. Informational message.

CKKRC03203I mm/dd/yy hh:mm:ss CKKRC032: LOGR ERROR, REQ: xxxxxxxx, - RC: xxxxxxxx, REASON CODE: xxxxxxxxxxxx

Explanation:
This message is issued when a System Logger request fails. The request, return code and reason code are reported. Review z/OS Vxx.x MVS Assembler Services Reference IAR-XCT for information on the logger command and direction for return code/reason code interpretation.

User response:
Review the error codes for correction, and restart RTS to resume logger facility.

CKKRC03204E mm/dd/yy hh:mm:ss - CKKRC032: RTS LOGR FUNCTION TERMINATED DUE TO ERRORS

Explanation:
This message is issued when a System Logger error occurs. The RTS System Logger function is terminated.

User response:
Resolve error condition and restart RTS to resume RTS System Logger.

CKKRC03301I mm/dd/yy hh:mm:ss - CKKRC033 - RT/HRT: TASKNAME PROGRAM ECB1ADDR/VALUE ECB2ADDR/ VALUE ECB3ADDR/VALUE

Explanation

This message is a header and detail sequence that identifies the Task/program and the three ECB/values associated with the identified subtask. The format of the detailed message is:

BRNVC03301I - mm/dd/yy hh:mm:ss - TASKNAME PROGRAM xxxxxxx/xxxxxxx xxxxxxx/xxxxxxxx xxxxxxxx/xxxxxxxx

Example:

BRNVC03301I - mm/dd/yy hh:mm:ss - (003) *RUNNING* VFIANLY1 BRNVC001 0001E4B8/808CD138 0001E4BC/40000000 0001E4C0/00000000

This example illustrates the case where the identified subtask (VFIANALY1/BRNVC001) is executing after 3 (003) CKKRC033 monitor checks. In this case, the subtask exceeds the single SMF record processing time.

User response:
None. Informational message.

CKKRC03302E * mm/dd/yy hh:mm:ss - CKKRC033 - RT/HRT ERROR DETECTED, RT/SMF TERMINATING

Explanation:
This message indicates the RTA is ending due to a task exceeding processing time.

User response:
Investigate the identified subtask for errors, and restart.

CKKRC03303I mm/dd/yy hh:mm:ss - CKKRC033 - RT/HRT: WAKE UP

Explanation:
This is the standard message issued every 10 minutes indicating the RTA Monitor is active and executing.

User response:
None. Informational message.

CKKRC04V01I mm/dd/yyyy hh:mm:ss - CHILD: ######## STARTED – WAITING FOR WORK

Explanation:
Issued by CKKR004V (RTD Child Processor) at sub-task initialization.

User response:
None. Informational message.

CKKRC04V02I mm/dd/yyyy hh:mm:ss - CHILD: ######## TERMINATING

Explanation:
Issued by CKKR004V (RTD Child Processor) at sub-task termination.

User response:
None. Informational message.

CKKRC04V03I mm/dd/yyyy hh:mm:ss – CHILD: ######## UID: xxxxxxxx CMD: cccccccc

Explanation:
Issued by CKKR004V (RTD Child Processor) when a command is processed. Lists child processing and user request.

User response:
None. Informational message.

CKKRC04V04I mm/dd/yyyy hh:mm:ss - CHILD: ######## COMMAND: cccccccc

PROGRAM: pppppppp ATTACHED, TCB: ttttttt

Explanation:
Issued by CKKR004V (RTD Child Processor) when the requested command requires attaching a command processor subtask; for example BRNVC014. The command, program and TCB address are listed.

User response:
None. Informational message.

CKKRC04V05I **mm/dd/yyyy hh:mm:ss – COMMAND: cccccccc TERMINATED, COMPLETION CODE: ####**

Explanation:
Issued by CKKR004V (RTD Child Processor) when an attached command completes.

User response:
None. Informational message.

CKKRC04V06I **mm/dd/yyyy hh:mm:ss – CHILD: ######## WAITING FOR TERMINATION OF ATTACHED COMMAND: cccccc**

Explanation:
Issued by CKKR004V (RTD Child Processor) when a STOP request was made and an attach subtask processor is still active.

System action:
The child processor waits 15 seconds and retry the command.

User response:
None. Informational message.

CKKRC04V07I **mm/dd/yyyy hh:mm:ss – CHILD: ######## COMMAND: xxxxxxxx – PROCESSOR ACTIVE, TRY AGAIN LATER**

Explanation:
Issued by CKKR004V (RTD Child Processor) when the command processor is busy.

User response:
Retry the command at a later time.

CKKRC04V25E **mm/dd/yyyy hh:mm:ss – COMMAND: xxxxxxxx – NOT AUTHORIZED FOR UID: uuuuuuuu, REQUEST BYPASSED**

Explanation:
Issued by CKKR004V (RTD Child Processor) when the command's authorization number is lower than the userid USERMAC.(RTSAUTH) level. Authorization numbers are from 1 through 9. Highest authorization is 1. A userid with authorization number of 9 is not permitted to execute commands with authorizations lower than 9.

User response:
Informational message. Contact your System Programmer regarding relevant authorization.

CKKRC04V31I **mm/dd/yyyy hh:mm:ss – CHILD: ######## WAITING FOR ECB TO BE POSTED: TAKESOCKET**

Explanation:
Issued by CKKR004V (RTD Child Processor) when child is in the process of accepting the client's request.

User response:
None. Informational message.

CKKRC04V32E **mm/dd/yyyy hh:mm:ss – CHILD: ######## INITIALIZATION FAILED: {SOCKET CALL {ezatype} FAILED (ERRNO=#####) | SOCKET API MACRO FAILURE FOR {ezatype} }**

Explanation:
Issued by CKKR004V (RTD Child Processor) at ezatype socket failure.

User response:
Document the error and contact Technical Support.

CKKRC04V33E **mm/dd/yyyy hh:mm:ss – CHILD: ######## EXCEEDED ERROR LIMIT**

Explanation:
Issued by CKKR004V (RTD Child Processor) after maximum number of errors (100) is exceeded. Child Processor abends with return code 0402.

User response:
Check the RTSMF execution DD=RTDPRINT for the 100 error (CKKRC004##E) messages. Adjust your processing as the messages indicate. Rerun RTSMF. If the problem reoccurs, document the error and contact Technical Support.

CKKRC04V34E **mm/dd/yyyy hh:mm:ss – CHILD: ######## NAME TOKEN SERVICES FAILURE**

Explanation:
Issued by CKKR004V (RTD Child Processor) after a non-zero return code from Name Token Services.

User response:
Document the error and contact Technical Support.

CKKRCAUT04E **mm/dd/yyyy hh:mm:ss - UID: uuuuuuuu CMD: LOGON TERMINATED, USERNAME INVALID TO {RTSAUTH | RACF}**

Explanation:

Issued by RTSAUTH (RTD Authorization Program) when the requesting userid is not defined to RACF or the USERMAC.(RTSAUTH) table.

User response:
Informational message. Check your username and retry.

CKKRCAUT08E **mm/dd/yyyy hh:mm:ss - UID: uuuuuuuu CMD: LOGON TERMINATED, PASSWORD INVALID TO RACF**

Explanation:
Issued by RTSAUTH (RTD Authorization Program) when the requesting user password is invalid to RACF.

User response:
Check password and retry.

CKKRCAUT10E **mm/dd/yyyy hh:mm:ss - UID: uuuuuuuu PGM: RTSCAUTH TERMINATED, (SEE RT/SMF JESMSGLG FOR ERRMSG)**

Explanation:
Issued by RTSAUTH (RTD Authorization Program) when an unexpected error has occurred.

User response:
Document the error and contact Technical Support.

CKKRCAUT16E **mm/dd/yyyy hh:mm:ss – COMMAND: LOGON INVALID WHEN RTSMF RUNNING APF UNAUTHORIZED**

Explanation:
Issued by RTSAUTH (RTD Authorization Program) when command LOGON is requested and IBM Z Batch Resiliency load library has not been APF Authorized.

User response:
Contact your System Programmer with the contents of the message.

CKKRCNTS01I **mm/dd/yyyy hh:mm:ss – {NTOKEN | TOKERR } {function} NAME1: nnnnnnnn NTSNAME2: nnnnnnnn**

Explanation
Issued by CKKRCNTS (RTD Name Token Services Program) when processing a NTS request.

Name Token Services Functions:

- GM GETMAIN storage
- RT Read
- CR Create
- DL Delete
- WT Write

User response:
If TOKERR, document the error and contact Technical Support.

CKKRRTB001E **mm/dd/yyyy hh:mm:ss - BRNVC007: OPEN FAILED FOR BKUPTBL FILE RC= xxxx, FEEDBK= YYYY**

Explanation:
Issued by BRNVC007 on a BKUPTBL file open error. The return code (RC) and feedback (FEEDBK) fields show the respective VSAM codes.

User response:
See the IEC161I message in the job log and *z/OS DFSMS Macro Instructions for Data Sets* for more information regarding this error.

CKKRRTB002E **mm/dd/yyyy hh:mm:ss - BRNVC007: GET FAILED FOR BKUPTBL FILE RC= xxxx, FEEDBK= YYYY**

Explanation:
Issued by BRNVC007 on a BKUPTBL file get error. The return code (RC) and feedback (FEEDBK) fields show the respective VSAM codes.

User response:

CKKRRTB003E **mm/dd/yyyy hh:mm:ss - BRNVC007: PUT FAILED FOR BKUPTBL FILE RC= xxxx, FEEDBK= YYYY**

Explanation:
Issued by BRNVC007 on a BKUPTBL file put error. The return code (RC) and feedback (FEEDBK) fields show the respective VSAM codes.

User response:

CKKRRTB004E **mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL QIO GET ERROR, RC=> xxx**

Explanation
Issued by BRNVC007 on a BKUPTBL QIO get error. The return code (RC) shows the QIO return code value.

Return Code	Function	Meaning
0	All	Request complete, no errors
1	All	File # less than 0 or greater than 31
2	Open	File is already open
3	All	File never opened

Return Code	Function	Meaning
4	Get	End of file, no record returned
5	Find	Record with matching key not found; also occurs when file is empty
6	Update	Get not previously issued for this file
7	Load/Save	Unable to open data set, check for missing DD statement
8	Read/Write	Relative record number outside record range for this file
9	Load/Save	Record format not fixed (F or FB)

User response:

CKKRRTB006E mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL QIO PUT ERROR, RC=>

Explanation:
Issued by BRNVC007 (RTB) when an error is returned by the QIO routine.

User response:
Document the return code and contact Technical Support.

CKKRRTB007E mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL QIO TABLE ERROR, RC=> xxx

Explanation:
Issued by BRNVC007 on a BKUPTBL QIO load error. The return code (RC) shows the QIO return code value.

User response:
Retry.

CKKRRTB008E mm/dd/yyyy hh:mm:ss - BRNVC007: NO BKUPTBL TABLE FOUND IN SYSTEM

Explanation:
Issued by BRNVC007 when no BKUPTBL entry is found in the RTS tables member (VFITABLS) in USERMAC.

User response:
Retry. If error reoccurs, contact Technical Support.

CKKRRTB009E mm/dd/yyyy hh:mm:ss - SUB-TASK TERM. - TASK = VFI/RTB, PGM = BRNVC007, RECORDS(nnn) UPDATES(nnn)

Explanation:
Issued by BRNVC007 at end of processing. The number of eligible records and the number of records updated is indicated.

User response:
None. Informational message.

CKKRRTB010E mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL QIO OPEN ERROR, RC=> xxx

Explanation:
Issued by BRNVC007 on a BKUPTBL QIO open error. The return code (RC) shows the QIO return code value.

User response:
Retry. If error reoccurs, contact your System Programmer.

CKKRRTB011E mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL QIO RESET ERROR, RC=> xxx

Explanation:
Issued by BRNVC007 on a BKUPTBL QIO reset error. The return code (RC) shows the QIO return code value.

User response:
Retry. If error reoccurs, contact your System Programmer.

CKKRRTB012I mm/dd/yyyy hh:mm:ss - BKUPTBL CLOSE:

Explanation:
Coupled with CKKRRTB013, these messages are issued by BRNVC007 at the end of BKUPTBL processing providing counts for GETs, INSERTs, and so on.

User response:
None. Informational message.

CKKRRTB013I mm/dd/yyyy hh:mm:ss - BKUPTBL STATS GET/KEY GET/SEQ INSERTS UPDATES POINTS DELETES

Explanation:
Coupled with CKKRRTB012, these messages are issued by BRNVC007 at the end of BKUPTBL processing providing counts for GETs, INSERTs, and so on.

User response:
None. Informational message.

CKKRRTB014E mm/dd/yyyy hh:mm:ss - BRNVC007: LOAD ON CKKRS006 FAILED, RC=

Explanation:

Issued by BRNVC007 when the load of module CKKRS006 fails.

User response:
Evaluate return code, contact Technical Support.

CKKRRTB015I mm/dd/yyyy hh:mm:ss - 'INTERVAL TERM. - TASK = VFI/ RTB, PGM = BRNVC007 , RECORDS( ) BKUPTBL UPDATES( )

Explanation:
Issued by BRNVC007 at the end of an SMF interval. The number of records processed and BKUPTBL updates are reported.

User response:
None. Informational message.

CKKRRTB015E mm/dd/yyyy hh:mm:ss- BRNVC007: LOAD ON CKKRS006 FAILED, RC=

Explanation:
Issued by BRNVC007 if the LOAD request for CKKRS006 fails.

User response:
Check the system codes for the meaning of the error

CKKRRTB016 mm/dd/yy hh:mm:ss - BRNVC007: xxxxxxxxxxxxxxxx IN USE JOB(xxxxxxxx) SYSID(xxxxxxxx)

Explanation

Issued by BRNVC007 (RTB) when the program detects that the IBM Z Batch Resiliency backup table enqueue is in use by another process:

Major name: VFYSYS01

Minor name: backup table data set name

When this enqueue is in use, BRNVC007 waits for 5 minutes for the resource to be released. If not freed in that time, BRNVC007 determines what process is holding the resource, reports the job and system where the resource is in use and continues to wait until RTB processing can continue. In addition, if the enqueue is available to BRNVC007, but the open of the backup table fails because another job is holding the SYSDSN enqueue for the table, this message is issued prior to the abend by BRNVC007.

User response:
Resolve who is holding the enqueue and let the job continue.

CKKRRTB016I mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL/FILE IN USE BY JOB(xxxxxxxxx)

Explanation:
Issued by BRNVC007 if the BKUPTBL file is not immediately available. The job holding the file is indicated in the message.

User response:
None. Informational message.

CKKRRTB017I mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL/ ACQUIRED FOR PROCESSING

Explanation:
Issued by BRNVC007 when BKUPTBL file is acquired for processing.

User response:
None. Informational message.

CKKRRTB018I mm/dd/yyyy hh:mm:ss - BRNVC007: NO RECORDS FOUND FOR UPDATE

Explanation:
Issued by BRNVC007 at end of BKUPTBL processing if no updates were processed because no eligible records were found.

User response:
None. Informational message.

CKKRRTB019I mm/dd/yyyy hh:mm:ss - BRNVC007: BKUPTBL/ RELEASED.

Explanation:
Issued by BRNVC007 at end of BKUPTBL processing when the BKUPTBL file is released.

User response:
None. Informational message.

CKKRRTB020I mm/dd/yyyy hh:mm:ss - BRNVC007: ACQUIRING BKUPTBL / FOR PROCESSING

Explanation:
Issued by BRNVC007 when BKUPTBL file processing is initiated and indicates the data set name for the BKUPTBL file.

User response:
None. Informational message.

CKKRRTB021E mm/dd/yyyy hh:mm:ss - BRNVC007: INCORE GETMAIN FAILURE, TERMINATING

Explanation:
Issued by BRNVC007 when an internal GETAMIN failed.

User response:
Increase region size and retry.

CKKRRTB022E mm/dd/yyyy hh:mm:ss - BRNVC007: INCORE TABLE EXCEEDED, TERMINATING

Explanation:
Issued by BRNVC007 when the in core table size is exceeded.

User response:
Contact Technical Support.

CKKRRTB023I **mm/dd/yyyy hh:mm:ss - BRNVC007: VFI/RTB 'BKUPTBL DSN STORAGE xxxxxxxxx - ADDRESS = xxxxxxxxx SUB-POOL/LENGTH = 09/xxxxxxxxx**

Explanation:
Issued by BRNVC007 when internal table storage is obtained. The address, subpool and length are reported.

User response:
None. Informational message.

CKKRRTB024I **mm/dd/yyyy hh:mm:ss - BRNVC007: DIAGNOSTIC PRINT -**

Explanation:
Issued by BRNVC007 when diagnostic mode is active.

User response:
None. Informational message.

CKKRRTB025I **mm/dd/yyyy hh:mm:ss - BRNVC007: TRCE( ) RSCE( ) RC( ) ARG=**

Explanation:
Issued by BRNVC007 when diagnostic mode is active. This message is issued for various trace entries during BKUPTBL processing.

User response:
None. Informational message.

CKKRRTD000I **mm/dd/yyyy hh:mm:ss - MODULE LOADED - xxxxxxxx , ADDRESS = yyyyyyyy**

Explanation
Issued by CKKRC001 at module load, where:

- xxxxxxxx - module name
- yyyyyyyy - load address

User response:
None. Informational message.

CKKRRTD001I **mm/dd/yyyy hh:mm:ss - PARMLIB OPENED - VOLUME = vvvvvv, DSNAME =yyyyyyyyyyyyyyyyy**

Explanation
Issued by CKKRC001 at PARMLIB Open, where:

- vvvvvv - Volume where the PARMLIB data set resides
- yyyyyyyy - data set name

User response:
None. Informational message.

CKKRRTD002I **mm/dd/yyyy hh:mm:ss - RTSRTDCB OBTAINED - ADDRESS = xxxxxx, SUB-POOL/LENGTH = yyzzzzzz**

Explanation
Issued by CKKRC001 when RTD dispatch control block is established, where:

- xxxxxx - control block location
- yy - storage sub pool
- zzzzzz - storage length

User response:
None. Informational message.

CKKRRTD003I **mm/dd/yyyy hh:mm:ss - PARMLIB ACCESS - MEMBER = xxxxxxxx, STATS = VER=yyy,MOD=zzz,MODDATE=yy.ddd,MODTIME=hh:mm**

Explanation
Issued by CKKRC001 at PARMLIB access, where:

- xxxxxxxx - PARMLIB member
- yyy - member version
- zzz - member mod level
- yy.ddd - last update
- hh:mm - last update time

User response:
None. Informational message.

CKKRRTD004I **mm/dd/yyyy hh:mm:ss – RTSSMFPM OBTAINED - ADDRESS = yyyyyy, SUB-POOL/LENGTH = zzllllll**

Explanation
Issued by CKKRC001, where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD005I **mm/dd/yyyy hh:mm:ss – VFIPARMS OBTAINED - ADDRESS = yyyyyy,SUB-POOL/LENGTH = zzllllll**

Explanation

Issued by CKKRC001, where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD006I **mm/dd/yyyy hh:mm:ss - VFIPARMS CONTAINS UN-RECOGNIZED PARAMETERS – SEE SNAP DATASET**

Explanation:
Issued by CKKRC001 when invalid parameter values were specified. The SNAP data set contains diagnostic information.

User response:
None. Informational message.

CKKRRTD007I **mm/dd/yyyy hh:mm:ss – RTSAPTBL OBTAINED - ADDRESS = yyyyyy, SUB-POOL/LENGTH = zzllllll**

Explanation

Issued by CKKRC001, where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD008I **mm/dd/yyyy hh:mm:ss – RTSEXJOB OBTAINED - ADDRESS = yyyyyy, SUB-POOL/LENGTH = zzllllllll**

Explanation

Issued by CKKRC001 where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD009I **mm/dd/yyyy hh:mm:ss – RTSTBLCB OBTAINED - ADDRESS = YYYYY, SUB-POOL/LENGTH = zzllllllll**

Explanation

Issued by CKKRC001 where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD010I **mm/dd/yyyy hh:mm:ss – RTSRTDSB OBTAINED - ADDRESS = YYYYY, SUB-POOL/LENGTH = zzllllllll**

Explanation

Issued by CKKRC001 where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD011I **mm/dd/yyyy hh:mm:ss – INVALID STATEMENT**

Explanation:
Issued by CKKRC001 when invalid parameter statement is found.

User response:
None. Informational message.

CKKRRTD012E **mm/dd/yyyy hh:mm:ss – CKKRTASK CONTAINS UN-RECOGNIZED PARAMETERS - SEE "SNAP" DATASET'**

Explanation:
Issued by CKKRC001 when invalid values are entered in VFISTASK.

User response:
Correct and retry.

CKKRRTD013I **mm/dd/yyyy hh:mm:ss - SMF-TASK (xxxxxxxx) STARTED, TCB zzzzzzzz**

Explanation

Issued by CKKRC001 at the sub task initiation, where:

- xxxxxxxx - TCB address
- yyyyyyyy - sub task name

User response:
None. Informational message.

CKKRRTD014I **mm/dd/yyyy hh:mm:ss - xxxxxxxx (yyyyyyyy) STARTED, TCB zzzzzzzz, RTSSBENT: rrrrrrrr, PORT: ppppp**

Explanation

Issued by CKKRC001 at the sub task initiation, where:

- xxxxxxxx - sub task name
- yyyyyyyy - sub task program
- zzzzzzzz - TCB address

The final two parameters, RTSSBENT and PORT are not issued in all cases; when they do appear, these are the meanings:

- rrrrrrrr - sub task control block address
- ppppp - port number

User response:
None. Informational message.

CKKRRTD015E mm/dd/yyyy hh:mm:ss - SUB-TASK TERMINATION - TCB: XXXXXXX COMPLETION CODE: yyyyyyyy (ZZZZZZZZZZ:TTTTTTTT)

Explanation

Issued by CKKRC001 at the end of the sub task, where:

- xxxxxxxx - TCB address
- yyyyyyyy - sub task completion code
- ZZZZZZZZZ - sub task name
- TTTTTTTTT - sub task program id

When the sub task terminates abnormally <===*EXC is appended to the text for easy identification.

User response:
Evaluate the return code and contact Technical Support.

CKKRRTD017I mm/dd/yyyy hh:mm:ss - RTSSRARY OBTAINED - ADDRESS = xxxxxxxx, SUB-POOL/LENGTH = xxxxxxxx

Explanation:
Issued by CKKRC001 after obtaining storage for the Scheduler Schedule Array. The storage address and sub-pool and length are reported.

User response:
None. Informational message.

CKKRRTD018I mm/dd/yyyy hh:mm:ss – QIO CYLTBL OBTAINED - ADDRESS = YYYY, SUB-POOL/LENGTH = zzllllll

Explanation

Issued by CKKRC001, where:

- yyyyyyyy - load address
- zz - storage sub pool
- llllll - storage length

User response:
None. Informational message.

CKKRRTD019I mm/dd/yyyy hh:mm:ss - PARMLIB & DIRECTORY CLOSED

Explanation:
Issued by CKKRC001 during process termination when the PARMLIB is closed.

User response:
None. Informational message.

CKKRRTD020I mm/dd/yyyy hh:mm:ss - ADDRESS SPACE QUIESING

Explanation:
Issued by CKKRC001 at normal process termination

User response:
None. Informational message.

CKKRRTD021I mm/dd/yyyy hh:mm:ss – WAITING FOR TERMINATION OF SUB-TASK: xxxxxxxx

Explanation:
Issued by CKKRC001 at process termination when waiting for a sub task to complete.

User response:
None. Informational message.

CKKRRTD022I mm/dd/yyyy hh:mm:ss - ADDRESS SPACE TERMINATING ABNORMALLY

Explanation:
Issued by CKKRC001 at abnormal process termination; note completion code Equals 999.

User response:
None. Informational message.

CKKRRTD025I mm/dd/yyyy hh:mm:ss - CKKRC001: RT/SMF RELEASE X.X.X

Explanation:
Issued by CKKRC001 to document the release and mod level of this execution of RT/SMF.

User response:
None. Informational message.

CKKRS00903I mm/dd/yyyy hh:mm:ss - DD( ) D/S CLOSE - n,nnn,nnn RECORDS FROM:yyyy.ddd hh:mm:ss.th TO:yyyy.ddd hh:mm:ss.th

Explanation:

Issued by CKKRS009 at the end of the indicated SMF data set. The range of record time/date stamps are listed.

User response:
None. Informational message.

CKKRS00904I mm/dd/yyyy hh:mm:ss - DD( ) FILE RE-STARTED AT: yyyy.ddd hh:mm:ss.th

Explanation:
Issued by CKKRS009 when a restart is performed on an SMF data set. The date/time of the restart is listed.

User response:
None. Informational message.

CKKRS00905I mm/dd/yy hh:mm:ss - CKPT INIT| UPDT ERROR: x QIO RC: 00000000

Explanation:
Issued on initialization (INIT) and update (UPDT) checkpoint errors.

User response:
Evaluate the error code and the return code for the cause.

CKKRS00906I mm/dd/yyyy hh:mm:ss - DD( ) SPIN DONE IN(yyyy.ddd hh:mm:ss.th) LAST(yyyy.ddd hh:mm:ss.th) COUNT: n,nnn,nnn

Explanation:
Issued by CKKRS009 when a file spin is completed. This occurs when the date/time stamp of the record just read is prior to the last processed SMF record. RT/SMF reads through this SMF file until the data/time stamp is greater than or equal to the last valid processed record.

User response:
None. Informational message.

CKKRS00907I mm/dd/yyyy hh:mm:ss - DD( ) FILE SPIN FROM: yyyy.ddd hh:mm:ss.th TO: yyyy.ddd' hh:mm:ss.th

Explanation:
Issued by CKKRS009 when a file spin is required. The date/time begin and end points are listed.

User response:
None. Informational message.

CKKRS00908I mm/dd/yyyy hh:mm:ss - RT/SMF CHECKPOINT SCHEDULED

Explanation:
Issued by CKKRS009 when a RT/SMF checkpoint is initiated. This is currently done in 10 minute SMF intervals.

User response:
None. Informational message.

CKKRS00910I - mm/dd/yy hh:mm:ss - I/O ERROR OCCURRED, EOF FORCED ON FILE, xxxxxxxx/ xxxxxxxxxxxxxxxxxxxxx

Explanation:
This message is printed in the RTDPRINT file for RT/SMF. It identifies the file that had the I/O by DDNAME and data set name. It is informational, and RT/SMF continues processing.

User response:
None. Informational message.

CKKRS00911I SMFDATA? I/O ERROR xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Explanation:
This WTO message appears in the job log. The affected SMF data file is identified. The variable portion passes the operating system error data to the SYNAD routine. It usually identifies the job, job step, device and error description.

System action:
This message is informational; RT/SMF forces EOF processing and continue.

User response:
None. Informational message.

CKKRS00909I mm/dd/yyyy hh:mm:ss - RT/SMF CHECKPOINT COMPLETED SMF DATE/TIME: yyyy.ddd hh:mm:ss.th

Explanation:
Issued by CKKRS009 when the checkpoint is completed. The end SMF date/time is listed.

User response:
None. Informational message.

CKKRS09901I mm/dd/yyyy hh:mm:ss - Text: MODULE LOADED - CKKRMFCB, ADDRESS =

Explanation

Issued during RT/SMF initialization. Documents location of CKKRMFCB control block.

CHECKPOINT ERROR X Issued when an error occurs processing the checkpoint file Issued by CKKRS009, the SMF read manager.

User response:
None. Informational message.

CKKRS09902E mm/dd/yyyy hh:mm:ss - OPEN FAILED, DD:

Explanation:

Issued by CKKRS009 when an open error occurs on one of the SMFDATA data sets. The DD in error is identified.

User response:
Determine the cause of the open error from the JOBLOG, and retry.

CKKRT99902I **mm/dd/yy hh:mm:ss - RT/SMF SUBTASK(XXXXXXX)/XXXXXXXX) STG BELOW: XXXXXXXX M STG ABOVE: XXXXXXXX M**

Explanation:
This message is issued by the CKKRT999 subtask to document the storage usage for the RTA subtasks.

User response:
None. Informational message.

CKKRVFI010I **mm/dd/yyyy hh:mm:ss - APP( ) GET/KEY GET/SEQ INSERTS UPDATES POINTS DELETES RTSVFI010I - mm/dd/yyyy hh:mm:ss - COLLISIONS**

Explanation:
Issued by BRNVC010 at the end of processing to report file statistics for specific applications or all applications.

User response:
None. Informational message.

VFI006000I **mm/dd/yyyy hh:mm:ss - OAT OBTAINED FOR: xxxxxxxxx AT: xxxxxxxxx**

Explanation:
Issued by BRNVC006 when an OAT (open application table) entry is obtained. The application and storage location are listed.

User response:
None. Informational message.

VFI006002I **mm/dd/yyyy hh:mm:ss - xxxxxxxxx CYCLE: yyyy.ddd-00 - OPENED ON SYSTEM: xxxx BY: xxxxxxxxx yyyy.ddd hh:mm:ss.th ( )**

Explanation:
Issued by BRNVC006 when an application is opened. The application, cycle date time, system location, job name, date and time and why (JOB, TIME, ...) are listed.

User response:
None. Informational message.

VFI006003I **mm/dd/yyyy hh:mm:ss - xxxxxxxxx - BASE: n,nnn,nnn GETS n,nnn,nnn NRFS n,nnn,nnn ADDS n,nnn,nnn MODS**

Explanation:
Issued by BRNVC006 after INVENTRY processing indicating the number of gets, record not found, adds and updates for base records.

User response:
None. Informational message.

VFI006004I **mm/dd/yyyy hh:mm:ss - xxxxxxxxx - COMP: n,nnn,nnn GETS n,nnn,nnn NRFS n,nnn,nnn ADDS n,nnn,nnn MODS**

Explanation:
Issued by BRNVC006 after INVENTRY processing indicating the number of gets, record not found, adds and updates for component records.

User response:
None. Informational message.

VFI006005E **mm/dd/yyyy hh:mm:ss - xxxxxxxxx NOT AVAILABLE, PROCESS TERMINATING**

Explanation:
Issued by BRNVC006 when the listed program is not available. The programs loaded are: BRNVC009 (GDG count), BRNVC010 (Global status processor) or BRNVC016 (checkpoint module).

User response:
Ensure that the listed module is available in STEPLIB and rerun the job.

VFI006006E **mm/dd/yyyy hh:mm:ss - xxxxxxxxx NOT AVAILABLE, PROCESS TERMINATING**

Explanation:
Issued by BRNVC006 when the listed program is not available. The programs loaded are: BRNVC009 (GDG count), BRNVC010 (Global status processor) or BRNVC016 (checkpoint module).

User response:
Ensure that the listed module is available in STEPLIB and rerun the job.

VFI010000E **mm/dd/yyyy hh:mm:ss - INVENTRY GLOBAL STATUS PROCESSOR ABENDING, CODE:**

Explanation

Issued by BRNVC010 (global status processor). The reason for the abend is listed.

1. Load fail (CKKRC002/ BRNVC016) code
2. INVENTRY POINT error code
3. INVENTRY GETNEXT error code
4. INVENTRY OPEN error code
5. INVENTRY POINT error (C rec) code

6. INVENTRY GETNEXT error (C rec) code
7. INVENTRY PUT error

User response:
Evaluate the error code and contact Technical Support.

VFI010001I **mm/dd/yyyy hh:mm:ss - BRNVC010: INVENTRY GLOBAL STATUS PROCESSOR xxxxxxxxx APPL(aaaaaaa)**

Explanation:
Issued by BRNVC010 at the start and end of the process. The value of xxxxxxxx is STARTED or COMPLETED and the application (*aaaaaaa*) is listed.

User response:
None. Informational message.

VFI016001I **mm/dd/yyyy hh:mm:ss - APPL:xxxxxxxxx CYCLE: 0CYYDDD ## - X'..' - JOBNAME YYYYDDD TTTTTTTT - n,nnn,nnn DSNS TI: n,nnn,nnn**

Explanation:
Issued by BRNVC016 , the RT/SMF ODTE checkpoint processor. This message documents the information contained in the internal OAT and ODTEs as well as the number of records this interval for this application.

User response:
None. Informational message.

VFI016002I **mm/dd/yyyy hh:mm:ss - OAT CHECK POINT BYPASSED**

Explanation:
Issued by BRNVC016 when there is no data to checkpoint for this interval; that is, no changes in the OAT and ODTEs.

User response:
None. Informational message.

VFI016003I **mm/dd/yyyy hh:mm:ss - APPL: NULL ODTE APPLS TI: 999999999**

Explanation:
Issued by BRNVC016 to report the total records processed this interval across applications.

User response:
None. Informational message.

BRNVC00102I **mm/dd/yyyy hh:mm:ss - DRVFI SMF WRAPPER RECORD TYPE INVALID, RTYP( )**

Explanation:
Issued by BRNVC001 . Indicates user SMF record type is invalid. The record is bypassed.

User response:
None. Informational message.

BRNVC00104I **mm/dd/yyyy hh:mm:ss - CUSTJOB USER EXIT(BRNUXIT1) ACTIVE**

Explanation:
Issued by BRNVC001 if the custom JOB exit is active.

User response:
None. Informational message.

BRNVC00105E **mm/dd/yyyy hh:mm:ss - DYNAMIC ALLOCATION ERROR DDNAME: R15( ) S99RSC( )**

Explanation:
Issued by BRNVC001 when a dynamic allocation occurs on either the AUDIT or HISTORY file. The system retries over a 5 minute period to acquire the respective file prior to this message. The SMF processor terminates with a return code of 4 indicating an error condition.

User response:
Ensure that the AUDIT/HISTORY file is available and rerun the job.

BRNVC00606E **mm/dd/yyyy hh:mm:ss - BRNVC006 xxxxxxxxxxxxx :R14( ) R15( ) R0( ) REASON CD/C21VS/ $VSAMIO**

Explanation:
Issued by BRNVC006 when there is a VSAM error on the INVENTRY file. The text portion (xxxxxxxxxxxx) contains an error description. The contents of register 14 and 15 are listed. Register 0 contains the VSAM reason code and the r0 value returned by C21VS macro.

User response:
Check the codes returned by C21VS or the VSAM feedback codes as appropriate, correct the error and retry. In many cases there are VSAM messages and codes in the JOB log that help determine the error.

BRNVC009023 *** mm/dd/yyyy hh:mm:ss – B GET ERR R15/R0/KEY: rrrrrrrr/rrrrrrrr/ xxxxxxxxxxxxxxxx**

Explanation:
Issued by BRNVC009 when a GET for the associated 'B' record (GDG base) fails. The contents of register 15 (the return code) and register 0 (the VSAM feedback code) are reported along with the 'B' record key.

System action:
This message is issued up to 20 times at which point additional messages are suppressed. Processing for other GDG files continues.

User response:
None. Informational message.

BRNVC01003I **mm/dd/yyyy hh:mm:ss - RECORDS WRITTEN TO xxxxxxxxx**

FILE:' nn,nnn,nnn FROM: yyyy.ddd hh:mm:ss.th TO: yyyy.ddd hh:mm:ss.th

Explanation:
Issued by BRNVC001 after writing records to AUDIT or HISTORY files. The xxxxxxxxx indicates HISTORY or AUDIT.

User response:
None. Informational message.

BRNVC01300I mm/dd/yyyy hh:mm:ss - OAT OBTAINED FOR: xxxxxxxxx AT: xxxxxxxxx

Explanation:
Issued by BRNVC013 when an open application table entry is obtained during a warm start. The data for RT/SMF restart comes from the ODTCKPT data set which contains the OAT and ODTE entries from the last RT/SMF checkpoint.

User response:
None. Informational message.

BRNVC01301I mm/dd/yyyy hh:mm:ss - OAT STARTUP: xxxx

Explanation:
Issued by BRNVC013 to identify the type of address space start - COLD or WARM.

User response:
None. Informational message.

BRNVC01302E mm/dd/yyyy hh:mm:ss - TABLE LOAD FAILED: xxxxxxxxx

Explanation:
Issued by BRNVC013 when there is a table load failure. The tables loaded/reloaded by BRNVC013 are the APPTABLE and APPTABLX.

User response:
Contact your System Programmer with the contents of the message.

BRNVC01303I mm/dd/yyyy hh:mm:ss - OAT xxxx START COMPLETED

Explanation:
Issued by BRNVC013 when the warm or cold start is complete.

User response:
None. Informational message.

BRNVC01304E mm/dd/yyyy hh:mm:ss - APPTABLE ENTRY NOT FOUND

Explanation:
Issued by BRNVC013 when the APPTABLE entry cannot be located. This could only occur if the APPTABLE is not defined in VFITABLS USERMAC. (member).

User response:
Correct and retry.

BRNVC01305E mm/dd/yyyy hh:mm:ss - VFIOANDX NOT FOUND

Explanation:
Issued by BRNVC013 if the internal Open Application table is not found.

User response:
This should not occur. Restart the region and if the error persists, contact Technical Support.

BRNVC01306E mm/dd/yyyy hh:mm:ss - VFIOANDX COUNT IS ZERO

Explanation:
Issued by BRNVC013 when the application count stored in the Open Application Table control block is 0.

System action:
The region terminates.

User response:
Restart and contact Technical Support.

BRNVC01307E mm/dd/yyyy hh:mm:ss - VFIOANDX SLOTS EXCEEDED

Explanation:
Issued by BRNVC013 when the number of OATs recovered exceeds the number established at startup.

System action:
The region terminates.

User response:
Restart and contact Technical Support.

BRNVC01308I mm/dd/yyyy hh:mm:ss - TOTAL OAT(S) RECOVERED: 9999999

Explanation:
Issued by BRNVC013 to document the number of Open Application Table (OAT) entries recovered.

User response:
None. Informational message.

BRNVC01309I mm/dd/yyyy hh:mm:ss - TOTAL ODTE(S) RECOVERED FOR APPLICATION: (xxxx) INV RECS RECOVERED (nnnn)

Explanation:
Issued by BRNVC013 to document the Open Dataset Table Entries (ODTE) and INVENTRY records recovered during restart.

User response:
None. Informational message.

BRNVC01310E mm/dd/yyyy hh:mm:ss - APPTABLX ENTRY NOT FOUND

Explanation:

Issued by BRNVC013 when the APPTABLX entry is not found. There should be an entry in the VFITABLS USERMAC member for APPTABLX.

User response:
Contact your System Programmer with the contents of the message.

BRNVC01310I mm/dd/yyyy hh:mm:ss - BKUPTBL CLOSE:

Explanation

Issued by BRNVC013 at the end of BKUPTBL load processing providing the following counts:

- BKUPTBL STATS GET/KEY GET/SEQ INSERTS UPDATES POINTS DELETES

User response:
None. Informational message.

BRNVC01311I mm/dd/yyyy hh:mm:ss - APPLICATION DELETED, OAT/ODTE(S) NOT RECOVERED

Explanation:
Issued by BRNVC013 when there is an OAT in the ODTCKPT data set, but the application does not exist in the APPTABLE. The OAT and associated ODTEs are not recovered.

User response:
None. Informational message.

BRNVC01312E mm/dd/yyyy hh:mm:ss - BRNVC013 xxxxxxxxxxxx :R14( ) R15( ) R0( ) REASON CD/C21VS/$VSAMIO

Explanation:
Issued by BRNVC006 when there is a VSAM error on the INVENTRY file. The text portion (xxxxxxxxxxx) contains an error description. The contents of register 14 and 15 are listed. Register 0 contains the VSAM reason code and the R0 value returned by C21VS macro.

User response:
Check the codes returned by C21VS or the VSAM feedback codes as appropriate, correct the error and retry. In many cases there are VSAM messages and codes in the JOB log that help determine the error.

BRNVC01400I mm/dd/yyyy hh:mm:ss - *message_text*

Explanation

Issued by BRNVC014 when a system display is requested. The *message_text* depends on the report requested:

- DRVFI RT/SMF: REPORT - Header, identifies report requested
- DRVFI RT/SMF: REPORT COMPLETE - Trailer, identifies end of report
- DRVFI RT/SMF SUBTASK NOT FOUND - Display subtask report, request subtask not defined
- DRVFI RT/SMF SUBTASK HAS NO COUNTERS - Display subtask report, no record types counted
- DRVFI RT/SMF STORAGE REPORT ERROR: VSMLIST RC=00000000 - Display storage report, storage map macro failed see return code
- DRVFI RT/SMF STORAGE REPORT ERROR: TCB=00000000 - Display storage report, storage map macro for identified subtask failed
- DRVFI RT/SMF STORAGE REPORT ERROR: SUBPOOL DESCRIPTOR - Display storage report, subpool error
- DRVFI RT/SMF STORAGE REPORT ERROR: SUBTASK NOT FOUND - Display storage report, subtask requested not found
- DRVFI RT/SMF STORAGE REPORT, TOTAL THIS SUBTASK - Display storage report, total storage reported
- DRVFI RT/SMF STORAGE REPORT, TOTAL ALL SUBTASKS - Display storage report, total storage used

User response

The displays accepted by BRNVC014 can be entered via CKKRC008 (JCL sample follows) or the TSO command processor. The available displays are:

- DISPLAY OATS - Displays Open Application table
- DISPLAY SUBTASKS - Displays all subtasks
- DISPLAY SUBTASK=VFI/RTB - Displays a specific subtask; for example RTB
- DISPLAY SUBTASK=BRNVC007 - Displays a specific subtask; for example BRNVC007
- DISPLAY STORAGE - Display address space storage
- DISPLAY STORAGE=BRNVC007 - Display storage for specific subtask; For example: BRNVC007
- DISPLAY TABLES - Display tables; For example: APPTABLE, EXCLJOB, EXCLDSN
- DISPLAY SMF - Display SMF information; For example: DSN, current record date/time

These commands are scheduled by the command processor subtask CKKRC004 which must be specified in the VFISTASK table.

Sample CKKRC008 JCL:

```
//jobcard
//STEP1 EXEC PGM=CKKRC008,
//           PARM='DISPLAY TABLES'
//*          PARM='REFRESH APPTABLE'
//*                1234567812345678
//STEPLIB  DD DSN=DRVFI_library,DISP=SHR
```

```
//QUEUE    DD DSN=RTCMDQ_dataset,DISP=SHR
//RTDPRINT DD SYSOUT=*
```

BRNVC01500I mm/dd/yyyy hh:mm:ss - DELETED APPLICATION OAT/ODTE FREE COMPLETE

Explanation:
Issued by BRNVC015 after an APPTABLE reload. The old OAT and ODTEs are freed by BRNVC015 which is called by BRNVC006 .

User response:
None. Informational message.

BRNVC01501I mm/dd/yyyy hh:mm:ss - OAT FOR APPLICATION: FREED, ADDRESS = xxxxxxxxx SUB-POOL/LENGTH = xxxxxxxxx

Explanation:
Issued by BRNVC015 after an OAT free is complete. The application, storage address and subpool/length are reported.

User response:
None. Informational message.

BRNVC01502I mm/dd/yyyy hh:mm:ss - VFIODTFL FREED - ADDRESS = xxxxxxxxx SUB-POOL/LENGTH = xxxxxxxxx

Explanation:
Issued by BRNVC015 after an ODTE is freed. The storage address, subpool and length are reported.

User response:
None. Informational message.

BRNVC01503I mm/dd/yyyy hh:mm:ss - TOTAL ODTE(S) RELEASED FOR APPLICATION: (9999999)

Explanation:
Issued by BRNVC015 to report the total number of Open Dataset Table Entries were released for the identified application.

User response:
None. Informational message.

BRNVC01504I mm/dd/yyyy hh:mm:ss - TOTAL OAT(S) RELEASED: 99999999

Explanation:
Issued by BRNVC015 to report the total number of Open Application table entries released.

User response:
None. Informational message.

BRNVC01505I mm/dd/yyyy hh:mm:ss - LRU FOR APPLICATION: FREED, ADDRESS = xxxxxxxxx SUB-POOL/LENGTH = xxxxxxxxx

Explanation:
Issued by BRNVC015 to report the Least Recently Used (LRU) table for application ODTEs was freed. The address and subpool and length are reported. The LRU is anchored off the internal OAT entry and is used for performance. When the new OAT was obtained a new LRU was also established.

User response:
None. Informational message.

BRNVC02001I mm/dd/yyyy hh:mm:ss - :R15(xxxxxxxxx) FEEDBACK(xxxxxxxxx)

Explanation

Issued by BRNVC020 during Keyed History RT/SMF processing. The text portion contains one of the following:

- LSR POOL 2 OPEN FAIL Usually a storage issue, increase region.
- LSR POOL 3 OPEN FAIL Usually a storage issue, increase region.
- KEYDHIST OPEN FAILED The KEYDHIST open failed.
- KEYJHIST OPEN FAILED The KEYJHIST open failed.
- KEYDHIST PUT ERROR A PUT request failed on the KEYDHIST file.
- KEYJHIST PUT ERROR A PUT request failed on the KEYJHIST file.

User response:
Check the return code and feedback, correct the error and retry.

BRNVC02002I mm/dd/yyyy hh:mm:ss - BRNVC020: KEYED HISTORY FILES xxxxx

Explanation:
Issued by BRNVC020 when the Keyed History VSAM files are opened or closed as indicated in the message.

User response:
None. Informational message.

BRNVC02003I mm/dd/yyyy hh:mm:ss - BRNVC020: xxxxxxxxx PUTS(n,nnn,nnn,nnn) DUPLICATES(n,nnn,nnn,nnn)

Explanation:
Issued by BRNVC020 at the end of processing to document for the indicated Keyed History file (KEYDHIST or KEYJHIST) the number of puts and duplicates.

User response:
None. Informational message.

BRNVC02004I mm/dd/yyyy hh:mm:ss - LSR STATISTICS: #BUFFERS CI SIZE READS LOOKASIDES NONUSER WRTS USER WRTS

Explanation:
Issued by BRNVC020 to report LSR usage for pools 2 and 3 built to support the Keyed History files.

User response:
None. Informational message.

BRNVC02101I mm/dd/yyyy hh:mm:ss - *INVENTRY_message_text*

Explanation
Issued by BRNVC021 , the INVENTRY file manager. A multi-line message with *INVENTRY_message_text* as follows:

```
INVENTRY CLOSE:

GET/KEY GET/SEQ INSERTS UPDATES POINTS
DELETES

9999999 999999 999999 999999 999999
9999999

COLLISIONS

9999999
```

These messages report the number of various file calls against the INVENTRY file. The number of collisions refers to concurrent requests against the INVENTRY file as access is serialized.

User response:
None. Informational message.

BRNVC02102I mm/dd/yyyy hh:mm:ss - R15: (xxxxxxxxx) R0: (xxxxxxxxx)

Explanation:
Issued by BRNVC021 during INVENTRY file Open/Close and LSR Pool Build. In the message, R15 contains the return code and, when applicable, the R0 field contains the reason code or feedback code.

User response
The text portion contains one of the following:

BLDVRP FAILED
Evaluate the return code for the BLDVRP, correct the error and retry. Only the return code is meaningful. The most common error, R15=8, implies insufficient virtual storage space to satisfy the request.

LSR STAT ERR
An attempt to gather LSR statistics using the SHOWCB macro failed.

INVENTRY FAILURE
Evaluate the return code in R15 and feedback in R0 for the OPEN macro; correct the error and retry.

INVENTRY LSR/CISZ ER
The CI Size for the Index and data portion is invalid. The default sizes are: Index 1024, data up to 32768.

The macros are documented in IBM *z/OS DFSMS Macro Instructions for Data Sets*.

BRNVC02103I mm/dd/yyyy hh:mm:ss - LSR STATISTICS: #buffers ci size reads lookasides nonuser wrts user wrts

Explanation:
Issued by BRNVC021 to report LSR usage for pools 2 and 3 built to support the INVENTRY file.

User response:
None. Informational message.

BRNVC02104I mm/dd/yyyy hh:mm:ss - BRNVC021: INVENTRY FILE xxxxxxxxx

Explanation:
Issued by BRNVC021 when the file is opened or closed as indicated.

User response:
None. Informational message.

BRNVC02105I mm/dd/yy hh:mm:ss - INVENTRY FILE CLOSE SUBTASK xxxxxxxx WAITING ON SUBTASK: xxxxxxxx - INVENTRY FILE CLOSE DELAYED

Explanation:
This message is issued by BRNVC021 when the INVENTRY file is being closed. Since multiple subtasks use the INVENTRY file it is necessary to ensure all are finished with it before the file manager closes it. The identified subtask is still actively using the INVENTRY file.

User response:
None. Informational message.

BRNVC06001I mm/dd/yyyy hh:mm:ss - INTERVAL TERM. - TASK xxxxxxxxx, PGM = BRNVC006, R/T nnn COUNTS = nnn,nnn,nnn

Explanation:
Issued by BRNVC006 at the end of interval processing. The SMF record type and count are listed.

User response:
None. Informational message.

BRNVL00101E mm/dd/yyyy hh:mm:ss - APPTABLE ERROR -

Explanation:

Issued by BRNVL001 during APPTABLE load processing. It identifies an APPTABLE record in error.

System action:
The record is bypassed.

User response:
Take note of the contents of the errro and contact your System Programmer.

BRNVL00102I mm/dd/yyyy hh:mm:ss - APPTABLE LOADED - x,xxx,xxx INPUT, x,xxx,xxx ERROR, x,xxx,xxx JOBS, X,xxx,xxxx APPS

Explanation:
Issued by BRNVL001 after APPTABLE load processing. It reports the number of records processed, the number of records found in error, the number of jobs, and the number of APPS defined.

User response:
None. Informational message.

BRNVL00103I mm/dd/yyyy hh:mm:ss - MAXIMUM OPEN APPLICATIONS: nnnn,nnn

Explanation:
Issued by BRNVL001 after APPTABLE load processing to document the number of available open applications.

User response:
None. Informational message.

BRNVL00104I mm/dd/yyyy hh:mm:ss - VFIOANDX OBTAINED - ADDRESS = xxxxxxxxx SUB-POOL/LENGTH = xxxxxxxxx

Explanation:
Issued by BRNVL001 when the Open Application Table (VFIOANDX) is obtained. It documents the storage address, subpool and length.

User response:
None. Informational message.

BRNVL00105I mm/dd/yyyy hh:mm:ss - BRNVL001: APPTABLE/APPTABLX LOAD STARTED

Explanation:
Issued by BRNVL001 when Application table load starts

User response:
None. Informational message.

BRNVL00106I mm/dd/yyyy hh:mm:ss - BRNVL001: xxxxxxxxx LOAD COMPLETE

Explanation:
Issued by BRNVL001 . APPTABLE load is complete.

User response:
None. Informational message.

BRNVL00107I mm/dd/yyyy hh:mm:ss - VFIODTFL DEFINED - ADDRESS = xxxxxxxxx' SUB-POOL/LENGTH = xxxxxxxxx

Explanation:
Issued by BRNVL001 during APPTABLE load. This documents storage obtained for the Open Data set table (ODTEs), the storage address and subpool and length.

User response:
None. Informational message.

BRNVL00201E mm/dd/yyyy hh:mm:ss - xxxxxxxx ERROR NN REC:

Explanation:
Issued by BRNVL002 while loading the IBM Z Batch Resiliency table 'xxxxxxxx' when an error is found in the input record. The record and the reason code are reported.

User response:
Take note of the contents of the error and contact your System Programmer.

BRNVL00201I mm/dd/yyyy hh:mm:ss - xxxxxxxx TABLE LOADER STARTED

Explanation:
Issued by BRNVL002 when the IBM Z Batch Resiliency table 'xxxxxxxx' table loader starts

User response:
None. Informational message.

BRNVL00202I mm/dd/yyyy hh:mm:ss - xxxxxxxx LOADED - x,xxx,xxx INPUT, x,xxx,xxx ERROR, x,xxx,xxx OUTPUT

Explanation:
Issued by BRNVL002 when the load is complete. It documents the number of records processed, the number of records rejected or in error and the number output for table 'xxxxxxxx'.

User response:
None. Informational message.

BRNVL00203E mm/dd/yyyy hh:mm:ss - xxxxxxxx LOAD FAILURE, REASON xxxx

Explanation:
Issued by BRNVL002 when there is a load failure for table 'xxxxxxxx'. The reason code indicates the failure: OPEN IN ERRRC=8 GET IN ERRRC=12 QIO PUT ERRRC=16

User response:
Correct the error and retry.

BRNVL00301E **mm/dd/yyyy hh:mm:ss - EXCLJOB ERROR NN REC:**

Explanation:
Issued by BRNVL003 while loading the exclude JOB table when an error is found in the input record. The record and the reason code are reported.

User response:
Correct the error and retry.

BRNVL00301I **mm/dd/yyyy hh:mm:ss - EXCLJOB TABLE LOADER STARTED**

Explanation:
Issued by BRNVL003 when the exclude JOB table loader starts.

User response:
None. Informational message.

BRNVL00302I **mm/dd/yyyy hh:mm:ss - EXCLJOB LOADED - x,xxx,xxx INPUT, x,xxx,xxx ERROR, x,xxx,xxx OUTPUT**

Explanation:
Issued by BRNVL003 when the load is complete. It documents the number of records processed, the number of records rejected or in error and the number output for the table.

User response:
None. Informational message.

BRNVL00303E **mm/dd/yyyy hh:mm:ss - EXCLJOB LOAD FAILURE, REASON xxxx**

Explanation:
Issued by BRNVL003 when there is a load failure. The reason code indicates the failure: OPEN IN ERRRC=8 GET IN ERRRC=12 QIO PUT ERRRC=16

User response:
Correct error and retry.

BRNVL01301I **mm/dd/yyyy hh:mm:ss - BRNVL013: BKUPTBL LOAD/ RELOAD STARTED**

Explanation:
Issued by BRNVL013 (BKUPTBL load program) at table load initiation.

User response:
None. Informational message.

REQ$TASK Messages

This section documents the REQ$TASK messages. The message prefixes are:

- BRNVCnnnnnI - Informational
- BRNVCnnnnnE - Error

BRNVC06801I **mm/dd/yy hh:mm:ss - INTERVAL TERM. - TASK = REQ$TASK, PGM = BRNVC068 , R/T = 99, COUNTS = 9999999**

Explanation:
This message is issued at the RT/SMF interval. It reports the number of record types processed during the interval.

User response:
None. Informational message.

BRNVC06803I **mm/dd/yy hh:mm:ss - BRNVC068: ERROR xxxxxxxxxxxxxx RTA/REQ TERMINATED**

Explanation

This message is issued for file error conditions. The error reported is as follows:

R15(XXXXXXXX) REQ$FILE OPEN FAILED

System action:
The address space is terminated with an ABEND 68.

User response:
Correct the error, and restart.

BRNVC06804I **mm/dd/yy hh:mm:ss - BRNVC068: ALLOCATING REQ$FILE FILE: xxxxxxxxxxxxxxxxxxxxxxxxx**

Explanation:
This message is issued to identify the allocated REQ $FILE. The file is only allocated if there is a record to write. The file is closed and freed at the 10 minuted interval.

User response:
None. Informational message.

BRNVC06805E *** mm/dd/yy hh:mm:ss - BRNVC068: VFIPARM ERROR- xxxxxxxxxxxxxxxxxxxxxxxxxx REQ FUNCTION TERMINATED**

Explanation

This message is issued for parameter errors. The errors are one of the following:

REQSEC ERROR
REQPRI ERROR

MEMBER NOT FOUND
REQPFX LENGTH ERROR

User response:
Correct the error, and restart

BRNVC06808I **mm/dd/yy hh:mm:ss - BRNVC068: REQ$FILE RECORDS WRITTEN: xxxxxxxx**

Explanation:
This message is issued at the interval to report the number of records processed.

User response:
None. Informational message.

BRNVC06809I **mm/dd/yy hh:mm:ss - BRNVC068: VFIPARM REQ MEMBER: RTREQPRM**

Explanation:
This message is issued to report the parameter member to support REQ$TASK. If you want to specify a different member, you can specify that in the VFIPARMS member (REQPRM=)

User response:
None. Informational message.

BRNVC06810I **mm/dd/yy hh:mm:ss - BRNVC068: NO REQ$ RECORDS THIS INTERVAL**

Explanation:
This message is issued at the interval when no records were created in the interval.

User response:
None. Informational message.

BRNVC06811I **mm/dd/yy hh:mm:ss - BRNVC068: REQ$FILE WORKAREA GETMAIN FAILED, RC(xxxx) FUNCTION TERMINATED**

Explanation:
This message is issued when the GETMAIN for the REQ$FILE work area failed.

User response:
Correct the error, and restart.

BRNVC06812I **mm/dd/yy hh:mm:ss - BRNVC068: CLOSE/FREE REQ$FILE FILE: xxxxxxxxxxxxxxxxxxxx**

Explanation:
This message is issued when the REQ$FILE is closed. The file is identified.

User response:
None. Informational message.

BRNVC06813I **mm/dd/yy hh:mm:ss - BRNVC068: CYLINDER ALLOCATION REDUCED TO 1000**

Explanation:
This message is issued when the primary or secondary cylinder allocation specified in RTREQPRM is greater than 1000.

User response:
None. Informational message.

BRNVC06806E **mm/dd/yy hh:mm:ss - BRNVC068: DYNAMIC ALLOCATION ERROR - R15(xxxx) S99ERROR(xxxxxxxx S99INFO(xxxxxxxx) FUNCTION TERMINATED**

Explanation:
This message is issued when there is a dynamic allocation error. The error code and reason code are reported.

User response:
Correct the error, and restart.

RPTMAINT Messages

This section documents the RPTMAINT messages. The message prefixes are:

- VFIRMTnnnI - Informational
- VFIRMTnnnW - Warning
- VFIRMTnnnE - Error

IZBR **REPORTS($INDEX) INUSE BY <jobname>. RETRY Y or N.**

Explanation:
RPTMAINT sends this message to the operator console after repeated attempts to enqueue the REPORTS($INDEX) member. The message alerts the operator to the job using the resource.

User response

(a) Ensure that no other job is using the file REPORTS($INDEX). Reply ##,Y at the console, rerun the job.

(b) Reply ##,N at the console to end the job.

VFIRMT090I **CHECK REPORTS ($INDEX) STARTS**

Explanation:
RPTMAINT program checks for the existence of REPORTS($INDEX). If not found, the $INDEX member is created and loaded with a dummy detail record. Also, a corresponding REPORTS(VFIDUMMY) member is created.

User response:
None. Informational message.

VFIRMT100I MAINTENANCE STARTS

Explanation:
RPTMAINT program processing has begun.

User response:
None. Informational message.

VFIRMT105I REPORTS($INDEX) NOT FOUND, INITIALIZING WITH VFIDUMMY/ #0000000 RECORD

Explanation:
RPTMAINT program checks for the existence of REPORTS($INDEX). If not found, the $INDEX member is created and loaded with a dummy detail record. Also, a corresponding member REPORTS(VFIDUMMY) is created.

User response:
None. Informational message.

VFIRMT106I ADD <member> <record|member> to REPORTS <$INDEX|PDS>

Explanation:
This informational message reports on records added to the REPORTS($INDEX) or PDS members added to the reports PDS.

User response:
None. Informational message.

VFIRMT107I MEMBER: <memname> DSNAME: <dataset name>

Explanation:
This informational message is included with an error message to identify failing dataset(member)name.

User response:
None. Informational message.

VFIRMT108I REPORTS FILE: <BACKUP| COMPRESS> SUCCESSFUL

Explanation:
RPTMAINT backs up or compresses the REPORTS PDS file if the program is APF authorized, the parameter BU is set to COND or FORCE and the REPORTS file is more than 85% full. It backs up the REPORTS PDSE file if the program is APF authorized and the BU parameter is set to FORCE. RPTMAINT has determined that a backup and compress of the REPORTS file is needed and has been completed successfully.

User response:
None. Informational message.

VFIRMT109I DELETE <memname>/<jobname>

Explanation:
RPTMAINT has successfully deleted REPORTS pds <memname> ember and its associated record in the $INDEX file.

User response:
None. Informational message.

VFIRMT110I KEEP <memname>/<jobname>

Explanation:
Based on VFIPARMS RPTRETPD and RPTGENS, RPTMAINT has determined that record <memname> in $INDEX qualifies to remain there. And it found that the associated REPORTS pds <memname> member exists.

User response:
None. Informational message.

VFIRMT111I REPORTS($INDEX) INUSE BY JOB <jobname> OPERATOR CANCEL REQUESTED

Explanation:
This informational message reports on an operator cancel request. See Informational Messages – Console.

User response:
None. Informational message.

VFIRMT112I <memname | rptname> TRUNCATED TO 7 CHARACTERS FOR ISPF STATISTICS

Explanation

RPTMAINT program assigns each PDS member it creates the same ISPF statistics ID as the original report name. If greater than 7 characters, the ID is truncated to 7. For example, member PULLTAPE is truncated to PULLTAP ID.

```
Name     VV.MM Changed       Size Init Mod ID
PULLTAPE 01.01 02/08/01 13:18  36   25   0 PULLTAP
```

User response:
None. Informational message.

VFIRMT113I REPORTS FILE: >= 85% USED, IS NOW/NEEDS COMPRESSING

Explanation

If running APF authorized and REPORTS file not PDSE and >= 85% full:

- Copy REPORTS PDS to same name PDS with added or replaced last qualifier of XDDDHHMM (DDD=Julian day, HH=hour, MM=minutes)

- Compress REPORTS PDS in place
- Message when REPORTS PDS >= 85% full with extents at maximum
- Message when REPORTS PDS last directory block used

User response:
None. Informational message.

VFIRMT114I RPTMAINT <is | not > RUNNING APF AUTHORIZED

Explanation:
This informational message reports if RPTMAINT program is running APF authorized or not. If authorized, and the REPORTS file is a PDS not a PDSE, RPTMAINT automatically backs up or compresses the REPORTS file, or both, if it is >= 85% full and the RPTMAINT parameter BU is set to COND or FORCE. RPTMAINT automatically backs up the REPORTS file if the program is authorized, the REPORTS file is a PDS and the RPTMAINT parameter BU is set to FORCE.

User response:
None. Informational message.

VFIRMT115I REPORTS FILE: PDSE – COMPRESS BYPASSED

Explanation:
Compression is bypassed if RPTMAINT program is not running APF authorized, or REPORTS file is a PDSE, and REPORTS file is >= 85% full.

User response

Manually compress REPORTS file to ensure the success of future RPTMAINT and BRNYPRNT runs. Additional messages sent:

- Message when REPORTS PDS >= 85% full with extents at maximum
- Message when REPORTS PDS last directory block used

VFIRMT116I REPORTS FILE: <BACKUP | COMPRESS> DSNAME= <backup_reports_dsn | reports_dsn>

Explanation:
This informational message reports when RPTMAINT runs a backup or a compress of the specified DSNAME.

User response:
None. Informational message.

VFIRMT117I PRE-MAINTENANCE BACKUP OPTION = <NO | CONDITIONAL | FORCE>

Explanation:
This informational message reports what BU= option you have run RPTMAINT with.

User response:
None. Informational message.

VFIRMT190I PROGRAM COMPLETED

Explanation:
RPTMAINT processing is complete.

User response:
None. Informational message.

VFIRMT101W <memname | jobname> DELETE ERROR. RETURN CODE: X"nnnnnnnn" RECORD OMITTED FROM $INDEX

Explanation:
Based on VFIPARMS RPTRETPD and RPTGENS, RPTMAINT has determined that record <memname> in $INDEX qualifies to be deleted. The program was unable to delete the existing associated REPORTS PDS <memname> member. The program removed this record from REPORTS($INDEX).

User response:
Delete REPORTS PDS <memname> manually. If the problem persists, contact Technical Support.

VFIRMT102W REPORTS FILE: >= 15 EXTENTS USED

Explanation:
This warning message is given if the REPORTS file is currently running with 15 or more extents.

User response:
As a best practice, rebuild the REPORTS file manually to handle its current size. It needs more primary and secondary space.

VFIRMT103W REPORTS FILE: LAST DIRECTORY BLOCK IN USE

Explanation:
This warning message is given if the REPORTS file last directory block is currently being used.

User response:
As a best practice, rebuild the REPORTS file manually to handle a greater number of directory blocks.

VFIRMT104W REPORTS($INDEX) MEMBER <archive | original> NOT FOUND IN REPORTS FILE, RECORD OMITTED FROM $INDEX

Explanation:
Based on VFIPARMS RPTRETPD and RPTGENS, RPTMAINT has determined that record <archive> in $INDEX qualifies to stay. The program determined that the associated REPORTS PDS <archive> member does

not exist. The program did remove the record from $INDEX.

User response:
Check prior job execution listings to determine what happened to the REPORTS PDS <archive> member. If still unsure, contact Technical Support.

VFIRMT105W INVALID PARAMETER BYPASSED <parameter>

Explanation:
This warning message is given if an invalid input parameter is found. The parameter is bypassed and processing continues.

User response:
See the RPTMAINT section for a list of parameters and their potential values.

VFIRMT100E VFIPARM IS COPY=NO, PRIOR MULTIPLE-REPORTS NOT GENERATED.

Explanation:
In order for RPTMAINT to maintain the REPORTS file, prior multiple-reports must have been generated.

User response:
To generate multiple-reports change the following: VFIPARMS COPY=YES, COPYRPT step in reporting jcl PARM=('BYPASS=NO').

VFIRMT101E SYSTEM ERROR IN JMR PROCESSING.

Explanation:
RPTMAINT uses current jobname and jobtime. There was a system error while extracting this information.

User response:
Contact Technical Support.

VFIRMT102E DYNAMIC <DE / ALLOCATE> ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
An error occurred when RPTMAINT tried to dynamically allocate or deallocate one of the REPORTS pds <memname> member.

User response

For an allocate error, ensure that no other job is using the identified data set (member) and that the data set is not out of space or control blocks. Rerun the job. If the problem persists, contact Technical Support.

For a dynamic deallocate error, the error occurred when RPTMAINT tried to deallocate a REPORTS PDS <memname> member. Do nothing; when a deallocate fails here, it means that <memname> was not allocated.

VFIRMT103E OPEN ERROR DD: <ddname>

Explanation:
RPTMAINT experienced an error opening DD <ddname>.

User response:
Check the joblog for any IEC type messages. Respond as indicated by the *IBM Error Codes and Messages Guide*. If the problem persists, contact Technical Support.

VFIRMT104E RDJFCB ERROR DD: <ddname>

Explanation:
RPTMAINT experienced an error processing DD <ddname>.

User response:
Check the joblog for any IEC type messages. Respond as indicated in the IBM Error Codes and Messages Guide. If the problem persists, contact Technical Support.

VFIRMT105E GETMAIN ERROR. RETCODE: nnnnnnnn, $INDEX NOT UPDATED

Explanation:
RPTMAINT experienced an error getting storage to build and index table

User response:
Increase the region size available to the program. Rerun the job. If the problem persists, contact Technical Support.

VFIRMT106E SORT PROCESSING ERROR. RETURN CODE X"nnnnnnnn"

Explanation:
An error occurred when RPTMAINT tried to sort REPORTS PDS ($INDEX) member.

User response:
Contact Technical Support.

VFIRMT107E ENQUE RET=USE ERROR. RETURN CODE: nnnnnnnn

Explanation:
RPTMAINT enqueues the REPORTS($INDEX) member that ensures exclusivity. The message is returned if, after a successful enqueue test, the actual enqueue fails.

User response:
Ensure that no other non-IBM Z Batch Resiliency jobs are using REPORTS($INDEX) then rerun the job. If the problem persists, contact Technical Support.

VFIRMT108E REPORTS FILE ERROR: BLDL

Explanation:
RPTMAINT is reporting a BLDL error for REPORTS file.

User response:

Check for any IEC messages on the joblog then contact Technical Support with this information.

VFIRMT109E **IBM Z Batch Resiliency WTOR ERROR!**

Explanation:
During communication with the console operator, RPTMAINT experienced an error.

User response:
Check for error messages in the joblog and contact Technical Support with this information.

VFIRMT111E **GQSCAN ERROR. REASON CODE: nnnnnnnn**

Explanation:
RPTMAINT experienced an error while checking to see if the REPORTS($INDEX) member was in use by another job.

User response:
Contact Technical Support.

VFIRMT112E **CKKRS003 ERROR DDN/MEM = <ddname/memname> RETURN CODE = nnnnnnnn**

Explanation:
RPTMAINT calls program CKKRS003 to build members. An error occurred during this process.

User response:
Contact Technical Support with this information.

VFIRMT113E **IEBCOPY ERROR IN <BACKUP| COMPRESS>**

Explanation:
RPTMAINT calls program IEBCOPY to backup/ compress the reports file. An error occurred during this process. The complete IEBCOPY error report is also displayed.

User response:
Check the IEB messages that were displayed. There may be an out-of-space condition occurring. Correct this error and rerun RPTMAINT. Contact Technical Support if this problem reoccurs.

VFIRMT114E **BU=COND INVALID FOR PDSE DD=REPORTS. USE BU=FORCE OR NO**

Explanation:
RPTMAINT calls program IEBCOPY to backup/ compress the reports file. An error occurred during this process. The complete IEBCOY error report is also displayed.

User response:
Check the IEB messages that were displayed. There may be an out-of-space condition occurring. Correct this error and rerun RPTMAINT. Contact Technical Support if this problem reoccurs.

SMFEXT Messages

This section documents the SMF Extractor messages, along with the associated program that produces them.

CKKX0355E **CKKXWTR - SMF$ file count exceeded, reply "O" to override limit, or "C" to cancel**

Explanation:
The SMF$; file count has been exceeded during processing. This process can be overridden and continued or canceled.

User response:
As stated in the message, reply O to override limit, or C to cancel.

CKKX0364E ***date time* CKKXWTR - Allocation failed. Reply "R" to retry**

Explanation:
The allocation process for CKKWTR has failed.

User response:
As stated in the message, reply R to retry the allocation.

CKK0130I **'*date time*** SMF Types processed: *type*'**

Explanation:
The displayed SMF Types were processed. *type* contains all the SMF types processed.

User response:
None. Informational message.

CKK0141I **'*date time* ** Queue stats for PC to SMF: &&3 /&&4 Curr: &&5 Max: &&6 NQ=&&7.x DQ=&&8.x'**

Explanation

Displays the status of the PC routine to SMF reader queue, where

- &&3 is the maximum number of entries allowed on the queue
- &&4 is the queue length threshold when pausing SMF collection
- &&5 is the current length of the queue
- &&6 is the current highwater mark of the queue

- &&7 is the number of SMF records that have been put on this queue
- &&8 is the number of SMF records that have been read from this queue

User response:
None. Informational message.

CKK0260I **'*date time* .----------+ ----------..'**

Explanation:
SMF record processing message header boundary.

User response:
None. Informational message.

CKK0261I **'*date time* | '*date time* | SMF Record Statistics |'**

Explanation:
Header 1 for the SMF record statistics for records processed by the SMF extractor exits table.

User response:
None. Informational message.

CKK0262I **'*date time* |Record| Processed| Not-processed|'**

Explanation:
Header 2 for the SMF record statistics for records processed by the SMF extractor exits table.

User response:
None. Informational message.

CKK0263I **'*date time* |&&3. |&&4.|&&5.|'**

Explanation

Count for an individual SMF record, where

- &&3 is the SMF record number
- &&4 is the count of the number of time this record was not processed
- &&5 is the count of the number of time this record was processed

User response:
None. Informational message.

CKK0317I **'*date time* Modify Sub-Command parameter <&&3> is not supported'**

Explanation:
A user has issued an extractor modify command (F jobname,xxxxxx), where &&3 is the invalid modify command.

User response:
Issue a valid extractor modify command.

CKK0366I **'*date time* -- SMF/WTR Counts'**

Explanation:
Header message for subsequent messages CKK03668I, CKK0369I, CKK0370I, CKK0371I and CKK0372I that will display the number of SMF records read and written to the various output destinations.

User response:
None. Informational message.

CKK0368I **'*date time* -- SMF records read(&&3.)'**

Explanation:
&&3 contains a count of the number of SMF records read.

User response:
None. Informational message.

CKK0369I **'*date time* -- SOUTDSN(&&3.) OUT1DSN(&&4.) OUT2DSN(&&5.)'**

Explanation

Informational message where

- &&3 is the number of SMF record written to OUTDSN
- &&4 is the number of SMF records written to OUT1DSN
- &&5 is the number of SMF records written to OUT2DSN

User response:
None. Informational message.

CKK0370I **'*date time* -- OUT3DSN(&&3.) OUT4DSN(&&4.) OUT5DSN(&&5.)'**

Explanation

Informational message where

- &&3 is the number of SMF record written to OUT3DSN
- &&4 is the number of SMF records written to OUT4DSN
- &&5 is the number of SMF records written to OUT5DSN

User response:
None. Informational message.

CKK0371I **'*date time* -- OUTLGR(&&3.) OUT1LGR(&&4.) OUT2LGR(&&5.)'**

Explanation

Informational message where

- &&3 is the number of SMF record written to OUTLGR
- &&4 is the number of SMF records written to OUT1LGR

- &&5 is the number of SMF records written to OUT2LGR

User response:

CKK0372I **'*date time* -- OUT3LGR(&&3.) OUT4LGR(&&4.) OUT5LGR(&&5.)'**

Explanation

Informational message where

- &&3 is the number of SMF records written to OUT3LGR
- &&4 is the number of SMF records written to OUT4LGR
- &&5 is the number of SMF records written to OUT5LGR

User response:

SCANSMF Messages

This section documents messages from VFI's LEGACY program and the SCANSMF process, both the batch version (program BRNVB052) and the SCANSMF subtask of RTSMF, as well as messages from utility programs.

BRNVB05101E **mm/dd/yy hh:mm:ss - XLEGACY LOAD ERROR PROGRAM TERMINATING**

Explanation:
There is a problem loading the XLEGACY member from USERMAC.

User response:

BRNVB05103E **mm/dd/yy hh:mm:ss - SORT ERROR, XXXXXXXX PROGRAM TERMINATING**

Explanation:
This is a sort error. The sort process is indicated.

User response:
Contact Technical Support.

BRNVB05109W **REQ/APPL/JOB/DSN:*reason* / XXXXXXXX/JJJJJJJJ/DSN**

Explanation

These messages can appear in the VB51MSGS report. They indicate that the record is dropped for one of the following reasons:

- NOSTEP1 – A job termination record appears for the job without a step termination record.
- NOSTEP2 – HISTORY end of file reached without an end of step record.
- DUPEJOB – A new job initiation record was processed.

User response:

BRNVB0511E **XXXXXXXX ERROR ON XLEGACY MEMBER, ABENDING**

Explanation

WTO message indicates a problem processing the XLEGACY table. The following could be reported:

- WTO message indicates the problem processing the XLEGACY table. The following could be reported:
- 'NO RECS' - No records in XLEGACY member
- 'ENTINUSE' - A duplicate entry or no SMF30PGM detected (fields must start after column 1)
- 'FFILTER' - An error with the filter definition
- 'DYNTBL' - More than 10 dynamic tables, one per filter DD is needed
- 'FNEWPGM' - An error on the new program definition
- 'JOB' - An error with the JOB definition
- 'STEP' - An error with the STEP definition
- 'EXTSTRNG' - An error extracting the string within quotes
- 'CARDERR' - An error spanning XLEGACY cards
- Correct error on XLEGACY cards and resubmit.

User response:
Correct the error on XLEGACY card or cards, and resubmit.

BRNVB05200I **mm/dd/yy hh:mm:ss - OAT OBTAINED FOR:XXXXXXXX, AT:XXXXXXXX**

Explanation:
Issued when an Open Application Table entry is built.

User response:
None.

BRNVB05201E **mm/dd/yy hh:mm:ss - BRNVB052: OPEN FAILED FOR BKUPTBL FILE, RC=XXXX, FEEDBK=XXXX**

Explanation:
Issued by BRNVB052 on a BKUPTBL file open error. The return code (RC) and feedback (FEEDBK) fields show the respective VSAM codes.

User response:

See the IEC161I message in the z/OS DFSMS Macro instructions for Data Sets for more information regarding this error.

BRNVB05202I mm/dd/yy hh:mm:ss - XXXXXXXX CYCLE: YYYY.DDD-00 - OPENED ON SYSTEM: XXXX BY: XXXXXXXX yyyy.ddd hh:mm:ss.th ( )

Explanation:
Issued when the application opens.

User response:
None. Informational message.

BRNVB05216I mm/dd/yy hh:mm:ss - BRNVB052: BKUPTBL XXXXXXXXXXXX IN USE JOB(XXXXXXXX) SYSID(XXXX)

Explanation:
When the specified enqueue is in use, BRNVB052 waits for 5 minutes for the resource to be released. If not freed in that time, BRNVB052 determines what process is holding the resource, reports the job and system where the resource is in use and continues to wait until RTB processing can continue. In addition, if the enqueue is available to BRNVB052, but the open of the backup table fails because another job is holding the SYSDSN enqueue for the table, this message is issued prior to the abend by BRNVB052.

User response:
Resolve who is holding the enqueue and let the job continue.

BRNVB05217I mm/dd/yy hh:mm:ss - BRNVB052: BKUPTBL/XXXXXXXXXXXX ACQUIRED FOR PROCESSING

Explanation:
Issued by BRNVB052 when BKUPTBL file is acquired for processing.

User response:
None. Informational message.

BRNVB05219I mm/dd/yy hh:mm:ss - BRNVB052: BKUPTBL/XXXXXXXXXXXX RELEASED

Explanation:
Issued by BRNVB052 at end of BKUPTBL processing when the BKUPTBL file is released.

User response:
None. Informational message.

BRNVB05220I mm/dd/yy hh:mm:ss - BRNVB052: ACQUIRING BKUPTBL/ XXXXXXXXXXXX FOR PROCESSING

Explanation:
This message indicates the data set name for the BKUPTBL file. It is issued by BRNVB052 when BKUPTBL file processing is initiated.

User response:
None. Informational message.

BRNVB05202I mm/dd/yy hh:mm:ss - INTERVAL TERM. - TASK = XXXXXXXX, PGM = BRNVB052 , R/T = XX, COUNTS = 999,999,999

Explanation:
Issued by BRNVB052 at the end of an SMF interval. The number of records processed and BKUPTBL updates are reported.

User response:
None. Informational message.

BRNVB05212I mm/dd/yy hh:mm:ss - BKUPTBL CLOSE: nn nn nn nn nn nn

Explanation:
Issued by BRNVC007 at the end of BKUPTBL processing. This message is coupled with message BRNVB05213 providing counts for GETs, INSERTs, and so on, as indicated in the message.

User response:
None. Informational message.

BRNVB05213I mm/dd/yy hh:mm:ss - BKUPTBL STATS GET/KEY GET/SEQ INSERTS UPDATES POINTS DELETES

Explanation:
Issued by BRNVC007 at the end of BKUPTBL processing. This message is coupled with message BRNVB05212 providing counts for GETs, INSERTs, and so on, as indicated in the message.

User response:
None. Informational message.

BRNVB0523E mm/dd/yy hh:mm:ss - SORT ERROR, XXXXXXXX PROGRAM TERMINATING

Explanation:
This is a sort error. The sort process involved is indicated.

User response:
Contact Technical Support.

BRNVB0527E mm/dd/yy hh:mm:ss - ABENDING XXXXXXXXXXXXX

Explanation:
This message indicates a severe error.

User response:
Contact Technical Support.

BRNVB05301E mm/dd/yy hh:mm:ss - BRNVB053: XXXXXXXX PROGRAM LOAD ERROR

Explanation:
The indicated program load failed.

User response:
Resolve program error and rerun.

BRNVB05303E mm/dd/yy hh:mm:ss - SORT ERROR, XXXXXXXX PROGRAM TERMINATING

Explanation:
This message is issued on a sort error. The sort process involved is indicated.

User response:
Contact Technical Support.

BRNVB05306E mm/dd/yy hh:mm:ss - CRITICAL FILE ERROR

Explanation:
An error occurred loading the CRITICAL file.

User response:
Contact Technical Support.

BRNVB05307E mm/dd/yy hh:mm:ss - ABENDING XXXXXXXXXXXXXX

Explanation:
This message indicates a severe error.

User response:
Contact Technical Support.

BRNVB05308W mm/dd/yy hh:mm:ss – BRNVB053 * APPLICATIONS WILL OPEN ON FIRST ACCESS AND REMAIN OPEN THROUGHOUT THE EXECUTION ***

Explanation:
This message documents that the application, in this case NONPROD, remains open for the entire process.

User response:
None. Informational message.

BRNVB05401E mm/dd/yy hh:mm:ss - BRNVB054: XXXXXXXX PROGRAM LOAD ERROR

Explanation:
Indicated program load failed.

User response:
Resolve program error and rerun.

BRNVB05406E mm/dd/yy hh:mm:ss - CRITICAL FILE ERROR

Explanation:
An error occurred loading the CRITICAL file.

User response:
Contact Technical Support.

BRNVB05407E mm/dd/yy hh:mm:ss - ABENDING XXXXXXXXXXXXXX

Explanation:
This message indicates a severe error.

User response:
Contact Technical Support.

BRNVB05408W mm/dd/yy hh:mm:ss – BRNVB054 * APPLICATIONS WILL OPEN ON FIRST ACCESS AND REMAIN OPEN TRHOUGHOUT THE EXECUTION ***

Explanation:
This message documents that the applications remain open for the entire process.

User response:
None. Informational message.

BRNVB05501E mm/dd/yy hh:mm:ss - BRNVB055: XXXXXXXX PROGRAM LOAD ERROR

Explanation:
The indicated program load failed.

User response:
Resolve program error and rerun.

BRNVB05503E mm/dd/yy hh:mm:ss - SORT ERROR, XXXXXXXX PROGRAM TERMINATING

Explanation:
Issued on a sort error. The sort process involved is indicated.

User response:
Contact Technical Support.

BRNVB05506E mm/dd/yy hh:mm:ss - CRITICAL FILE ERROR

Explanation:
An error occurred loading the CRITICAL file.

User response:
Contact Technical Support.

BRNVB05507E mm/dd/yy hh:mm:ss - ABENDING XXXXXXXXXXXXXX

Explanation:
This message indicates a severe error.

User response:
Contact Technical Support.

BRNVB05508W mm/dd/yy hh:mm:ss – BRNVB055 * APPLICATIONS WILL OPEN ON FIRST ACCESS AND REMAIN**

OPEN TRHOUGHOUT THE EXECUTION *

Explanation:
This message documents that the applications remains open for the entire process.

User response:
None. Informational message.

BRNVC03101E mm/dd/yy hh:mm:ss - BRNVC031 RTA SCANSMF PROCESSOR(BRNVC052) NOT AVAILABLE

Explanation:
Issued by BRNVC031 if the module, BRNVC052, is not available in the STEPLIB. Processing is terminated for RTA (abend 3101 issued).

User response:
Resolve all SCANSMF modules: BRNVC052, BRNVC053, BRNVC034, BRNVC031, BRNVC032, BRNVC033.

BRNVC03102E mm/dd/yy hh:mm:ss - BRNVC031 APPTABLE NOT AVAILABLE

Explanation:
Issued by BRNVC031 if APPTABLE is not found. Processing is terminated for RTA (abend 3101 issued).

User response:
Resolve the APPTABLE and restart.

BRNVC03103I mm/dd/yy hh:mm:ss - BRNVC031 SCANSMF TERMINATION SUSPENDED, WAITING ON SUBTASK: VFI/RTB

Explanation:
Issued by BRNVC031 during RTA termination. The SCANSMF must wait for RTB to complete first.

User response:
None. Informational message.

BRNVC03104I mm/dd/yy hh:mm:ss - INTERVAL TERM. - TASK = XXXXXXXX, PGM = BRNVB034 , R/T = XX, COUNTS = 999,999,999

Explanation:
Issued by BRNVC031 at the end of an SMF interval. The number of records processed is reported.

User response:
None. Informational message.

BRNVC03105I mm/dd/yy hh:mm:ss - BRNVC031 SAT RESET

Explanation:
Issued by BRNVC031 when the SCANSMF Application Table is reset after an APPTABLE reload.

User response:
None. Informational message.

VFI032001I mm/dd/yy hh:mm:ss - BRNVC032: SCANSMF STATS - APL ENTRIES: 999999 JOB ENTRIES: 999999 DSN ENTRIES: 999999

Explanation:
Issued by BRNVC031 when the SCANSMF Application Table is reset after an APPTABLE reload.

User response:
None. Informational message.

BRNVC03300I mm/dd/yy hh:mm:ss - BRNVC033: SAT OBTAINED FOR: XXXXXXXX AT: XXXXXXXX

Explanation:
Issued by BRNVC033 when the SCANSMF Application Table entry is acquired.

User response:
None. Informational message.

BRNVC03301I mm/dd/yy hh:mm:ss - BRNVC033: TOTAL SJDE(S) RECOVERED FOR APPLICATION: XXXXXXX (99999999)

Explanation:
Issued by BRNVC033 after recovering the SAT.

User response:
None. Informational message.

BRNVC03302I mm/dd/yy hh:mm:ss - BRNVC033: SAT INITIALIZATION COMPLETE

Explanation:
Issued by BRNVC033 after SAT initialization.

User response:
None. Informational message.

BRNVC03303E mm/dd/yy hh:mm:ss - BRNVC033: VFISANDX SLOTS EXCEEDED

Explanation:
Issued by BRNVC033 when the SAT index table is exceeded. RTA terminated (abend 3309).

User response:
Retry.

BRNVC03304I mm/dd/yy hh:mm:ss - BRNVC033: TOTAL SAT(S) RECOVERED: 99999999

Explanation:
Issued by BRNVC033 after SAT recovery.

User response:
None. Informational message.

BRNVC03305I mm/dd/yy hh:mm:ss - VFISANDX OBTAINED - ADDRESS = XXXXXXXX',C', SUB-POOL/ LENGTH = XXXXXXXX

Explanation:
Issued by BRNVC033 when the SAT index is acquired.

User response:
None. Informational message.

BRNVC03401E * mm/dd/yy hh:mm:ss - BRNVC034: DYNAMIC ALLOCATION ERROR - R15(XXXX) S99ERROR(XXXX) DDN: XXXXXXXX

Explanation:
Issued by BRNVC034 when a dynamic allocation error occurs. The values of register 15, the dynamic allocation S99ERROR and S99INFO fields are displayed.

User response:
Refer to the IBM Assembler Services Guide for descriptions of the error codes.

BRNVC03401E * mm/dd/yy hh:mm:ss - BRNVC034: DSN: XXXXXXXXXXX FUNCTION TERMINATED

Explanation:
Issued by BRNVC034 when a dynamic allocation error occurs. The RTA address space is terminated, abend 3203.

User response:
Correct and retry.

BRNVC03402E * mm/dd/yy hh:mm:ss - *CSI_error*

Explanation
Issued by BRNVC034 for any of the following CSI errors:

R15(XXXXXXX) CSI GETMAIN FAILED
R15(XXXXXXX) CSI CALL FAILED
CSICRETN(XXXXXXXX) CSI CALL FAILED

User response:
Review the error returned, and correct.

BRNVC03403I mm/dd/yy hh:mm:ss - BRNVC034 SSP$ DATASET XXXXXXXXXXX DELETED, RECS: NNN,NNN,NNN

Explanation:
Issued by BRNVC034 after processing the BKUPTBL records, the file is deleted and the number of records processed is reported.

User response:
None. Informational message.

BRNVC03404E mm/dd/yy hh:mm:ss - BRNVC034 VSAM ERROR ON BKUPTBL FILE :R14(XXXXXXXX) R15(XXXXXXXX) R0(XXXXXXXX) REASON CD/C21VS/$VSAMIO

Explanation:
Issued by BRNVC034 when a VSAM error occurs writing to the BKUPTBL file. The RTA address space is terminated, abend 3203.

User response:
None. Informational message.

BRNVC03405E mm/dd/yy hh:mm:ss - BRNVC034: *BKUPAUD_file_error*

Explanation
Issued by BRNVC034 when the indicated error occurs on the BKUPAUD file:

OPEN FAILED FOR BKUPAUD FILE, RC= XXXX
REASON CODE= XXXX
PUT FAILED FOR BKUPAUD FILE, RC= XXXX
REASON CODE= XXXX

System action:
The RTA address space is terminated, abend 3203.

User response:
Correct the error and rerun.

BRNVC05301I mm/dd/yy hh:mm:ss - SAT OBTAINED FOR: XXXXXXXX AT: XXXXXXXX

Explanation:
Issued by BRNVC053 when an SCANSMF application table entry is obtained. Data for SCANSMF restart comes from the SSPCKPT data set which contains the entries from the last SCANSMF checkpoint.

User response:
None. Informational message.

BRNVC05303E mm/dd/yy hh:mm:ss - BRNVC053: ERROR R15(XXXXXXXX) SSP$FILE OPEN FAILED RTA/SCANSMF TERMINATED

Explanation:
Issued by BRNVC053 when an open error occurs on the SSP$file.

User response:
Evaluate the return code and correct.

BRNVC05304I mm/dd/yy hh:mm:ss - BRNVC053: ALLOCATING SSP$FILE FILE: XXXXXXXXXXX

Explanation:
Issued by BRNVC053 when an SSP$ file is allocated. The file name is documented.

User response:
None. Informational message.

BRNVC05305E mm/dd/yy hh:mm:ss - BRNVC053: VFIPARM ERROR- XXXXXXXXXX SCANSMF FUNCTION TERMINATED

Explanation

Issued by BRNVC053 when one of the following VFIPARM errors is detected:

SSPLPRI ERROR
MEMBER NOT FOUND
SSPPFX LENGTH ERROR

User response:
Evaluate the error and correct.

BRNVC05306E mm/dd/yy hh:mm:ss - BRNVC053: DYNAMIC ALLOCATION ERROR - R15(XXXX) S99ERROR(XXXX) DDN: XXXXXXXX

Explanation:
Issued by BRNVC053 when a dynamic allocation error occurs.

User response:
Evaluate the error and correct.

BRNVC05308I mm/dd/yy hh:mm:ss - BRNVC053: SSP$FILE RECORDS WRITTEN: 999999999

Explanation:
Issued by BRNVC053 to document the number of records written to the SSP$ file.

User response:
None. Informational message.

BRNVC05310I mm/dd/yy hh:mm:ss - JJJJJJJJ JOBnnnnn YYYY.DDD hh:mm:ss.TH LEGTABLE IIIIIIIIIIIIIIII KEY J/S/P:JJJJJJJJ SSSSSSSS PPPPPPPP

Explanation

Issued by BRNVC053 when the DIAGNOSE function is active for SCANSMF. The job (JJJJJJJJ), job number, and SMF date/time for the job are listed. Additionally, the status information (IIIIIIIIIIII) and LEGTABLE key are listed. If the criteria is met, the status is MATCH, indicating the job met the LEGTABLE specifications and a BKUPTBL record is built. If the job fails to meet the specifications, the reason is listed in the information status field. Examples:

SBJCBKUP JOB05230 YYYY.DDD hh:mm:ss.TH LEGTABLE MATCH KEY J/S/P:SBJCBKUP MACLIB IEBCOPY

This message indicates a MATCH, and BKUPTBL record is built.

#BACKUP JOB05193 YYYY.DDD hh:mm:ss.TH LEGTABLE JOB/STEP KEY J/S/P:#BACKUP JS60 ADRDSSU

This message indicates a failure in processing. The mismatch is indicated in the status field - JOB/STEP. The LEGTABLE indicates job - #BACKUP, step - JS60. The #BACKUP job JOB5193 did not have a step JS60 to continue processing. No BKUPTBL record is built.

User response:
None. Informational message.

BRNVL01701E mm/dd/yy hh:mm:ss - INPT FILE ERROR NN, REC:

Explanation:
Issued by BRNVL017 to indicate an error in the input record. There are four tables loaded by BRNVL017: LEGTABLE, REPLMETH, REPLPGM and EXCLDDNS.

User response:
None. Informational message.

BRNVL01701I mm/dd/yy hh:mm:ss - XXXXXXXX TABLE LOADER STARTED

Explanation:
Issued by BRNVL017 to indicate a load table process is starting. The table is identified.

User response:
None. Informational message.

BRNVL01702I mm/dd/yy hh:mm:ss - XXXXXXXX LOADED - 999 INPUT, 999 OUTPUT, 999 ERROR

Explanation:
Issued by BRNVL017 to indicate a load table process is starting. The table is identified.

User response:
None. Informational message.

BRNVL01703E mm/dd/yy hh:mm:ss - XXXXXXXX LOAD FAILURE, REASON XXXXXXXXXX R15 =

Explanation

Issued by BRNVL017 to indicate a load table process failed. The table and cause is identified. The REASON is one of the following:

- 0008 - GET INPUT ERR0R
- 0012 - QIO OPEN ERROR
- 0016 - QIO PUT ERROR

- 0020 - EXCLDDNS CARD ERROR
- 0024 - OPEN INPUT ERROR

User response:
None. Informational message.

BRNVB0511E XXXXXXXX ERROR ON XLEGACY MEMBER, ABENDING

Explanation
WTO message indicates the problem processing the XLEGACY table. The following could be reported:

- 'NO RECS' - No records in XLEGACY member
- 'ENTINUSE' - A duplicate entry or no SMF30PGM detected (fields must start after column 1)
- 'FFILTER' - An error with the filter definition
- 'DYNTBL' - More than 10 dynamic tables, one per filter DD is needed
- 'FNEWPGM' - An error on the new program definition
- 'JOB' - An error with the JOB definition
- 'STEP' - An error with the STEP definition
- 'EXTSTRNG' - An error extracting the string within quotes
- 'CARDERR' - An error spanning XLEGACY cards

User response:
Correct the error on XLEGACY cards, and resubmit.

Simulated Recovery Messages

This section documents the messages originating in the Simulated Recovery process.

BRNVC01101E mm/dd/yy hh:mm:ss - BRNVC011: PARM ERROR- *parameter_error*

Explanation
This error message is issued by BRNVC011 when an unacceptable parameter is entered. The *parameter_error* can be any of the following:

- MEMBER NOT FOUND - Parameter member RTVRSPRM not found in USERMAC.
- SIMPFX LENGTH ERROR - The length of the value for SIMPFX is invalid; it must be 1 to 19 characters. Note that this value is combined with the value of SIMNAME to build the SIMLOGF data set name so it should conform to data set naming conventions. Don't end the value with a period.
- SIMNAME LENGTH ERROR - The length of the value for SIMNAME is invalid, it must be 1 to 8 characters. Note that this value is combined with the value of SIMPFX to build the SIMLOGF data set name so it should conform to data set naming conventions. Don't end the value with a period.
- SIMPFX/SIMNAME LENGTH ERROR - The length of the value for SIMNAME combined with SIMPFX is invalid. The total must be 1 to 28 characters when joined and qualified.
- SIMLPRI ERROR - You may specify a primary value for the SIMLOGF definitions. A value of zero causes this message to be issued. The maximum value is 1000, and a value larger than 100 is adjusted to 1000. The default is 50 cylinders.
- SIMLSEC ERROR - You may specify a secondary value for the SIMLOGF definitions. A value of zero causes this message to be issued. The maximum value is 1000, and a value larger than 100 is adjusted to 1000. The default is 50 cylinders.

User response:
Evaluate the error text, correct and retry.

BRNVC01102E mm/dd/yy hh:mm:ss - BRNVC011: DYNAMIC ALLOCATION ERROR - R15(xxxxxxxx) S99ERROR(xxxx) S99INFO(xxxx)

Explanation:
This message is issued when SIMLOGF dynamic allocation fails. The values of register 15, the dynamic allocation S99ERROR and S99INFO fields are displayed.

User response:
Refer to the IBM *Authorized Assembler Services Guide* for descriptions of the error codes. Message BRNVC01104I precedes this message; evaluate the DSN allocating and ensure it meets DSN naming conventions. For example, S99ERROR(035C) is an invalid DSN structure. Evaluate the error codes, correct and retry.

BRNVC01103E mm/dd/yy hh:mm:ss - BRNVC011: ERROR *error_text* SIMULATION RECOVERY FUNCTION TERMINATED

Explanation
The *error_text* of the message is issued for the following:

- BKUPTBL ENTRY NOT FOUND - The BKUPTBL entry is not defined in VFITABLS.

- R15(XXXXXXXX) VRSCKPT OPEN FAILED - The VRS Checkpoint Data set failed to open, the contents of register 15 are printed.
- R15(XXXXXXXX) SIMLOGF OPEN FAILED - The currently allocated SIMLOGF file failed to open, the contents of register 15 are printed.
- R15(XXXXXXXX) IGGCSI00 CALL FAILED - IGGCSI00 is invoked to determine the uniqueness of the data set being allocated. The call failed, the contents of register 15 are printed.
- SIM$ HLQS EXIST, UPD USERMAC.(RTVRSPRM) - The data set qualifiers for this VRS run already exists. The data sets for each run must be unique. You must either delete any existing data sets with the same generated high level qualifier or update USERMAC. member RTVRSPRM values for SIMNAME and SIMPFX.
- DATA RET ERR - The data returned from IGGCSI00 is in error.
- BKUPTBL NOT LOADED, FUNCTION TERMINATED - The VRS process attempts to create a condensed copy of the BKUPTBL loaded into storage for VFIV007 (RTB). The BKUPTBL is not loaded.
- R15(XXXXXXXX) SIMTSNAP OPEN FAILED - The condensed Simulate Table built from the BKUPTBL by BRNVC011 is copied to the SIMTSNAP file to support restart. The open for this data set failed. The contents of register 15 are printed.
- R15(XXXXXXXX) VRSCKPT OPEN FAILED - VRSCKPT contains a copy of the internal header information for the VRS process. It is used for restart if necessary. Otherwise, there is no data in this file. That is, if there is a record in this data set, an active VRS process is running. The contents of register 15 is printed.
- R15(XXXXXXXX) VRSCKPT OPEN FAILED - The VRSCKPT data set failed to open. The contents of register 15 is printed.

User response:
Evaluate the error text, correct and retry.

BRNVC01104I mm/dd/yy hh:mm:ss - BRNVC011: ALLOCATING SIMLOGF FILE: *dsn*

Explanation:
This informational message is issued when the SIMLOGF is allocated. *dsn* identifies the data set.

User response:
None. Informational message.

BRNVC01105I mm/dd/yy hh:mm:ss - BRNVC011: RUN NAME=xxxxxxxx *status*

Explanation
This informational message reports status regarding the execution of the VRS process. In addition to reporting the name of the VRS run, the following *status* text further identifies the intent:

- RECOVERY DATA COLLECTION STARTED - Reports the start of collecting data for the VRS process.
- ACTIVE RE-STARTED WARM - Reports the process is restarting and data was collected during the prior execution.
- ACTIVE - Reports the VRS process is active.
- ENDING (END TIME REACHED) - Reports the VRS process in terminating due to the 24 hour time interval.
- ENDING (END BY REQUEST) - Reports the VRS process is terminating by request (via Command Processor).

User response:
None.

BRNVB01201E * mm/dd/yy hh:mm:ss - BRNVB012: DYNAMIC ALLOCATION ERROR - R15(xxxxxxxx) S99ERROR(xxxx) S99INFO(xxxx) DDN:xxxxxxxx

Explanation
Sub message A: BRNVB01201E * mm/dd/yy hh:mm:ss - BRNVB012: DSN: xxxxxxxxxxxxx FUNCTION TERMINATED

Sub message B: BRNVB01201E * mm/dd/yy hh:mm:ss - BRNVB012: DSN: xxxxxxxxxxxxx NO DSNS FOR HLQ FOUND FUNCTION TERMINATED

These messages are issued by BRNVB012, the batch report SIM$ allocation routine, when there is a failure during dynamic allocation. Each SIM$ data set created during the VRS process is dynamically allocated and then concatenated under the input ddname SMF$IN. If the error codes indicate no high level qualifiers exist for the requested execution, sub message B is issued.

User response:
Refer to IBM *Authorized Assembler Services Guide* for descriptions of the error codes. Evaluate the error text, correct and retry.

BRNVB01202E * mm/dd/yy hh:mm:ss - PARM ERROR

Explanation
These common messages are issued for the identified error. When appropriate the content of register 15 is printed.

BRNVB01202E * mm/dd/yy hh:mm:ss - R15(XXXXXXX) CSI GETMAIN FAILED

BRNVB01202E * mm/dd/yy hh:mm:ss - R15(XXXXXXX) DDNLIST GETMAIN FAILED

BRNVB01202E * mm/dd/yy hh:mm:ss - R15(XXXXXXX) CSI CALL FAILED

BRNVB01202E * mm/dd/yy hh:mm:ss - SICRETN(XXXXXXXX) CSI CALL FAILED

User response:
Evaluate the error text, correct and retry.

BRNVB01203E *** mm/dd/yy hh:mm:ss - BRNVB012: DYNAMIC CONCATENATION ERROR - R15(xxxxxxxx) S99ERROR(xxxx) S99INFO(xxxx)**

Explanation:
This message is issued when the concatenation of the SIM$ files fails.

User response:
Refer to IBM *Authorized Assembler Services Guide* for descriptions of the error codes. Evaluate error codes, correct and retry.

BRNVL01401I **mm/dd/yy hh:mm:ss - MIROVOLS TABLE LOADER STARTED**

Explanation:
This message indicates the MIROVOLS table load is starting.

User response:
None

BRNVL01402I **mm/dd/yy hh:mm:ss - MIROVOLS LOADED – X,XXX,XXX INPUT, X,XXX,XXX ERROR, X,XXX,XXX OUTPUT**

Explanation:
This message indicates the MIROVOLS table load is finished and documents the records processed to load the table.

User response:
None. Informational message.

BRNVL01403E **mm/dd/yy hh:mm:ss - MIROVOLS LOAD FAILURE, REASON xxxxxxxxxxx**

Explanation
This message indicates the MIROVOLS table load encountered an error.

Reasons:

0004 MIROVOLS HEADER ERR

The header record in the MIROVOLS USERMAC. member is invalid, simulated recovery continues, but mirror volume processing is ignored.

0008 R15=

There was an open error on the input MIROVOLS USERMAC. member, evaluate the IO error return code in R15 to correct (CKKRC002 usually). Mirror processing for this simulated recovery is ignored.

0012 R15=

There was a get input error processing the MIROVOLS member, evaluate R15 to correct (CKKRC002 error usually). Mirror processing for this simulated recovery is ignored.

0016 R15=

There was an QIO open error building the in core MIROVOLS table, evaluate the QIO error return code in R15 to correct. Mirror processing for this simulated recovery is ignored.

User response:
To include mirror volume processing, correct the error.

BRNVB02501I **BRNVB025 xxxxxxxxxxx VOLSERS RECORDED**

Explanation:
This message indicates the MIROVOLS member is built and the number of volumes defined as mirror volumes.

User response:
None. Informational message.

SMFDSSU Messages

This section documents the messages originating from the IBM Z Batch Resiliency SMFDSSU process.

VAP1000I **VAPLDT <timestamp> ARCSKEXT: DSNAME=<dsn> Verification request**

Explanation:
Issued by: ABARS exit. Reports the data set name sent to an exit for verification .

User response:
None. Informational message.

SMFC001I **VDUDSS 06:48:52.177 Build: Version 4.40.L02 06/26/08 11.29**

Explanation:
Build date and time information.

User response:

None. Informational message.

VFI1005I **VFIAUT <timestamp> Product ID: BE is licensed to <name>**

Explanation:
License control: <name> - company name as coded in the license key.

User response:
None. Informational message.

SMF1021W **<id>(taskid) <timestamp> DDNAME=<d> Error from <m>., Rc=<rc>x, Rsn=<rs>x'**

Explanation:
Error after a macro <m> issued for ddname <d>, return code and a reason code are reported.

User response:
Review macro documentation and report to Technical Support.

SMF1024T **<id>(taskid) <timestamp> Output vol=<vol> selected for dsn=<dsn> ddname=<ddname>**

Explanation:
This message is issued by the DSS interface during the logical dump, when output data set <dsn> on a volume <vol> has been selected for processing, using <ddname> as an output DD.

User response:
None. Informational message.

SMF1025T **<id>(taskid) <timestamp> List of potential data sets that were selected:**

Explanation:
This message is issued by the DSS interface during the logical dump, after ADRDSSU determined that <N> data sets satisfied the selection criteria in the control statements, and started writing a dump to the output device. Message SMF1026T follows, displaying selected data sets.

User response:
None. Informational message.

SMF1026T **<id>(taskid) <timestamp> ++ CAT(Y) <dsn>**

Explanation:
This message is issued by the DSS interface during the logical dump, (see message SMF1025T). CAT(Y) means that the selected data set is cataloged, and CAT(N) can also be printed. Message SMF1027T is issued when the list is exhausted.

User response:
None. Informational message.

SMF1027T **<id>(taskid) <timestamp> End of list**

Explanation:
See message "SMF1026T" on page 337.

User response:
None. Informational message.

SMF1029T **<id>(taskid) <timestamp> + +Selected dsn=<DSN> Vol=<VOL> Seq=<SEQ>**

Explanation:
This message is issued by the DSS interface during the logical dump, when selected DSN tracks began to be written to an output volume, file sequence number is also reported.

User response:
None. Informational message.

SMF1030T **<id>(taskid) <timestamp> + +Finished dsn=<dsn> Rc=<nn> Bytes processed <mm>x**

Explanation:
This message is issued by the DSS interface during the logical dump, at the end of a dump of the selected data set. Return code is provided as reported by ADRDSSU and a hex value of bytes processed (8-bytes integer) is also displayed. Timestamps from this message and from SMF1029T message can be compared to estimate the time necessary to process the selected data set.

User response:
None. Informational message.

SMF1031T **<id>(taskid) <timestamp> --Task Completed; Rc=<n> Msg=<msg> Cputime=<time>x**

Explanation:
This message is issued by the DSS interface during the logical or physical dump, at the end of a task (there may be more than one task specified in the control statements). Task return code is provided, as well as a last 4 bytes of a message if any. For example, Rc=4 Msg=411W may be seen when ADR411W .. DATASET ... WAS NOT SERIALIZED ON REQUEST is issued and TOL(ENQF) is specified in the control statements.

User response:
None. Informational message.

SMF1032E **<id>(taskid) <timestamp> ** Level of DFSMS is not supported **

Explanation:
This message is issued by the ADRDSSU dump analysis routine, VFIRAD, when it determines that

DFSMS level is so low that QSAM support for LBI is not available (before OS/390 2.9).

User response:
Execution terminated. Contact Technical Support.

SMF1033T **<id>(taskid) <timestamp> Selected DD:<dd> Volser:<vol> Dsn:<dsn>**

Explanation:
This message is issued by the FDR interface during the physical dump (if SMF records are cut for full volume FDR dumps) or during the logical FDRDSF dump, and lists the data set being dumped.

User response:
None. Informational message.

SMF1034I **<id>(taskid) <timestamp> ::::::::: Encrypt:<e> HWcompress:<h> RSA:<r>**

Explanation

This message is issued by the DSS interface during the logical or physical dump, (if installation options enable analysis of hardware compressed and encrypted dump records), and report the status for the data set from SMF1029T message. Values can be as follows:

Encrypt (<e>):

- NONE - no encryption
- CLRTDES - encryption method
- CLRAES128 - encryption method
- ENCTDES - encryption method

HWcompress(<h>):

- YES - Hardware compression used
- NO - Hardware compression not used

RSA(<r>):

- YES - RSA label specified in the control statements (label is not displayed)
- NO - RSA label was not specified

User response:
None. Informational message.

SMF1500I **<id>(taskid) FDR Options in effect - Start**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, prior to listing some options from SMFFDRO module.

User response:
Review the options. Run VFIOPT module to list all options from SMFFDRO module.

SMF1501I **<id>(taskid) ++ SMF record number - nnn**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, and lists the SMF record number (202 is the optimal default) defined in the SMFFDRO options module.

User response:
None. Informational message.

SMF1502I **<id>(taskid) ++ After U0888 abend SMF records <opt> be cut**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, and lists one of the options from the SMFFDRO options module, where <opt> may be either "will" or "will not".

User response:
None. Informational message.

SMF1503I **<id>(taskid) ++ SMF records <opt> be cut for full volume FDR dumps**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, and lists one of the options from the SMFFDRO options module, where <opt> may be either "will" or "will not".

User response:
None. Informational message.

SMF1600I **<id>(taskid) FDR Options in effect - End**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, after the listing of some options from SMFFDRO module.

User response:
None. Informational message.

SMF2001E **<id>(taskid) <timestamp> Dataset is not in ADRDSSU Dump format.**

Explanation:
This message is issued by VFIRAD (ADRDSSU dump reading module), when it determines that it cannot recognize the input data set as ADRDSSU dump.

User response:
Review the JCL and correct.

SMF2002I **<id>(taskid) <timestamp> <f> Physical dump of Volume=<vol> Created on <date time>**

Explanation:
This message is issued by the DSS interface during the physical dump, at the beginning of a dump. It is also

issued by the ADRDSSU dump analysis module VFIRAD, during a read of a dump. <f> value is DUMP, when issued by the ADRDSSU interface, and READ when issued by VFIRAD.

User response:
None. Informational message.

SMF2003I **<id>(taskid) <timestamp> <f> Cyls <cyl> Tracks/cyl <n> Max.blksz <b> UCBTBYT4 <ucb>x Model 00x**

Explanation:
This message is issued by the DSS interface during the physical dump, after SMF2002I message, displaying the parameter of an input DASD volume.

User response:
None.

SMF2004I **<id>(taskid) <timestamp> <f> Total tracks <tracks>, tracks dumped <n>**

Explanation:
This message is issued by the DSS interface during the physical dump, after SMF2004I message, displaying the total number of tracks on the input volume and the number of tracks dumped to an output device.

User response:
None. Informational message.

SMF2005I **<id>(taskid) <timestamp> <f> End of physical dump for Input volume=<vol>**

Explanation:
This message is issued by the DSS interface during the physical dump, at the end of a task.

User response:
None. Informational message.

SMF2007I **<id>(taskid) <timestamp> Logical dump of <N> datasets, Created on <date time>**

Explanation:
This message is issued by the DSS interface during the logical dump, after ADRDSSU determined that <N> data sets satisfied the selection criteria in the control statements, and started writing a dump to the output device.

User response:
None. Informational message.

SMF2018E **<id> Abend code <a>x Reason=<r>x from LOAD EP=FDROPT - FDR options not available.**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. Attempt to preload the FDR installation option module resulted in a non-zero return code from the LOAD macro.

System action:
Processing terminates.

User response:
Review the LOAD macro return and reason codes. Ensure FDROPT module is available.

SMF2019E **<id> Disk Volume Open Exit is not active in FDROPT module; use FDRZAPOP**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. After the load of FDROPT installation options module, its analysis determined that the OPENEXIT is not defined in FDROPT.

System action:
Processing terminates.

User response:
SMFFDRO requires that this exit is enabled by the installation, and an example of a null exit is provided. Example of FDRZAPOP execution JCL, to enable required exits, is provided. Make sure that correct FDROPT module is loaded.

SMF2020E **<id> Dataset Security Exit is not active in FDROPT module; use FDRZAPOP**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. After the load of FDROPT installation options module, its analysis determined that the SECEXIT is not defined in FDROPT.

System action:
Processing terminates.

User response:
SMFFDRO requires that this exit is enabled by the installation, and an example of a null exit is provided. Example of FDRZAPOP execution JCL, to enable required exits, is provided. Make sure that correct FDROPT module is loaded.

SMF2021I **<id>(taskid) Local versions of OPENEXIT and SECEXIT will be executed before SMFFDR exits**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. SMFFDR uses both exits and dynamically defined them in such a way as to allow installation versions of these exits to run first. SMFFDR honors the return code from these exits, for example, if installation security exit rejects the requested data set, no further processing is needed.

User response:
None. Informational message.

SMF2022E **<id> PARM=<p> is not supported; SMF will not be collected**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. Single letter FDR parm (for example, PARM=R) has been provided, but SMFFDR only supports dump-related parameters (for example, PARM=D or PARM=A).

User response:
Review the reason this job has been selected for SMFFDR run.

SMF2023E **<id> SYSIN dataset exists but cannot be processed**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. SMFFDR Sysin analysis module determined that the data set is not usable.

User response:
Review the SYSIN data set. If it appears to be correct and specified DUMP functions, contact Technical Support.

SMF2025I **<id>(taskid) Local Exits: OPENEXIT=<oname> SECEXIT=<dname>**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, and lists local exits of interest defined in the FDROPT options module, accessible by the SMFFDR code.

User response:
Make sure FDROPT module accessible by SMFFDR is the one that installation intends to use.

SMF2026I **++ <id>(taskid) Modname=<name> Ddname=<dd> Dsname=<dsn>**

Explanation:
This message is issued by the FDR interface during the physical or logical dump, and lists local exits of interest defined in the FDROPT options module, accessible by the SMFFDR code. This message follows SMF2025I message for OPENEXIT and SECEXIT modules, to help installation identify these modules.

User response:
Make sure FDROPT module accessible by SMFFDR is the one that installation intends to use.

SMF2027E **++ <id>(taskid) Preload for <m> failed - Exit module is in LPA or preloaded**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. An attempt to preload the exit module <m> failed for the specified reason. SMFFDR does not support LPA-resident FDR exit modules and attempts to continue.

User response:
Review FDR exits.

SMF2028E **++ <id>(taskid) Preload for <m> failed - Exit module is not reentrant**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. An attempt to preload the exit module <m> failed for the specified reason. SMFFDR does not support non-reentrant FDR exit modules and attempts to continue.

User response:
Review FDR exits.

SMF2029E **++ <id>(taskid) Preload for <m> failed - Exit module is in non-APF LINKLIST library**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. An attempt to preload the exit module <m> failed for the specified reason SMFFDR attempts to continue.

User response:
Review PROGxx member of a system's logical PARMLIB concatenation and make sure that the exit library is added to the APF list.

SMF2030E **++ <id>(taskid) Preload for <m> failed - <text>**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. An attempt to preload the exit module <m> failed as specified in the <text>. SMFFDR attempts to continue.

User response:
Review FDR exits and contact Technical Support.

SMF2033I **<id> <timestamp> Execution complete**

Explanation:
This message is issued by the DSS or FDR interface during the logical or physical dump, at the completion of the processing of control statements.

User response:
None. Informational message.

SMF2034E **++ <id> ++ FDR Rc=<rc> is higher than specified maxrc=<maxrc>**

Explanation:

This message is issued by the FDR interface during the physical or logical dump. The return code after running FDR/FDRDSF is higher than the MAXRC specified in the SMFFDRO module. SMF records are not written.

User response:
Review the listing. If it is desired to produce SMF records, make appropriate changes to either JCL or SMFFDRO module.

SMF2035E **++ <id> ++ SMF records are not generated because of FDR errors.**

Explanation:
See message "SMF2034E" on page 340

User response:
Correct FDR errors and retry. If the error continues, contact your System Programmer.

SMF2036E **++ <id> ++ Options module is not compatible with an installed version of SMFFDR**

Explanation:
This message is issued by the FDR interface during the physical or logical dump.

User response:
Rebuild the SMFFDRO options module from source, or, if source is missing, run VDUOPT module. Contact Technical Support.

SMF2037E **++ <id> ++ Could not load SMFFDRO options module**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. The installation options module could not be loaded during initialization. Other messages are issued before this message.

User response:
Review the job log.

SMF2038E **++ <id> ++ ADRDSSU Rc=<rc> is higher than specified maxrc=<maxrc>**

Explanation:
This message is issued by the ADRDSSU interface during the physical or logical dump. Return code after running ADRDSSU is higher than the maxrc specified in the SMFDSSUO module. SMF records are not cut.

User response:
Review the listing. If it is desired to produce SMF records, make appropriate changes to either JCL or SMFDSSUO module.

SMF2039E **++ <id> ++ SMF records are not generated because of ADRDSSU errors.**

Explanation:
See message "SMF2038E" on page 341.

User response:
Take note of the contents of the error and contact your System Programmer.

SMF2040E **++ <id> ++ U0888: SMF records are not cut - user option U0888=NOSMF**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. Installation options module SMFFDRO instructed SMFFDR to skip generation of SMF records after FDR user abend 888.

User response:
Review the listing. If it is desired to produce SMF records, make appropriate changes to either the JCL or the SMFFDRO module.

SMF2041I **++ <id> ++ U0888: Cutting SMF records - user option U0888=SMF**

Explanation:
This message is issued by the FDR interface during the physical or logical dump. Installation options module SMFFDRO instructed SMFFDR to generate SMF records after FDR user abend 888.

User response:
None. Informational message.

SMF2042E **++ <id> ++ Options module is not compatible with an installed version of SMFDSSU**

Explanation:
This message is issued by the SMFDSSU interface during the physical or logical dump.

User response:
Rebuild the SMFDSSUO options module from source, or, if source is missing, run VDUOPT module. Contact Technical Support.

SMF2043E **++ <id> ++ Could not load SMFDSSUO options module**

Explanation:
This message is issued by the ADRDSSU interface during the physical or logical dump. Installation options module could not be loaded during initialization. Other messages are issued before this message.

User response:
Review the job log.

SMF2044W **<id>(taskid) ddname <d> could not be processed; SMF type <t> will be written**

Explanation:

When writing SMF records, options module (either SMFFDRO or SMFDSSUO) instructed that SMF records have to be written both to the DD:SMFDSSUS and by using SMFWTM macro. For some reason DD:SMFDSSUS was not configured properly and could not be processed. SMFWTM macro is still issued and SMF records are written to SMF data sets.

User response:
Review the job log and options module.

SMF2045E **<id>(taskid) ddname <d> could not be processed; SMF is not written**

Explanation:
When writing SMF records, options module (either SMFFDRO or SMFDSSUO) instructed that SMF records have to be written the DD:SMFDSSUS. For some reason DD:SMFDSSUS was not configured properly and could not be processed. SMFWTM macro is NOT issued and SMF records are NOT written to SMF data sets.

User response:
Review the job log and options module.

SMF2046E **<id>(taskid) SMFWTM returned RC=<rc>; SMF record was not written because:**

Explanation:
When writing SMF records, SMFWTM macro returned non-zero return code as shown. Other messages follow, explaining the error in detail.

User response:
Review the job log.

SMF3008E **<id>(taskid) ++ The length specified in the RDW was less than 18 bytes**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for RC=8.

User response:
Contact Technical Support.

SMF3016E **<id>(taskid) ++ SMF is not active or has ended abnormally**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for RC=16.

User response:
Start SMF recording.

SMF3020E **<id>(taskid) ++ The installation-written IEFU83 exit suppressed the record**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for RC=20.

User response:
Review SMFU83 exit and correct.

SMF3024E **<id>(taskid) ++ The data was lost**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for RC=24.

User response:
See SMF documentation for detail explanation.

SMF3036E **<id>(taskid) ++ The record specified is not currently being recorded**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for RC=36.

User response:
Review SMF options and enable SMF recording for 202 records.

SMF3040E **<id>(taskid) ++ A buffer shortage problem caused the data to be lost**

Explanation:
See message SMF2046E. This is an explanation for RC=40.

User response:
See SMF documentation for detail explanation.

SMF3044E **<id>(taskid) ++ SMF could not establish recovery**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for RC=44.

User response:
See SMF documentation for detail explanation.

SMF3045E **<id>(taskid) ++ Undocumented reason**

Explanation:
See message "SMF2046E" on page 342. This is an explanation for an undocumented RC.

User response:
Review SMF documentation and contact Technical Support.

BRNVB020 Messages

This section documents the BRNVB020 messages.

BRNVB02001I mm/dd/yy hh:mm:ss - BRNVB020: KEYDHIST PROCESSOR xxxxxxx

Explanation:
Issued by BRNVB020 when the process is started and when it is completed, where xxxxxxx is STARTED or COMPLETED.

User response:
None. Informational message.

BRNVB02000E mm/dd/yy hh:mm:ss - KEYDHIST PROCESSOR ABENDING, CODE: x

Explanation

Issued by BRNVB020 when a serious error is encountered. There are two possible error codes:

- Error code 1 - indicates that the load of the RTDPRINT module CKKRC002 failed.
- Error code 2 - indicates that the load of the Keyed History processor module, BRNVC020, failed.

User response:
Correct the error and retry.

BRNVB031 Messages

The following messages may be issued by program BRNVB031 during the Keyed History Maintenance processing.

BRNVB031 PARM LENGTH IS 0, ABENDING

Explanation:
This message indicates no parameter is defined on the EXEC card.

User response:
Add a valid parameter and rerun.

BRNVB031 KEYDSN/KEYJOB BOTH SPECIFIED, ABENDING

Explanation:
This message indicates both KEYDSN and KEYJOB are specified.

User response:
Specify one or the other value and rerun.

BRNVB031 JGENS/DAYS BOTH SPECIFIED, ABENDING

Explanation:
This message indicates both JGENS and DAYS are specified.

User response:
Specify one or the other value and rerun.

BRNVB031 JGENS/DAYS > 2 POSITIONS, ABENDING

Explanation:
This message indicates the specified JGENS or DAYS value is greater than 2 positions.

User response:
Correct the JGENS or DAYS parameter value and rerun.

BRNVB031 JGENS/DAYS NOT NUMERIC, ABENDING

Explanation:
This message indicates the specified JGENS or DAYS value is not numeric.

User response:
Correct the JGENS or DAYS parameter value and rerun.

BRNVB031 DELDATE DAYS NUMERIC ERROR, ABENDING

Explanation:
This message indicates the DELDATE value is not numeric.

User response:
Correct the DELDATE parameter value and rerun.

BRNVB031 DELDATE DATE ERROR, ABENDING

Explanation:
This message indicates a DELDATE specification error.

User response:
Correct the DELDATE parameter value and rerun.

BRNVB031 UDHIST OPEN ERROR, ABENDING

Explanation:
This message indicates an open error on the UDHIST file.

User response:
Correct the file/DD specification error and rerun.

BRNVB031 KEYDSN/KEYJOB NOT SPECIFIED IN PARM, ABENDING

Explanation:
This message indicates KEYDSN or KEYJOB is not specified.

User response:

Specify KEYDSN or KEYJOB correctly and rerun.

BRNVB031 JGENS/DAYS NOT SPECIFIED IN PARM, ABENDING

Explanation:
This message indicates JGENS or DAYS is not specified.

User response:
Specify JGENS or DAYS correctly and rerun.

BRNVB031 REPORT OPTION INVALID, ABENDING

Explanation:
This message indicates REPORT= option is not SUMMARY, DETAIL or ONLY.

User response:
Specify the correct option and rerun.

BRNVB03102I TOTAL KEEP RECORDS PROCESSED *nnn,nnn,nnn*

Explanation:
Specifies the number of records kept after processing.

User response:

BRNVB03102I Total Delete Records Processed nn,nnn

Explanation:
Specifies the number of records deleted during processing.

User response:

BRNVB03101E Invalid PARM provided, terminating

Explanation:
Only one parameter may be specified; it must be one of JGENS or DAYS and must have a numeric value.

User response:
Correct the parameter statement and resubmit the job.

BRNVB032 (TABLEVFY) Messages

The following messages may be issued by program BRNVB032 in job TABLEVFY during RT/SMF Table Verification processing.

BRNVB03201E mm/dd/yy hh:mm:ss - BRNVB032: xxxxxxxx PROGRAM LOAD ERROR

Explanation:
This message indicates the identified load module is not found in the STEPLIB data set.

User response:
Ensure the identified module is in the STEPLIB library, and then rerun the job.

BRNVB03202E mm/dd/yy hh:mm:ss - BRNVB032: TABLE MEMBER NOT FOUND: xxxxxxx

Explanation:
This message is issued when the VFITABLS member is not found in the data set identified by PARMLIB.

User response:
Ensure the table member identified is in the PARMLIB data set, and rerun the job.

BRNVB03203I mm/dd/yy hh:mm:ss - PARMLIB ACCESS - MEMBER = xxxxxxxx, STATS = VER=xxx,MOD=xxx,MODDATE=yy.ddd,MODTIME=hh:mm

Explanation:
This message is issued to document the table's member information, including the member name, version, modification level, modification date and time.

User response:
None.

BRNVB03204I mm/dd/yy hh:mm:ss - PARMLIB OPENED - VOLUME = *volser*, DSNAME = xxxxxxxxx

Explanation:
This message indicates the name and VOLSER of the data set defined to the PARMLIB DD statement.

User response:
None. Informational message.

BRNVB03205E mm/dd/yy hh:mm:ss - BRNVB041 ERROR

Explanation:
This message is issued if there is a problem with the fake RTA builder.

User response:
Review RTDPRINT, correct the error, and rerun job.

BRNVB03206I mm/dd/yy hh:mm:ss - INVOKING TABLE LOADER: loadpgm TABLE: table

Explanation:
This message is issued before the identified loader program is called to load the identified table.

User response:
None. Informational message.

BRNVB03207I mm/dd/yy hh:mm:ss - TABLE LOADER COMPLETE

Explanation:
This message is issued at the end of the load process.

User response:
None. Informational message.

BRNVB03208E mm/dd/yy hh:mm:ss - INVENTRY FILE OPEN ERROR

Explanation:
This message if the INVENTRY file manager program, BRNVC021, is not in the STEPLIB or an error code is returned from BRNVC021 .

User response

Ensure BRNVC021 is available. Review other RTDPRINT message that may be issued by BRNVC021 for additional information. Correct the errors and rerun the job.

Each table loader routine issues their normal messages during this job. Review the messages for the loader to resolve errors or track the loader process.

BRNVB034 Messages

The following messages may be issued by programs in the MAINTLST process.

BRNVB03401E mm/dd/yy hh:mm:ss - BRNVB034: xxxxxxxx PROGRAM LOAD ERROR

Explanation:
This message is issued during initiation. The indicated program encountered an error after the LOAD request.

User response:
Determine why the program received an error (for example, not in the STEPLIB DSN). Correct the error and rerun.

BRNVB03402E mm/dd/yy hh:mm:ss - BRNVB034: TABLE MEMBER NOT FOUND: xxxxxxx

Explanation:
This message is issued if the indicated table member is not found in the supplied USERMAC.

User response:
Resolve missing table member and rerun.

BRNVB03403I mm/dd/yy hh:mm:ss - PARMLIB ACCESS - MEMBER = XXXXXXXX STATS = VER=NNN,MOD=NNN,MODDATE=yy.jjj,MODTIME=hh:mm

Explanation:
This message is issued to provide statistical information related to the indicated member.

User response:
None.

BRNVB03404I mm/dd/yy hh:mm:ss - PARMLIB OPENED - VOLUME = XXXXXX DSNAME = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX

Explanation:
This message is issued to provide PARMLIB information.

User response:
None. Informational message.

BRNVB03405E mm/dd/yy hh:mm:ss - PARMLIB NOT FOUND

Explanation:
This message is issued if the PARMLIB DD/MEMBER is not found.

User response:
Provide a valid PARMLIB and rerun.

BRNVB03406I mm/dd/yy hh:mm:ss - INVOKING TABLE LOADER: xxxxxxxx TABLE: xxxxxxxx/xxxxxxxx

Explanation:
This message is issued to identify the table loader and associated table.

User response:
None. Informational message.

BRNVB03407I mm/dd/yy hh:mm:ss - TABLE LOADER COMPLETE

Explanation:
This member is issued when the table load process is complete.

User response:
None. Informational message.

BRNVB03408E mm/dd/yy hh:mm:ss - SORT ERROR, PROCESS: xxxxxxxx PROGRAM TERMINATING

Explanation:
This message is issued if an error code is receive from SORT. The active process is indicated.

User response:
Review the SORT messages, correct and rerun.

BRNVB03410E mm/dd/yy hh:mm:ss - NO VFIPARMS

Explanation:

This message is issued if the VFIPARMS data set is not available.

User response:
Correct and rerun.

BRNVB03411E mm/dd/yy hh:mm:ss - RECORDS OUT OF ORDER

Explanation:
This message is issued if the returned records are not in order.

User response:
Ensure SORT completed successfully, and rerun.

BRNVB03412I mm/dd/yy hh:mm:ss – *EOJ_totals*

Explanation

This pair of messages is issued at the end of job to report various total categories. The *EOJ_totals* are:

- TOTALS CRIT IN: NNN,NNN,NNN CRITLIST: NNN,NNN,NNN UPD RECS : NNN,NNN,NNN EXCLDSN : NNN,NNN,NNN
- NO APPL: NNN,NNN,NNN STAT CHG: NNN,NNN,NNN NOT CATLG : NNN,NNN,NNN CRITICAL: NNN,NNN,NNN

User response:
None. Informational message.

BRNVB044 Messages

The following messages may be issued by the Inventory Update program BRNVB044.

BRNVB04401E mm/dd/yy hh:mm:ss - TRANIN ERROR XXXXXXXX, REC:

Explanation:
This message is issued when transaction verification fails. The error is indicated and the transaction record is listed.

User response:
Correct the error, and rerun.

BRNVL01501E mm/dd/yy hh:mm:ss - TRANIN ERROR REC:

Explanation:
This message is issued by the transaction table loader, BRNVL015 used by BRNVB044 to manage the transaction provided as input.

User response:
Correct the transaction in error, and rerun.

BRNVL01502I mm/dd/yy hh:mm:ss - TRANIN LOADED - X,XXXXX INPUT, X,XXX,XXX ERROR, X,XXX,XXX OUTPUT

Explanation:
This message documents the number of transactions input and loaded.

User response:
None. Informational message.

BRNVL01503E mm/dd/yy hh:mm:ss - TRANIN LOAD FAILURE, REASON XXXX

Explanation:
This message indicates a transaction load error. The reason is shown.

User response:
Correct the transaction in error, and rerun.

BRNVBCAT8 Messages

The following messages may be issued by program BRNVBCAT8 during Backup/Restore processing.

BRNVBCAT8 DEFAULT BACKUP METHOD NOT FOUND IN USERMAC.(VFIPARMS). ENTER EITHER THIRD INPUT PARM OR CHANGE (VFIPARMS).

Explanation:
This warning message reports that neither the input request line nor *hlq*.USERMAC.(VFIPARMS) lists a backup methodology.

System action:
Invalid request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct your *hlq*.USERMAC.(VFIPARMS) DMS option or add your BKUPMETH to the input request line for future executions of BRNVBCAT8. For details, see Generating IBM Z Batch Resiliency Backup JCL, Inclusions section.

BRNVBCAT8 REQUIRED INPUT PARM FIELD {APPLID | DSNPATT} MISSING - RECORD BYPASSED.

Explanation

This warning message reports that the DD=INCLDSN input request line is not valid, missing the required APPLID or DSNPATT field. APPLID,DSNPATT are

minimum required fields. INCLDSN RECORD POSITIONAL FIELDS INCLUDE:

{APPLID,DSNPATT,BKPMETH,DAYS_FROM_CREATE,DATATYPES}

System action:
Invalid request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct DD=INCLDSN input request line in question for future executions of BRNVBCAT8. For details, see "Adding data sets to the Include list" on page 75.

BRNVBCAT8 SOMETHING WRONG WITH INPUT PARAMETERS!

Explanation
This warning message reports that the DD=INCLDSN input request line is not valid. Line editing cannot find the first available blank. APPLID,DSNPATT followed by a blank are minimum input line requirements. INCLDSN RECORD POSITIONAL FIELDS INCLUDE:

{APPLID,DSNPATT,BKPMETH,DAYS_FROM_CREATE,DATATYPES}

System action:
Invalid request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct DD=INCLDSN input request line in question for future executions of BRNVBCAT8. For details, see Adding data sets to the Include list.

BRNVBCAT8 LESS THAN 2 INPUT PARAMETERS !

Explanation
This warning message reports that the DD=INCLDSN input request line is not valid, missing the required DSNPATT field. APPLID,DSNPATT are minimum required fields. INCLDSN RECORD POSITIONAL FIELDS INCLUDE:

{APPLID,DSNPATT,BKPMETH,DAYS_FROM_CREATE,DATATYPES}

System action:
Invalid request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct DD=INCLDSN input request line in question for future executions of BRNVBCAT8. For details, see Adding data sets to the Include list.

BRNVBCAT8 {4|5}TH PARM, {DAYS-FROM-CREATE|NUMBER_OF_GDGS}, MUST BE NUMERIC

Explanation:
This warning message reports that either 4th parm days-from-create or 5th parm number-of-gdgs is not numeric.

System action:
Invalid request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct DD=INCLDSN input request line in question for future executions of BRNVBCAT8. For details, see Adding data sets to the Include list.

BRNVBCAT8 4TH PARM, DAYS-FROM-CREATE=N, IGNORED WHEN COMBINED WITH RELREF GDG

Explanation:
This warning message reports that the 4th parm days-from-create is not valid when used with value N. This value is a request that BRNVBCAT8 bypass Catalog Search for DSNPATT. CS is required to extract the correct number of GDGs requested.

System action:
Program processes the request ignoring 4th parm. Informational message is generated.

User response:

BRNVBCAT8 FILE <'dsname'> MASKING NOT ALLOWED FOR REL REFERENCE GDG

Explanation:
This warning message reports an input request line edit error. It is invalid for any of the backup methodologies, but ABARS and DSS, to use masking in a relative reference GDG request.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct the input request line for future executions of BRNVBCAT8. For details, see "Adding data sets to the Include list" on page 75.

BRNVBCAT8 FILE <'dsname'> MASKING NOT ALLOWED: DAYS-FROM-CREATE HAS VALUE "N"

Explanation:
This warning message reports an input request line edit error. As stated above it is invalid for any of the backup methodologies, but ABARS and DSS, to use

masking in a relative reference GDG request. It is also invalid for any of the backup methodologies, but ABARS and DSS, to use masking with the value "N" in the days-from-create field.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct the input request line for future executions of BRNVBCAT8. For details, see "Adding data sets to the Include list" on page 75.

BRNVBCAT8 4TH PARM, DAYS-FROM-CREATE=N, IGNORED WHEN COMBINED WITH REL#GENS

Explanation:
This warning message reports an input request line edit error. It is invalid for any of the backup methodologies to request days-from-create value "N" in combination with a relative number of generations request.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct the input request line for future executions of BRNVBCAT8. For details, see "Adding data sets to the Include list" on page 75.

BRNVBCAT8 5TH PARM, REL#GENS, IGNORED WHEN COMBINED WITH RELATIVE REF GDG

Explanation:
This warning message reports an input request line edit error. It is invalid for any of the backup methodologies to combine relative reference GDG with a relative number of generations request.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct the input request line for future executions of BRNVBCAT8. For details, see "Adding data sets to the Include list" on page 75.

BRNVBCAT8 5TH SUBPARM RECTYPE ERROR. USE ONE OR MORE OF G###;C;N;A;X

Explanation:
This warning message reports an input request line edit error. Valid RECTYPE options include one or more of: G###;C;N;A;X. Options must be separated by a semicolon.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct the input request line for future executions of BRNVBCAT8. For details, see Chapter 9, "IBM Z Batch Resiliency Maintenance," on page 139: Generating Backup JCL - Data set Inclusions section.

BRNVBCAT8 FILE <'dsname'> CANNOT ADD GDGNODE .G%%%%V%% TO GDG BASE DSNPATT (>CL36)

Explanation:
This warning message reports an input request line edit error. Input dsname must be <=CL36 when not specifying a final node mask of "**" or "*" or "G*" or "G %%%%V%%" for a relative number of GDG request.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Correct the input request line for future executions of BRNVBCAT8. For details, see "Adding data sets to the Include list" on page 75.

BRNVBCAT8 FILE <'dsname'> GETMAIN FAILED FOR IGGCSI00

Explanation:
This warning message reports a GETMAIN error processing input request line DSNAME.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response

First increase your region size and rerun. If problem still occurs examine your dsname. Occasionally a masked request generates many dsname matches. This can require a work area that is not attainable. If this is the case it is advisable to split your dsname mask into smaller groupings. For example, instead of input request:

APPLID,*hlq*.**,DSS

use these input requests:

APPLID,*hlq*.A*.**,DSS

APPLID,*hlq*.BA*.**,DSS ...

For details, see Adding data sets to the Include list. Correct the input request for future executions of BRNVBCAT8. If the problem persists, contact Technical Support with your execution listing and copy of DD=INCLDSN.

BRNVBCAT8 FILE <'dsname'> CSI INVOCATION ERROR, R15(####) - RC(####) REASON(####)

Explanation:
This warning message reports an error during a call to catalog search interface.

System action:
Message line displayed with return code and reason code. Request line is bypassed. Program continues on to process the next input request line.

User response:
If the problem reoccurs, contact Technical Support with your execution listing.

BRNVBCAT8 FILE <'dsname'> IDCAMS ERROR. RC=####

Explanation:
This warning message reports an error running pgm=XCAMS. RC=#### is greater than 4.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Check that the data set in question is good by running an IDCAM LISTCAT against it. If the LISTCAT runs RC=0, contact Technical Support with the LISTCAT and your execution listing. If the LISTCAT runs RC<>0, fix the data set for future executions of BRNVBCAT8. For details, see Adding data sets to the Include list.

BRNVBCAT8 FILE <'dsname'> NOT FOUND IN CATALOG!

Explanation:
This warning message reports that the requested dsname was not found during a catalog search.

System action:
Request line is bypassed. Message generated with return code 4. Program continues on to process the next input request line.

User response:
Check that the DSNAME in question is correct. If not, correct input request line for future executions of BRNVBCAT8. For details, see Adding data sets to the Include list.

BRNVBCAT8 NO I/P REC MATCHES FOUND

Explanation:
This message reports that no cataloged data sets matched any input record requests.

System action:
Message generated with return code 4. Program ends.

User response:
Check DD=ERRLOG for prior error or warning messages related to DD=INCLDSN input cards. Correct the editing errors for future executions of BRNVBCAT8. For details, see “Adding data sets to the Include list” on page 75.

BRNVBCAT8 ERROR OPENING DD=INCFILE

Explanation:
This message reports on an error opening DD=INCFILE for output processing.

System action:
Message generated with return code. Program ends.

User response:
Check DD=JESMSGLG for any associated error codes and messages. Correct the error for future executions of BRNVBCAT8. For details, see Adding data sets to the Include list.

BRNVBCAT8 ERROR OPENING DD=INCLDSN

Explanation:
This message reports on an error opening DD=INCLDSN for input processing.

System action:
Message generated with return code. Program ends.

User response:
Check DD=JESMSGLG for any associated IBM error codes and messages. Correct the error for future executions of BRNVBCAT8. For details see “Adding data sets to the Include list” on page 75.

BRNVBCAT8 ERROR OPENING DD=VFIPARMS

Explanation:
This message reports on an error opening DD=VFIPARMS for input processing.

System action:
Message generated with return code. Program ends.

User response:
Check DD=JESMSGLG for any associated IBM error codes and messages. Correct the error for future executions of BRNVBCAT8. For details, see “Adding data sets to the Include list” on page 75.

BRNVBCAT8 QIO PROCESSING ERROR IN %%%%%%%%, RETURN CODE(####)

Explanation:
This message reports a QIO processing error. %%%% %%%% is for system debugging use.

System action:
Message generated with return code. Program ends.

User response:
Contact Technical Support with your execution listing.

BRNVBCT81 EXCLDSN REQUEST ERROR

Explanation:
This message indicates an error in the EXCLDSN request.

User response:
Correct the card and resubmit.

BRNVBCT81 PREMATURE END OF DATA ON INCLDSN

Explanation:
This error message indicates an end of file occurred on the INCLDSN dataset. This should not occur and indicates an EXCLDSN continuation request was made, but there was no additional card.

User response:
Correct the input in INCLDSN and resubmit.

BRNVBCT81 EXCLDSN *message_text*

Explanation

The *message_text* is either:

- INVALID CONTINUATION
- REQUEST ERROR

These messages indicate an error in the INCLDSN request EXCLDSN option. Additional information is listed in the ERRLOG output. The following indicates a reformatted initial card to include the positional parameter sensitivity and the error on the EXCLDSN statement AG.WORK00C.SUBFILE.G0052*. The continuation statement should begin in column 1.

```
====================================

ABC,AG.WORK400C.SUBFILE,DSS,,G:AG.WORK400
C.SUBFILE.G0053*,<==ORIGINAL

ABC,AG.WORK400C.SUBFILE,DSS,,G,:AG.WORK40
0C.SUBFILE.G0053*,<==REFORMATTED

AG.WORK400C.SUBFILE.G0053*

AG.WORK400C.SUBFILE.G0052*<== ERROR
```

System action:

User response:
None. Informational message.

BRNVBCAT8 EXCLDSN ERROR

Explanation:
Error message.

User response:
Correct the error in the INCLDSN member and rerun the job.

BRNVBDMS Messages

This section documents the BRNVBDMS messages.

ERROR01 ERRORS IN INPUT PARAMETERS

Explanation:
An error occurred when BRNVBDMS was editing input parameters <APPLID,SUBAPPID>. APPLID is required. SUBAPPID is optional. Both parameters must be 8 characters or less.

User response:
Make sure that the parameters are correct. Rerun the job.

ERROR03 IMPOSSIBLE TO READ TAPEHOLD DD

Explanation:
An error occurred when BRNVBDMS tried to read the DMS tape.

User response:
Contact Technical Support.

BRNVBDSS Messages

This section documents the BRNVBDSS messages.

INFO0010 NO RECORDS SELECTED FOR BKUPTBL PROCESSING. RC:0004

Explanation:
This message signals that BRNVBDSS found no data selected for backup processing.

User response:
None. Informational message.

INFO0099 THERE ARE NO ERROR MESSAGES, RETURN CODE=0000

Explanation:
This message signals the end of BRNVBDSS processing and displays the final return code.

User response:

WARN002D TDD PARM MUST BE <=8

Explanation:
BRNVBDSS has detected a syntax error while processing backup grouping DD=TAPEHOLD override TDD=.

System action:
BRNVBDSS continues processing but generate a return code of 4.

User response:
None. Informational message.

WARN002E DEFAULT DD=TAPE1 USED FOR DD=TDD

Explanation:
BRNVBDSS has detected a syntax error while processing backup grouping DD=TAPEHOLD override TDD=. No value was specified for TDD=. BRNVBDSS uses TDD=TAPE1 for the missing value.

System action:
BRNVBDSS continues processing using TDD=TAPE1 and generate a return code of 4.

User response:
None. Informational message.

WARN002G DEFAULT DD=SYSPRINT USED FOR DD=PDD

Explanation:
BRNVBDSS has detected a syntax error while processing backup grouping DD=DSSOUT override PDD=. BRNVBDSS uses PDD=SYSPRINT for the missing value.

System action:
BRNVBDSS will continue processing using PDD=SYSPRINT and generate a return code of 4.

User response:
None. Informational message.

ERROR01A INPARM FILE ALLOCATE ERROR

Explanation:
BRNVBDSS has detected a system error while allocating the DD=INPARM dataset.

System action:
BRNVBDSS ends processing and generate a return code of 8 or higher.

User response:
Rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR01B INPARM FILE OPEN ERROR

Explanation:
BRNVBDSS has detected an error while opening DD=INPARM.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response:
Check the joblog to see if there are any IEC type messages associated with the DD=INPARM. If so, correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR01C DD=INPARM INPUT PARAMETER ERROR %%%%%%%%

Explanation:
BRNVBDSS has detected an error while processing the input parameter specified.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response

Valid input parameters within DD=INPUT are:

- AID=%%%%%%%% - (Applid, up to 8 characters)
- SID=%%%%%%%% - (Subapplid up to 8characters)
- UIN=%%%%%%%%%%%% - (Userinfo up to 12 characters)
- TDD=%%%%%%%% - (DD=TAPEHOLD override for BGMAIN)
- PDD=%%%%%%%% - (DD=DSSOUT override for BGMAIN)
- GRP=%%%% - (GRP option for BGMAIN, up to 4 characters)

Check the JCL to see what input parameter is in error. Correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR01D PROCESSING ERROR IN GETMAIN

Explanation:
BRNVBDSS has detected an error while performing a GETMAIN command. The program is requesting 2048 * 44 = 90K bytes.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response:
Increase the job region size and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR02A APPLID PARM MUST BE <=8

Explanation:
BRNVBDSS has detected a syntax error while processing application id parm AID=.

System action:

BRNVBDSS ends processing and generate a return code of 8.

User response:
Check the JCL and make sure that the value of the APPLID parameter is from 1 to 8 characters long. Correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR02B SUBAPPL PARM MUST BE <=8

Explanation:
BRNVBDSS has detected a syntax error while processing subapplication id parm SID=.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response:
Check the JCL and make sure that the value of the SUBAPPL ID parameter is from 1 to 8 characters long. Correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR02C USERINFO PARM MUST BE <=12

Explanation:
BRNVBDSS has detected a syntax error while processing user information parm UIN=.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response:
Check the JCL and make sure that the value of the userid parameter is from 1 to 12 characters long. Correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR03A INSTREAM PARAMETERS MISSING

Explanation:
BRNVBDSS has detected that the instream parameters are missing.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response:
Check the JCL to see if the input parameters have been omitted. If so, correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR03B INSTREAM PARAMETER IN ERROR

Explanation:
BRNVBDSS has detected an error while processing the input parameter specified.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response

Valid input parameters are:

- AID=%%%%%%%% (Applid, up to 8 characters)
- SID=%%%%%%%% (Subapplid up to 8 characters)
- UIN=%%%%%%%%%%%% (Userinfo up to 12 characters)

Check the JCL to see what input parameter is in error. Correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

ERROR04A UNABLE TO READ TAPEHOLD DD

Explanation:
BRNVBDSS has detected an error while issuing a RDJFCB on DD=TAPEHOLD.

System action:
BRNVBDSS ends processing and generate a return code of 8.

User response:
Check the joblog to see if there are any IEC type messages associated with the DD=TAPEHOLD. If so, correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

VFIVSAMI BKUPTBL WAS >>NOT<< UPDATED!!!!

Explanation:
BRNVBDSS has detected an error while running BKUPTBL update program VFIVSAMI.

System action:
BRNVBDSS ends processing and generate a return code of 16.

User response:
Check the job log to see if there are any IEC type messages associated with the DD=BKUPTBL. If so, correct the error and rerun BRNVBDSS. If the problem reoccurs, contact Technical Support.

BRNVBFDR Messages

There are three types of messages that can be generated by the BRNVBFDR program:

- Informational message. Reports input, delete, and final program results. These require no user response.

- Warning message. Identifies processing concerns such as input edit errors. These warnings does not end program processing but generates a return code of 4 that might require user intervention to resolve.
- Error message. A serious error that causes BRNVBFDR processing to end with return code of 8 or greater.

BRNVBFDRI001 INPUT ARCB1DSN: <dsname)>

Explanation:
This message displays the BRNVBFDR ARCB1DSN data set requested in the execution JCL. This value can be a fully qualified data set name or relative-reference GDG.

User response:
None. Informational message.

BRNVBFDRI002 USING ARCB1DSN: <dsname)>

Explanation:
This message displays the BRNVBFDR ARCB1DSN fully qualified data set name that the program determines belongs to this execution. The <dsname> BRNVBFDR decides to use has been matched to the FDR305 record VOLSER and FILESEQ in DD FDROUT.

User response:
None. Informational message.

BRNVBFDRI098 NO RECORDS SELECTED FOR BKUPTBL PROCESSING. RC:0004

Explanation:
This message signals that BRNVBFDR found no data selected for backup processing.

User response:
None. Informational message.

BRNVBFDRI099 NO ERROR MESSAGES FOUND, RETURN CODE=0000

Explanation:
This message signals the end of BRNVBFDR processing and displays the final return code.

User response:
None. Informational message.

BRNVBFDRW005 CHECK ARCB1DSN: <dsname> NOT A MATCH RC=0004

Explanation:
This message displays the BRNVBFDR ARCB1DSN fully qualified GDG data set name that the program has checked for belonging to this execution. BRNVBFDR rejects <dsname> because its catalog information does not match the FDR305 record VOLSER/FILESEQ in DD FDROUT.

System action:
BRNVBFDR bypasses the <dsname> listed and checks the prior relative reference GDG number (up to -4) for a match. Processing continues.

User response:
None. Informational message.

BRNVBFDRW006 LOCATE ARCB1DSN=<dsname)> NOT CATALOGED RC=0004

Explanation:
This message displays the BRNVBFDR ARCB1DSN relative-reference GDG dataset that the program has checked for belonging to this execution. BRNVBFDR rejects <dsname> because it is not cataloged.

System action:
BRNVBFDR has determined that it is at the end of the relative-reference GDG chain. It bypasses the <dsname> listed and retrieves the partial dataset name from the FDR305 record in DD=FDROUT. This is used for BKUPTBL update. Processing continues.

User response:
None. Informational message.

BRNVBFDRW007 DD=FDROUT REC=FDR305 VSER=vvvvvv FSEQ=#### RC=0004

Explanation:
This message displays the BRNVBFDR record=FDR305 VOLSER and FILESEQ details for the prior W002 ARCB1DSN=<dsname>.

System action:
BRNVBFDR has been unsuccessful in matching a relative-reference GDG <dsname> to the current backup. It uses the data in FDR305 record to update the backup table for this execution and return code=0004. Processing continues.

User response:
Determine why the GDG requested by ARCB1DSN does not exist. Correct the problem and rerun the job. If the problem reoccurs, contact Technical Support.

BRNVBFDRE011 ALLOCATE ERROR DD INPARM

Explanation:
BRNVBFDR has detected a system error while allocating the DD=INPARM dataset.

System action:
BRNVBFDR ends processing and generates a return code of 8 or higher.

User response:
Rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 INSTREAM PARAMETER MISSING 2

Explanation:
BRNVBFDR has detected that the in-stream parameters are missing.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the JCL to see if the input parameters have been omitted. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 INSTREAM PARAMETER ERROR 3

Explanation:
BRNVBFDR has detected an error while processing the in-stream parameters.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the JCL to determine which input parameters are in error. Correct the errors and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 OPEN ERROR DD INPARM 4

Explanation:
BRNVBFDR has detected an error while opening DD=INPARM.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the DD=INPARM. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 PARM ERROR AID= MUST BE <= 8 5

Explanation:
BRNVBFDR has detected a syntax error while processing application id parameter AID=.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the JCL and make sure that the APPLID parameter is from 1 to 8 characters long. Correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 PARM ERROR SID MUST BE <= 8 6

Explanation:
BRNVBFDR has detected a syntax error while processing sub-application id parameter SID=.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the JCL and make sure that the SUBAPPID parameter is from 1 to 8 characters long. Correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 PARM ERROR UIN= MUST BE <= 12 7

Explanation:
BRNVBFDR has detected a syntax error while processing user id parameter UIN=.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the JCL and make sure that the USERID parameter is from 1 to 12 characters long. Correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 PARM ERROR CPY= MUST BE 1 OR 8 2

Explanation:
BRNVBFDR has detected a syntax error while processing copy parameter CPY=.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the JCL and make sure that the copy parameter is either 1 or 2. Correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE01 RECORD ERROR FDR301 NEW DATE <> E= 9

Explanation:
BRNVBFDR has detected a syntax error while processing DD=FDROUT REC=FDR301.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:

Contact Technical Support.

BRNVBFDRE020 RECORD ERROR FDR301 NEW DATE <> yyyy.ddd

Explanation:
BRNVBFDR has detected a syntax error while processing DD=FDROUT REC=FDR301.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Contact Technical Support.

BRNVBFDRE021 RECORD ERROR FDR301 OLD DATE <> yy.ddd

Explanation:
BRNVBFDR has detected a syntax error while processing DD FDROUT REC=FDR301.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Contact Technical Support.

BRNVBFDRE022 RECORD ERROR FDR311 DSN TEXT NOT FOUND

Explanation:
BRNVBFDR has detected a syntax error while processing DD FDROUT REC=FDR311.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Contact Technical Support.

BRNVBFDRE023 RECORD ERROR FDR311 ARCHIVE NODE NOT FOUND

Explanation:
BRNVBFDR has detected a dataset name error while processing DD=FDROUT REC=FDR311.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Contact Technical Support.

BRNVBFDRE024 LOCATE ERROR VOLUME COUNT = 0

Explanation:
BRNVBFDR has detected a system error while processing ARCB1DSN=<dsname>.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the ARCB1DSN=<dsname>. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE025 LOCATE ERROR VOLUME COUNT > 20

Explanation:
BRNVBFDR has detected a system error while processing parameter ARCB1DSN=<dsname>.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the ARCB1DSN=<dsname>. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE030 CSI ERROR LINK DSN=<dsname>

Explanation:
BRNVBFDR has detected a system error while processing DSN=<dsname>.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the DSN=<dsname>. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE031 CSI ERROR DATAENTRY DSN=<dsname> (###/###)

Explanation:
BRNVBFDR has detected a system error while processing DSN=<dsname>.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the DSN=<dsname>. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE032 CSI ERROR DATA AREA DSN=<dsname> (###/###)

Explanation:

BRNVBFDR has detected a system error while processing DSN=<dsname>.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the DSN=<dsname>. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE033 CSI ERROR DSN NOT CATALOGED, DSN=<dsname>

Explanation:
BRNVBFDR has detected that required DSN=<dsname> is not cataloged.

System action:
BRNVBFDR ends processing and generates a return code of 8.

User response:
Check the job log to see if there are any IECxxxxx messages associated with the DSN=<dsname>. If so, correct the error and rerun BRNVBFDR. If the problem reoccurs, contact Technical Support.

BRNVBFDRE099 ERROR MESSAGES FOUND, RETURN CODE=####

Explanation:
At program end BRNVBFDR reports the highest final error return code.

System action:
BRNVBFDR ends processing and reports on the return code.

User response:
Refer to the BRNVBFDRE0## error causing the return code. Correct the error and rerun BRNVBFD. If the problem reoccurs, contact Technical Support.

BRNVBFVR Messages

This section documents the BRNVBFVR messages.

ERROR01 ERRORS IN INPUT PARAMETERS

Explanation:
An error occurred when BRNVBFVR was editing input parameters <APPLID,SUBAPPID>. APPLID is required. SUBAPPID is optional. Both parameters must be 8 characters or less.

User response:
Make sure that the parameters are correct. Rerun the job.

ERROR03 IMPOSSIBLE TO READ TAPEHOLD DD

Explanation:
An error occurred when BRNVBFVR tried to read the DMS tape.

User response:
Contact Technical Support.

BRNVYNEX Messages

This section documents messages from the BRNVYNEX and BRNVLEGX programs. All of these messages are issued to the job log and to the file defined by the REPORT DD statement if it is allocated.

BRNVYNEX01I BRNVYNEX STARTED

Explanation:
This message is issued at the start of the BRNVYNEX module.

User response:
None. Informational message.

BRNVYNEX02I BRNVYNEX TERMINATED

Explanation:
This message is issued when the process is complete.

User response:
None. Informational message.

BRNVYNEX03I SYS.IEFUJV EXIT NOT DEFINED

Explanation:
This message is issued if the system exit IEFUJV is not defined to the active SMFPRMxx member. For this facility to process successfully the JES system exit must be available.

User response:
Add IEFUJV to the EXITS list in the SMFPRMxx member at your installation.

BRNVYNEX04I SYS.IEFUJV EXIT LIST ERROR R0(XXXXXXXX) R15 (XXXXXXXX)

Explanation:
This message is issued if the system macro CSVDYNEX macro receives a non-zero return code. CSVDYNEX macro return and reason codes can be found in the Authorized Assembler Reference manual. Register 0

contains the reason code and register 15 contains the return code.

User response:
Evaluate the return and reason codes, correct and rerun.

BRNVYNEX05I REPORT FILE OPEN ERROR, BYPASSED

Explanation:
This message is issued if the REPORT file open failed. In that case, messages are written to the job log only.

User response:
Add REPORT DD to the job stream.

BRNVYNEX06I EXAAH#REC NOT = 1

Explanation:
This message is issued if information on more than one exit is returned on the LIST request. It should not occur, since the request is for SYS_IEFUJV only.

User response:
Contact Technical Support.

BRNVYNEX07I EXAAENAME NOT = SYS_IEFUJV

Explanation:
This message is issued if the information returned from the LIST request is not SYS_IEFUJV. This should not occur.

User response:
Contact Technical Support.

BRNVYNEX08I EXAAEFIRSTENT@ IS 0

Explanation:
This message is issued if the LIST pointers are invalid.

User response:
Contact Technical Support.

BRNVYNEX10I SYS.IEFUJV EXIT DELETE ERROR R0(XXXXXXXX) R15(XXXXXXXX)

Explanation:
This message is issued when the DELETE request for program VFIUJV at exit SYS.IEFUJV fails. The return and reason codes are returned in registers 15 and register 0 respectively.

User response:
Evaluate the return and reason code, correct and rerun.

BRNVYNEX11I SYS.IEFUJV EXIT ADD ERROR R0(XXXXXXXX) R15(XXXXXXXX)

Explanation:
This message is issued when the ADD request for program VFIUJV at exit SYS.IEFUJV fails. The return and reason codes are returned in registers 15 and register 0 respectively.

User response:
Evaluate the return and reason codes, correct and rerun.

BRNVYNEX12I SYS.IEFUJV EXIT ACTIVATE ERROR R0(XXXXXXXX) R15(XXXXXXXX)

Explanation:
This message is issued when the ACTIVATE request for program VFIUJV at exit SYS.IEFUJV fails. The return and reason codes are returned in registers 15 and register 0 respectively.

User response:
Evaluate the return and reason codes, correct and rerun.

BRNVYNEX13I SYS.IEFUJV EXIT PROGRAM (VFIUJV) ADD SUCCESSFUL

Explanation:
This message is issued when the ADD for VFIUJV to the exit SYS.IEFUJV is successful.

User response:
None.

BRNVYNEX14I SYS.IEFUJV EXIT PROGRAM(VFIUJV) ACTIVATE SUCCESSFUL

Explanation:
This message is issued when the ACTIVATE for VFIUJV to the exit SYS.IEFUJV is successful.

User response:
None.

BRNVYNEX15I SYS.IEFUJV EXIT DSN=

Explanation:
This message documents the request data set that contains the exit module VFIUJV.

User response:
None.

BRNVYNEX16I DSN PARAMETER INVALID

Explanation:
The data set specified in the PARM card is invalid.

User response:
Correct DSN specified, and rerun.

BRNVYNEX17I SYS.IEFUJV EXIT PROGRAM (VFIUJV) DELETE SUCCESSFUL

Explanation:
This message is issued when the DELETE for VFIUJV to the exit SYS.IEFUJV is successful.

User response:
Evaluate the return and reason codes, correct and rerun.

BRNVYNEX18I MODIFY COMMAND INVALID, DATA ENTERED(XXXXXXXXXX)

Explanation:
This message is issued if the modify command is other than STOP or RELOAD. The invalid test is returned in the message.

User response:
Correct the command and rerun.

BRNVYNEX19I STOP COMMAND ISSUED

Explanation:
This message is issued when a STOP command is issued to BRNVYNEX.

User response:
None. Informational message.

BRNVYNEX20I RELOAD COMMAND ISSUED

Explanation:
The RELOAD command was issued. BRNVYNEX reloads the table using the LEGFILE.

User response:
None.

BRNVYNEX21I OPEN ERROR LEGTABLE FILE

Explanation:
This message is issued if the LEGFILE OPEN request fails.

User response:
Evaluate the reason for the open failure, correct and rerun.

BRNVYNEX22I LOAD LEGFILE ERROR

Explanation:
This message is issued if the request to reload the table fails.

User response:
Correct the failure and rerun.

BRNVYNEX23I GETMAIN ERROR, LENGTH(XXXXXXXX) R15(XXXXXXXX)

Explanation:
This message is issued when there is an error on a GETMAIN request. The length of the request and the return code are documented.

User response:
Make additional storage available to the region and rerun.

BRNVYNEX24I LEG LOAD OK, ENTRIES(00000000) ADDR(00000000) LENGTH(00000000)

Explanation:
This message is issued when the LEGFILE is successfully loaded into ECSA. The number of entries, the address and the length are listed.

User response:
None.

BRNVYNEX25I NS CREATE ERROR R15(XXXXXXXX)

Explanation:
This message reflects an error creating the name storage: BRNVYNEX_HDRPTR. The return code is listed. The named storage contains a pointer to the ECSA table storage.

User response:
Correct the reason that Named Storage request failed and rerun.

BRNVYNEX26I NS STORAGE FREED ADDR(00000000) LENGTH(00000000)

Explanation:
This message is issued at task termination when the named token storage is released.

User response:
None. Informational message.

BRNVYNEX27I NS RELOAD COMPLETE

Explanation:
This message is issued when a RELOAD command completes successfully.

User response:
None. Informational message.

BRNVYNEX28I NS DELETE ERROR R15(XXXXXXXX)

Explanation:
This message is issued when the request to release named storage fails. The contents of register 15 are listed.

User response:
Determine the Named Storage request error, correct and rerun.

BRNVYNEX29I NS GET ADDR(00000000) LENGTH(00000000)

Explanation:
This message is issued during task initiation. It lists the address and length of the named storage.

User response:
None.

BRNVYNEX30I LEG STORAGE FREED ADDR(00000000) LENGTH(00000000)

Explanation:

This message is issued when the LEGFILE is reloaded and at task termination. The address and length of the ECSA storage is listed.

User response:
None.

BRNVYNEX30E BRNVYNEX NOT APF AUTHORIZED

Explanation:
The module BRNVYNEX must be executed from an authorized library. If not, this error message results.

User response:
Ensure that BRNVYNEX is in an authorized library and all the libraries within the STEPLIB are authorized. Rerun.

BRNVYNEX31I ESTAE SET ERROR R0(XXXXXXXX) R15(XXXXXXXX)

Explanation:
In order to ensure ECSA storage is freed in the event BRNVYNEX abends, an ESTAE is established. This message is issued if the ESTAE request fails. The reason and return codes are listed.

User response:
Correct the error and rerun.

BRNVYNEX32I BRNVYNEX ESTAE ENTERED

Explanation:
This message is issued if the ESTAE was entered. In this event, BRNVYNEX has abended and the exit is removed and storage is freed before terminating.

User response:
None.

BRNVYNEX33I ALLOCATION ERROR ON LEGTABLE, REASON CODE(XXXXXXXX)

Explanation:
This message is issued when there is an allocation error on the LEGFILE. The reason code is listed.

User response:
The most common reason is that the file is already allocated to another task. When the other task ends, retry the RELOAD if BRNVYNEX is already active.

BRNVYNEX34I INFILE OPEN ERROR

Explanation:
This message is issued if the INFILE open fails. The INFILE data set contains the LEGFILE data set name.

User response:
Evaluate the open error and rerun.

BRNVYNEX37I LEGFILE EMPTY

Explanation:
This message is issued if there is no record in the INFILE. This is where the facility determines the LEGFILE to use to load the table.

User response:
Correct INFILE and rerun.

BRNVYNEX38I LEGFILE RELOAD FAILED

Explanation:
This message is issued if the request to reload the LEGFILE fails.

User response:
Correct the error and rerun.

BRNVYNEX39I <<HEAD>> NOT FOUND, INVALID LEGFILE

Explanation:
This message indicates the LEGFILE is not formatted correctly.

User response:
Re-execute program BRNVLEGX. Correct LEGFILE contents and rerun.

BRNVYNEX40I <<HEAD>> ENTRIES INVALID

Explanation:
This message indicates the LEGFILE is not formatted correctly.

User response:
Re-execute program BRNVLEGX. Correct LEGFILE contents and rerun.

BRNVLEGX Messages

This section documents the BRNVLEGX Messages.

BRNVLEGX01 xxxxxxxx OPEN ERROR

Explanation:
This message is issued if an open error occurs on one of the files where xxxxxxxx represents the DD name for the file in error.

User response:
Correct the OPEN error and rerun.

BRNVLEGX02 PROGRAM XLATE ERROR, JOB(xxxxxxxx) OLD PGM(xxxxxxxx) WHICH (xxx) SORTED RECORD NUM (99999999)

Explanation:

This message is issued when an input record fails verification; that is, either the old program or new program does not match an entry in the internal translate table. The job name, old and new program names, and record number of the sorted input record are listed. Error message BRNVLEGX05 follows and list the offending card text.

User response:
Correct the card in error and rerun.

BRNVLEGX03 LEGFILE FORMAT COMPLETE: RECORDS IN(nnnn) LEGFILE RECORDS OUT(nnnn) TABLE ENTRIES(nnnn)

Explanation:
This message is issued at the end of execution. It reports the number of input records and the number of LEGFILE records. In addition, the number of table entries is also reported

User response:
None. Informational message.

BRNVLEGX04 NO VALID ENTRIES IN INPUT FILE

Explanation:
This message is issued when there are no valid translate cards in the input stream.

User response:
Correct the input file and rerun.

BRNVLEGX05 INVALID RECORD--- >'xxxxxxxxxxxxxxxxxxxx<---

Explanation:
This message follows BRNVLEGX02 on a program translate error. It lists the offending record.

User response:
The response is to message BRNVLEGX02; that is, correct the card in error and rerun.

BRNVGENR Messages

This section documents messages from the BRNVGENR program, which is used to copy tape or disk data sets to tape.

BRNTCPY001I <id> <timestamp> Build: Version <version> <build time>

Explanation:
BRNVGENR has started.

User response:
None. Informational message.

BRNTCPY100E <id> <timestamp> Input DDname=<xxxxxx> is not available or incorrect type.

Explanation:
BRNVGENR has determined that an input DDname xxxxxx is not allocated or is not a tape device.

User response:
Correct and restart the job.

BRNTCPY101E <id> <timestamp> Output DDnames are not available or incorrect type.

Explanation:
BRNVGENR has determined that one or all output DDnames were not allocated or are allocated to unsupported device type: input: seq. DASD data sets or tape output: tape.

User response:
Correct and restart the job.

BRNTCPY102W <id> <timestamp> Print DDname is not available, WTO is used instead.

Explanation:
The DDname for the print file was not supplied.

User response:
Correct if desired and restart the job.

BRNTCPY103I <id> <timestamp> Output DDnames found: <X>; the last one is <xxxxxxxx>

Explanation:
BRNVGENR determined that X output DDnames were allocated and usable, and the last DDname is xxxxxxxx

User response:
None.

BRNTCPY107I <id> <timestamp> DDname MAXBLK: val1 OPTBLK: val2 UCBTYP: val3

Explanation:
BRNVGENR reports the maximum allowed block size for the device allocated to <DDname>, the optimum block size for this device as well as the UCB type for it.

User response:
None. Informational message.

BRNTCPY108E <id> <timestamp> Error from OPEN for DDname = <xxxxxx>

Explanation:
BRNVGENR determined that there was an error after OPEN macro for DDname <xxxxxx>.

User response:

Review the other messages; contact Technical Support.

BRNTCPY109E **<id> <timestamp> I/O Errors for DDname = <xxxxxx> ; messages were issued.**

Explanation:
BRNVGENR determined that there was an error after read or write operation for DDname <xxxxxx>. Other messages can be issued using WTO and must be reviewed.

User response:
Review any other messages. Correct the space problem and restart. Report the command reject messages to Technical Support.

BRNTCPY110E **<id> <timestamp> Unsupported RECFM, DCBRECFM = xxx**

Explanation:
BRNVGENR determined that the RECFM=<xxx> is not supported for an input DASD data set.

User response:
Review any other messages; contact Technical Support.

BRNTCPY111E **<id> <timestamp> Blocksizes for input DDname <xxxxxx> and output DDname <yyyyyy> are incompatible**

Explanation:
BRNVGENR determined that maximum supported block size for an output device is less than maximum blocksize of an input device.

User response:
Use a different output device if the job step terminated with return code 16.

BRNTCPY112I **<id> <timestamp> Processing <completed terminated <mismatch> > ; <NNNNNNNN>x blocks were read, Maxblksi=<mmmmmmm>x**

Explanation:
The BRNVGENR copy operation is finished. If the word 'terminated' is in the message text, errors were encountered and other messages are present in the syslog and job log. The size of the largest input data set block is reported in hex. This message is also used by the VTCCMP (Compare program). If <mismatch> is displayed, a BLKSIZE/LRECL /RECFM mismatch is found, but data has been compared to be exactly the same.

User response:
Review the other error messages. They might indicate a bad tape, and other correctable conditions. Report the command reject messages to Technical Support, as well as any other suspicious messages.

BRNTCPY113W **<id> <timestamp> Unknown RECFM, DCBRECFM = <XX> process as RECFM=U**

Explanation:
BRNVGENR could not determine the RECFM for an input DDname and uses internal logic for RECFM U. Performance is degraded in most cases.

User response:
Specify the correct RECFM in the JCL.

BRNTCPY114I **<id> <timestamp> Processing <completed terminated> ; <NNNNNNNN>x blocks were written.**

Explanation:
BRNVGENR copy operation is finished. If the word 'terminated' is in the message text, errors were encountered and other messages are present in the syslog and joblog.

User response:
Review any other error messages. They might indicate a bad tape, insufficient space, and other correctable conditions. Report the command reject messages to Technical Support, as well as other suspicious messages.

BRNTCMP115E **<id> <timestamp> Blocks size mismatch, Block <xxx>; UT1 = <nnnnnn> UT2 = <mmmmmm>**

Explanation:
The tape compare program determined that the input and output block sizes do not match while comparing the contents of two tape data sets. xxx is the hexadecimal block number relative to the current volume for the input data set. nnnnnn is the SYSUT1 data set block size and mmmmmm is the SYSUT2 data set block size.

User response:
If the output tape was produced by BRNVGENR, contact Technical Support. Otherwise review the tape copy procedures; it is possible that IEBGENER re-blocked the input tape to utilize SDB.

BRNTCMP116E **<id> <timestamp> Premature EOF for <xxxxxx>; Block <nnn>x.**

Explanation:
The tape compare program encountered end-of-file for DDname xxxxxx while the other tape had more data. nnn is the hexadecimal block number relative to the current volume for the specified DDname.

User response:

If the output tape was produced by BRNVGENR, contact Technical Support; otherwise review the tape copy procedures.

BRNTCPY117I <id> <timestamp> DDname: <xxxxxx> Volumes processed: <n>

Explanation:
BRNVGENR processing is completed, see message "BRNTCPY112I" on page 361. This message is issued for an input DDname and for all output DDnames.

User response:
None. Informational message.

BRNTCPY118E <id> <timestamp> Input BDW is invalid: <xxxxxxxx>

Explanation:
BRNVGENR has determined from the input data set parameters that the RECFM=V logic path is to be used. While validating the record format, it determined that BDW (block descriptor word) does not conform to a published specification. The value of BDW is reported in hexadecimal.

User response:
Review the JCL parameters. For example, specifying RECFM=V for a data set with RECFM=F produces this message.

BRNTCPY120I <id> <timestamp> Internal limit <NNN> exceeded - <MMM>

Explanation:
BRNVGENR has found an internal logic error. Execution is not affected.

User response:
Report this message to Technical Support. You may be asked to rerun the job.

BRNTCMP123E <id> <timestamp> Records mismatch, Block <xxx>;

Explanation:
Tape compare program determined that blocks data do not match while comparing the contents of two tape data sets. xxx is the hexadecimal block number relative to the current volume for the input data set.

User response:
If the output tape was produced by BRNVGENR, contact Technical Support. Otherwise review tape copy procedures; it is possible that IEBGENER re-blocked the input tape to utilize SDB.

BRNTCPY127E <id> <timestamp> Error from <nnnnnn> macro; rc = <nnn> rsn = <mmm>

Explanation:
BRNVGENR's attempt to locate a UCB for an input DDname failed other messages might follow.

User response:
Review the documentation for a reported macro. Report this error to Technical Support.

BRNTCPY128I <id> <timestamp> All datasets were opened; R=xxxx B=NNNNx L=MMMMx

Explanation:
BRNVGENR successfully opened all input and output data sets. xxxx is an internal value, NNNN is the block size in hexadecimal for an input data set, and MMMM is the LRECL for it as provided.

User response:
None. Informational message.

BRNTCPY129I <id> <timestamp> DDname: nnnnnnn EXCP Count: mmmmmm

Explanation:
VFIGENR completed the tape copy process. The job step EXCP count for each DDname is reported and it is the same as in the associated SMF30 record.

User response:
None. Informational message.

BRNTCPY130I <id> <timestamp> DDname: value DSNAME: value

Explanation:
BRNVGENR reports the data set name that is allocated to a particular DDname.

User response:
None. Informational message.

BRNTCPY131I <id> <timestamp> DDname: value CYL: xxxx T2C: xx FLG: NNx ...

Explanation:
BRNVGENR reports the number of cylinders for the device allocated to DDname as well as the number of tracks per cylinder for it.

User response:
None. Informational message.

BRNTPRD135I <id> <timestamp> DDname: value Ext: nn Vol: xxxxxx DASD: value

Explanation:
BRNVGENR reports miscellaneous information for an input DASD data set extent being processed.

User response:
None. Informational message.

BRNTCPD139E <id> <timestamp> DDname: <ddddd> has unsupported format - <text>

Explanation:
BRNVGENR determined that an attempt was made to read the DASD data set allocated to DDname:

<ddddd>, which has an unsupported format. The <text> can be: - `Keylength not zero` - `DSORG is not PS`

User response:
Use a different program to copy the file.

BRNTPRD141I **<id> <timestamp> DDname: value Volser: value DS1LSTAR: value MBBCCHHR: value**

Explanation:
BRNVGENR reports misc values for the DASD input data set.

User response:
None. Informational message.

BRNTCPY142I **<id> <timestamp> DDname: value Addr: cuu LastVolser: xxxxxx EXCPcount: nnnnn**

Explanation:
BRNVGENR completed the tape copy process. The Job Step EXCP count for each DDname is reported, as well as the last VOLSER used and the corresponding device address for it. It should be the same as in the associated SMF30 record.

User response:
None. Informational message.

BRNTCPY144I **<id> <timestamp> Dsorg: ....**

Explanation:
BRNVGENR reports miscellaneous parameters for the input data set, such as Dsorg, RECFM, block size (dcbe, used and device).

User response:
None. Informational messages.

BRNTCMP145I **<id> <timestamp> PGM=IEBCOMPR is used when input dataset resides on dasd.**

Explanation:
The Dataset Compare (VTCCMP) program issued this message.

User response:
None. Informational message.

BRNTCMP146I **<id> <timestamp> PGM=IEBCOMPR completed with RC= value**

Explanation:
The Dataset Compare (VTCCMP) program issued this message, when IEBCOMPR finished its processing.

User response:
None. Informational message.

BRNTCPE147W **<id> <timestamp> DD: value : The maximum tape block size <value> is larger than maximum Device block size <value>**

Explanation:
While copying RECFM=U data sets, BRNVGENR determined that the maximum block size for the data set on an input tape device is larger than the maximum block size reported for the output tape device (by Devtype macro) and the maximum block size as read from the input data set is larger than the maximum block size reported for the output tape device. BRNVGENR completed the copy with rc=6.

User response:
Review the JCL. Warning: If an output tape must be read by a program using QSAM or BSAM from this output device, errors might result. Unless a special program is used to read this tape, rerun the job or use BRNVGENR to copy this data set to another device.

BRNTCPE149I **<id> <timestamp> DDname: <DDname> DSNAME: <dsname>**

Explanation:
BRNVGENR reports the input or output data set name at the end of a copy or compare process.

User response:
None. Informational message.

BRNTCPE150I **<id> <timestamp> DDname: <DDname> FSEQN: <n> DEVTYPE: <d> VOLCOUNT: <n>**

Explanation:
BRNVGENR reports the file sequence number, device type, and volume count at the end of the copy or compare process, for input and output data sets.

User response:
None. Informational message.

BRNTCPE151I **<id> <timestamp> DDname: <DDname> VOLSER: *volser* ... *volser***

Explanation:
BRNVGENR reports all *volser*s at the end of the copy or compare process, for input and output data sets.

User response:
None. Informational message.

VTCCPE152I **<id> <timestamp> All datasets were opened; R1= ....**

Explanation

Reported RECFM for input (R1) and output (R2), Block size (B1/B2), LRECL (L1/L2) at the end of compare process. If a mismatch is found, message VTCCPE153E follows. For example:

```
VTCCMP152I VTCCMP 10:05:39.172 All datasets
were opened; R1=FB B1=00005A19x L1=0293x <>
```

```
R2=FB B2=00005A19x L2=0293x

VTCCPE153E <id> <timestamp> BLKSIZE/LRECL /
RECFM mismatch, see VTCCMP152I
```

User response:
None. Informational message.

VTCCPE153E **<id> <timestamp> BLKSIZE/LRECL /RECFM mismatch**

Explanation:
Follows message VTCCPE152I if a mismatch occurs with BLKSIZE/LRECL /RECFM.

User response:

BRNTCPE154I **<id> <timestamp> Bytes written out: *hhhhhhhhx* (+*nnnnnnnnd*)**

Explanation:
Reports the total number of bytes written to all output data sets in hex and decimal.

User response:
None. Informational message.

BRNTCPY993E **!!! SYNADAF SMS msg: <message text>**

Explanation:
BRNVGENR encountered an I/O error. This is a WTO-only message.

User response:
Review the message. Report the command reject messages to Technical Support, as well as other suspicious messages.

BRNTCPY994E **!!! SYNADAF msg: <message text>**

Explanation:
BRNVGENR encountered an I/O error. This is a WTO-only message.

User response:
Review the message. Report the command reject messages to Technical Support, as well as other suspicious messages.

BRNTCPY995E **!!! SYNAD entered, SYNADAF R0=<nnn>x, R1=<mmm>x**

Explanation:
BRNVGENR encountered an I/O error. This is a WTO-only message. Another message follows.

User response:
Review other messages.

BRNTCPY996E **!!! Buffer address=<xxx>x, No. of Bytes Read=<nnn>x**

Explanation:
BRNVGENR encountered an I/O error. This is a WTO-only message. Another message might be present.

User response:
Review other messages.

HISMAINT Messages

The following messages may be issued by program HISMAINT during maintenance processing for IBM Z Batch Resiliency AUDIT and HISTORY databases.

BRNVMNT **VER 1.1.0 mm/dd/yyyy hh:mm:ss TERACLOUD S.A. AND IBM CORP.**

Explanation:
This message is the header to the HISMAINT report in DD=ERRORS.

User response:
None. Informational message.

BRNVMNTI01 **PROGRAM COMPLETED SUCCESSFULLY**

Explanation:
This message signals completion of HISMAINT.

User response:
None. Informational message.

VFIMAINTE01 **SORT ERROR (RC=####) - PROGRAM TERMINATED**

Explanation:
This message reports that an error occurred processing the sort parameter.

System action:
Message generated with return code 8. Program ends.

User response:
Contact Technical Support with your execution listing.

VFIMAINTE02 **%%%%%%% PARAMETER MISSING - SEE MESSAGES DD FOR DETAILS**

Explanation:
This message reports that required retention parameters in VFIPARMS are missing.

System action:
Message generated with return code 8. Program ends.

User response:

Correct DD=VFIPARMS retention period parameter. Rerun the program. For parameter details, see Maintaining the HISTORY and AUDIT data sets (HISMAINT).

VFIMAINTE03 RETENTION PARAMETER NOT NUMERIC

Explanation:
This message reports one of the retention parameters in VFIPARMS is not numeric.

System action:
Message generated with return code 8. Program ends.

User response:
Correct DD=VFIPARMS retention period parameter. Rerun the program. For parameter details, see Maintaining the HISTORY and AUDIT data sets (HISMAINT).

IZBR/RTS Messages

This section documents the IZBR/RTS (Real-Time Dynamic Dataset Scheduler) messages.

BRNVC00201I mm/dd/yy hh:mm:ss - SCHDJDS DCB AREA OBTAINED - ADDRESS = xxxxxxxx, SUB-POOL/LENGTH = yyzzzzzz

Explanation

Issued by BRNVC002 when staging journal SCHDJDS DCB area is established where:

- xxxxxxxx - DCB area location
- yy - storage sub pool
- zzzzzz - storage length

User response:
None. Informational message.

BRNVC00202I mm/dd/yy hh:mm:ss - SCHDJDS FILE OPENED FOR OUTPUT

Explanation:
Issued by BRNVC002 when staging journal SCHDJDS DCB is opened for output.

User response:
None. Informational message.

BRNVC00203I mm/dd/yy hh:mm:ss - SCHDJDS FILE CLOSED, n,nnn,nnn RECORDS WRITTEN TO journal.dsname

Explanation:
Issued by BRNVC002 when staging journal SCHDJDS records are moved to the schedule journal and SCHDJDS file is closed.

User response:
None. Informational message.

BRNVC00204E mm/dd/yy hh:mm:ss - GETMAIN ERROR FOR SCHDJDS DCB AREA, SCHEDULE JOURNALING DISABLED

Explanation:
Issued by BRNVC002 when acquiring storage for SCHDJDS DCB area. A GETMAIN error occurred while acquiring storage for SCHDJDS DCB area.

User response:
Determine the problem and correct it or contact Technical Support.

BRNVC00205E mm/dd/yy hh:mm:ss - UNABLE TO OPEN SCHDJDS DATASET, SCHEDULE JOURNALING DISABLED

Explanation:
Issued by BRNVC002 when attempting to open SCHDJDS staging journal file. The error occurred while attempting to open the file.

User response:
Determine the problem and correct it or contact Technical Support.

BRNVC00206E mm/dd/yy hh:mm:ss - UNABLE TO COPY SCHDJDS RECORDS TO SCHDJRNL *<reason>*

Explanation

Issued by BRNVC002 when copying journal records from staging journal SCHDJDS to the schedule journal file, where *reason* is one of the following:

SCHDJDS DCB ADDRESS INVALID
OPEN FAILED FOR SCHDJDS
CLOSE FAILED FOR SCHDJDS
OPEN FAILED FOR SCHDJRNL
SCHDJRNL ALLOCATION ERROR

User response:
Determine the problem and correct it or contact Technical Support.

BRNVC00207I mm/dd/yy hh:mm:ss - GETMAIN ERROR FOR JOURNAL BUFFERS, SCHEDULE JOURNALING DISABLED

Explanation:
Issued by BRNVC002 when acquiring storage for journal buffers. Schedule journaling is disabled.

User response:
None.

BRNVC00901E **mm/dd/yy hh:mm:ss - BRNVC009 GCOUNT DISABLED, VSAM ERROR(xxxxxxxx): R15(XXXXXXXX) R0(XXXXXXXX) REASON CD/C21VS/$VSAMIO**

Explanation
This error is issued for a VSAM error. The error text designates one of the following VSAM functions:

POINT, GETNEXT, GET B, PUT B

Additionally, the contents of register 15 and register 0 are documented.

User response:
Evaluate reason codes according to C21VS macro.

BRNVLSCH01I **mm/dd/yy hh:mm:ss - SCHEDULE LOAD/RELOAD STARTED**

Explanation:
Issued by BRNVLSCHD when loading/reloading the schedule for Dynamic Dataset Scheduling.

User response:
None. Informational message.

BRNVLSCH02I **mm/dd/yy hh:mm:ss - SCHEDULE EVENT ARRAYS LOADED - n,nnn,nnn IRKA, n,nnn,nnn SEA/J, n,nnn,nnn SEA/T**

Explanation:
Issued by BRNVLSCHD after a load/reload of the schedule for Dynamic Dataset Scheduling.. Displays the number of entries loaded into each of the three Schedule Event Arrays IRKA, SEA/J and SEA/T.

User response:
None. Informational message.

BRNVLSCH03I **mm/dd/yy hh:mm:ss - RECORD NOT SCHEDULED, Reason; REC: record-id**

Explanation
Issued by BRNVLSCHD for inventory records which are not scheduled, where Reason is one of the following:

- EXCL IN SRA
- NOT IN SRA
- NO BASE REC
- NOT CRITCAL
- NO BASE DSN

User response:
None. Informational message.

BRNVLSCH04I **mm/dd/yy hh:mm:ss - INVENTORY FILE UPDATED WITH SCHEDULE - n,nnn,nnn SCHEDULED, n,nnn,nnn NOT SCHEDULED**

Explanation:
Issued by BRNVLSCHD after a load/reload of the schedule for Dynamic Dataset Scheduling. Displays the total count of scheduled and non-scheduled inventory records.

User response:
None. Informational message.

BRNVLSCH05W **mm/dd/yy hh:mm:ss - REC NOT INSERTED SEA, REASON rc; REC: record-id**

Explanation
Warning message issued by BRNVLSCHD for inventory records which are not inserted into Schedule Event Arrays where rc is:

0001 – record had no valid scheduling events
0002 – invalid key for inventory record
0003 – inventory update failed for record

User response:
None.

BRNVLSCH06E **mm/dd/yy hh:mm:ss - SCHEDULE EVENT ARRAYS NOT LOADED – NO COMPONENT RECORDS IN INVENTORY FILE**

Explanation:
Issued by BRNVLSCHD during schedule load/reload.

User response:
Contact Technical Support.

BRNVLSCH07E **mm/dd/yy hh:mm:ss - SCHEDULE LOAD/RELOAD FAILURE – *reason***

Explanation
Issued by BRNVLSCHD during schedule load/reload where *reason* is one of the following:

- TABLE STORAGE EXHAUSTED FOR table
- NEXT FREE ADDR. INVALID FOR table
- INVALID HASH TIME FIELD FOR table
- FILE INVENTRY IS NOT OPENED
- OPEN FAILED FOR FILE filename, RC=xxxx, ERROR=xxxxxxxx
- GETKEY FAILED FOR FILE filename, RC=xxxx, ERROR=xxxxxxxx
- GETNEXT FAILED FOR FILE filename, RC=xxxx, ERROR=xxxxxxxx

- LOAD FAILED FOR MODULE xxxxxxxx
- Action: Contact technical support

User response:
Contact Technical Support.

BRNVLSCH08W **mm/dd/yy hh:mm:ss - INVENTORY QIO OPEN ERROR, RC=rc: DEFAULT FOR SCHEDULING GDGS WILL BE ALL GENERATIONS**

Explanation:
Warning message issued by BRNVLSCHD during load/reload of the schedule for Dynamic Dataset Scheduling. Unable to build QIO file and therefore schedule GDGs by GCOUNT but instead schedules all generations of GDG.

User response:
None.

BRNVLSEA01I **mm/dd/yy hh:mm:ss - SCHEDULE EVENT ARRAY (SEA) LOAD/RELOAD STARTED**

Explanation:
Issued by BRNVLSEA when loading/reloading the Schedule Event Arrays (SEA).

User response:
None. Informational message.

BRNVLSEA02I **mm/dd/yy hh:mm:ss - SCHEDULE EVENT ARRAYS LOADED - n,nnn,nnn IRKA, n,nnn,nnn SEA/J, n,nnn,nnn SEA/T**

Explanation:
Issued by BRNVLSEA after load/reload of the Schedule Event Arrays. Displays the number of entries loaded into each of the three Schedule Event Arrays IRKA, SEA/J and SEA/T.

User response:
None. Informational message.

BRNVLSEA05W **mm/dd/yy hh:mm:ss - REC NOT INSERTED SEA, REASON rc; REC: record-id**

Explanation
Warning message issued by BRNVLSEA for inventory records which are not inserted into Schedule Event Arrays where rc is either:

- 0001 – record had no valid scheduling events
- 0002 – invalid key for inventory record

User response:
None. Informational message.

BRNVLSEA06E **mm/dd/yy hh:mm:ss - SCHEDULE EVENT ARRAYS NOT LOADED – NO COMPONENT RECORDS IN INVENTORY FILE**

Explanation:
Issued by BRNVLSEA during load/reload of Schedule Event Arrays (SEA).

User response:
Contact Technical Support.

BRNVLSEA07E **mm/dd/yy hh:mm:ss - SCHEDULE EVENT ARRAY (SEA) LOAD FAILURE – *reason***

Explanation
Issued by BRNVLSEA during schedule load/reload of Schedule Event Arrays where *reason* is one of the following:

- TABLE STORAGE EXHAUSTED FOR table
- NEXT FREE ADDR. INVALID FOR table
- INVALID HASH TIME FIELD FOR table
- FILE INVENTRY IS NOT OPENED
- GETKEY FAILED FOR FILE filename, RC=xxxx, ERROR=xxxxxxxx
- GETNEXT FAILED FOR FILE filename, RC=xxxx, ERROR=xxxxxxxx
- LOAD FAILED FOR MODULE xxxxxxxx
- Action: Contact technical support

User response:
Contact Technical Support.

BRNVL01101I **mm/dd/yy hh:mm:ss - SCHDREQ ARRAY (SRA) LOADER STARTED**

Explanation:
Issued by BRNVL011 when loading/reloading the Schedule Request Array (SCHDREQ).

User response:
None. Informational message.

BRNVL01102I **mm/dd/yy hh:mm:ss SCHDREQ ARRAY LOADED - n,nnn,nnn INPUT, n,nnn,nnn OUTPUT, n,nnn,nnn ERROR**

Explanation:
Issued by BRNVL011 after a load/reload of the Schedule Request Array (SCHDREQ). Displays the count of the number of records read from SCHDREQ, number of records loaded into SCHDREQ table, and number of SCHDREQ records in error.

User response:
None. Informational message.

BRNVL01103E **mm/dd/yy hh:mm:ss SCHDREQ ERROR ec-rs, REC: schdreq entry**

Explanation

Issued by BRNVL011 when an error occurred during validation of Schedule Request Array (SCHDREQ) entry, where rs is the keyword number and ec is one of the following:

01 – parsing failed for SCHDREQ record
02 – length of keyword value too long
03 – keyword value invalid
04 – keyword value missing
05 – length of parameter too long
06 – keyword default value invalid
07 – option keyword value invalid

User response:
Check documentation on the syntax for SCHDREQ and correct the error.

BRNVL01104E **mm/dd/yy hh:mm:ss SCHDREQ LOAD FAILURE, REASON nnnn**

Explanation

Issued by BRNVL011 when failure occurred during load of Schedule Request Array (SCHDREQ) into storage, where nnnn is one of the following:

0008 – open failed for SCHDREQ
0012 – read failed for SCHDREQ
0016 – QIO put error for SCHDREQ

User response:
Determine the problem and correct it or contact Technical Support.

VFIRTS001I **mm/dd/yy hh:mm:ss – SCHEDULE CB OBTAINED - ADDRESS = xxxxxxxx, SUB-POOL/LENGTH = yyzzzzzz**

Explanation

Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask where:

- xxxxxxxx - Dynamic Dataset Scheduling Control Block area location
- yy - storage sub pool
- zzzzzz - storage length

User response:
None. Informational message.

VFIRTS002I **mm/dd/yy hh:mm:ss – SCHEDULE PARAMETERS READ FROM PARMLIB, MEMBER = *member***

Explanation:
Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask where *member* is the member containing Dynamic Dataset Scheduling parameters.

User response:
None. Informational message.

VFIRTS003E **mm/dd/yy hh:mm:ss – SCHDPARM CONTAINS UN-RECOGNIZED PARAMETERS – SEE 'SNAP' DATASET**

Explanation:
Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask when an unrecognized parameter was found in SCHDPARM. The SNAP data set contains the parameter in error.

User response:
Determine the problem and correct the parameter in error.

VFIRTS004E **mm/dd/yy hh:mm:ss – SCHDPARM CONTAINS INVALID PARAMETER VALUE FOR parameter**

Explanation:
Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask when an invalid parameter value was encountered.

User response:
Determine the problem and correct the parameter in error.

VFIRTS005I **mm/dd/yy hh:mm:ss – INVENTORY FILE OPENED, DDNAME = ddname**

Explanation:
Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask.

User response:
None. Informational message.

VFIRTS006I **mm/dd/yy hh:mm:ss – SCHEDULE EVENT ARRAY OBTAINED - NAME = seaname, ADDRESS = xxxxxxxx, SUB-POOL/LENGTH = yyzzzzzz**

Explanation

Informational message issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask, where:

- seaname - Schedule Event Array name
- xxxxxxxx - Schedule Event Array location
- yy - Storage subpool
- zzzzzz - Storage length

User response:

None. Informational message.

VFIRTS007I mm/dd/yy hh:mm:ss – MODULE LOADED - nnnnnnnn, ADDRESS = xxxxxxxx

Explanation

Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask, where:

- nnnnnnnn - name of load module being loaded
- xxxxxxx - load module area location

User response:
None. Informational message.

VFIRTS008E mm/dd/yy hh:mm:ss – UNABLE TO LOAD MODULE – *nnnnnnnn*

Explanation

Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask when an error occurred attempting to load module into storage, where:

- nnnnnnnn - name of load module being loaded

User response:
Contact Technical Support.

VFIRTS009W mm/dd/yy hh:mm:ss – INCLDSN NOT LOADED – *reason*

Explanation

Warning message issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask when attempting to load INCLDSN into storage, where *reason* is one of the following:

TABLE NOT LOCATED
UNSUCCESSFUL LOAD

User response:
None. Informational message.

VFIRTS010I mm/dd/yy hh:mm:ss – VFI/RTS TERMINATING

Explanation:
Issued by BRNVC008 during termination of Dynamic Dataset Scheduling subtask.

User response:
None. Informational message.

VFIRTS011I mm/dd/yy hh:mm:ss – VFI/RTS WAITING TO RELEASE SCHEDULE EVENT ARRAYS – CURRENTLY IN USE

Explanation:
Issued by BRNVC008 during termination of Dynamic Dataset Scheduling subtask.

User response:
None. Informational message.

VFIRTS012I mm/dd/yy hh:mm:ss – VFI/RTS SCHEDULE EVENT ARRAYS RELEASED

Explanation:
Issued by BRNVC008 during termination of Dynamic Dataset Scheduling subtask.

User response:
None. Informational message.

VFIRTS013I mm/dd/yy hh:mm:ss – VFI/RTS SCHEDULE JOB EVENTS PROCESSED DURING LAST SMF CYCLE

Explanation

Issued by BRNVC008 displaying counts for each scheduling event during the last SMF cycle. This is a multi-line message where the second line displays the counts:

mm/dd/yy hh:mm:ss – n,nnn,nnn EVENT1, n,nnn,nnn EVENT2, n,nnn,nnn CUTOFF, n,nnn,nnn CLOSE, n,nnn,nnn PENDING JOB

User response:
None. Informational message.

VFIRTS014I mm/dd/yy hh:mm:ss – VFI/RTS SCHEDULE TIME EVENTS PROCESSED DURING TIME INTERVAL ENDING AT *hh:mm*:

Explanation

Issued by BRNVC008 displaying counts for each scheduling event during the last time interval ending at *hh:mm*. This is a multi-line message where the second line displays the counts:

mm/dd/yy hh:mm:ss – n,nnn,nnn EVENT1, n,nnn,nnn EVENT2, n,nnn,nnn CUTOFF, n,nnn,nnn CLOSE, n,nnn,nnn

User response:
None. Informational message.

VFIRTS015I mm/dd/yy hh:mm:ss – VFI/RTS stype STARTUP COMPLETED

Explanation

Informational message issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask, where stype is one of the following:

- COLD - Cold start reads inventory component records and rebuilds the schedule for each record based on SCHDREQ criteria.

- WARM - Warm start reads inventory component records and resumes existing schedule from the record.

User response:
None.

VFIRTS016W mm/dd/yy hh:mm:ss – VFI/RTS RESOURCE SHORTAGE OF seaname - nn % UTILIZATION REACHED

Explanation
Warning message issued by BRNVC008 when threshold is reached for storage for Scheduling Event Array where seaname is one of the following arrays:

- IRKA - The amount of storage for this array should be incremented by - changing the value for SEAMAXDS parameter in SCHDPARM.
- SEA/J - The amount of storage for this array should be incremented by - changing the value for SEAJMAX parameter in SCHDPARM.
- SEA/T - The amount of storage for this array should be incremented by - changing the value for SEATMAX parameter in SCHDPARM.
- SEA/JIXO - The amount of storage for this array should be incremented by - changing the value for SEAMAXJB parameter in SCHDPARM.

User response:
None.

VFIRTS017E mm/dd/yy hh:mm:ss – VFI/RTS INVENTORY UPDATE FAILED; REC: Inventory record

Explanation:
Issued by BRNVC008 while attempting to update a record in the Inventory file with Dynamic Dataset Scheduling information.

User response:
None.

VFIRTS018E mm/dd/yy hh:mm:ss – VFI/RTS SEA INSERT FAILED; REC: Inventory record

Explanation
Multi-line message where the second line is:

mm/dd/yy hh:mm:ss – *reason seaname*

Issued by BRNVC008 while attempting to add an entry into the Schedule Event Array *seaname*, where *reason* is one of the following:

TABLE STORAGE EXHAUSTED FOR
NEXT FREE ADDR. INVALID FOR
INVALID HASH TIME FIELD FOR

and *seaname* is one of the following:

IRKA
SEA/J
SEA/T
SEA/JIXO

User response
If *reason* is TABLE STORAGE EXHAUSTED FOR, then increment values for arrays as follows:

- IRKA - The amount of storage for this array should be incremented by - changing the value for SEAMAXDS parameter in SCHDPARM.
- SEA/J - The amount of storage for this array should be incremented by - changing the value for SEAJMAX parameter in SCHDPARM.
- SEA/T - The amount of storage for this array should be incremented by - changing the value for SEATMAX parameter in SCHDPARM.
- SEA/JIXO - The amount of storage for this array should be incremented by - changing the value for SEAMAXJB parameter in SCHDPARM.

For other reasons, contact Technical Support.

VFIRTS019I mm/dd/yy hh:mm:ss – VFI/RTS SCHEDULE JOURNALING IS type

Explanation
Issued by BRNVC008 during startup of Dynamic Dataset Scheduling subtask, where type is one of the following:

- ENABLED - Journaling is enabled for Dynamic Dataset Scheduling
- DISABLED - Journaling is disabled for Dynamic Dataset Scheduling

User response:
None. Informational message.

VFISCMD01I mm/dd/yy hh:mm:ss – VFISCHED COMMAND PROCESSING SUB-COMMAND subcmd

Explanation:
Informational message issued by VFISCMD while processing a VFISCHED command where subcmd is the VFISCHED sub-command.

User response:
None. Informational message.

VFISCMD02E mm/dd/yy hh:mm:ss – VFISCHED COMMAND CONTAINS UN-RECOGNIZED SUB-COMMAND subcmd

Explanation:

Issued by VFISCMD while processing a VFISCHED command. An unrecognized sub-command was specified.

User response:
Check documentation on the syntax of VFISCHED command and correct the error.

VFISCMD03E mm/dd/yy hh:mm:ss – VFISCHED COMMAND IS MISSING REQUIRED SUB-COMMAND

Explanation:
Issued by BRNVSCMD while processing VFISCHED command. A required sub-command is missing.

User response:
Check documentation on the syntax of VFISCHED command and correct the error.

VFISCMD04E mm/dd/yy hh:mm:ss – INVALID SYNTAX FOR VFISCHED SUB-COMMAND subcmd

Explanation:
Issued by VFISCMD while processing VFISCHED sub-command subcmd. The syntax for sub-command subcmd is invalid.

User response:
Check documentation on the syntax of VFISCHED command and correct the error.

VFISCMD05E mm/dd/yy hh:mm:ss – INVALID VALUES FOR VFISCHED SUB-COMMAND subcmd

Explanation:
Issued by BRNVSCMD while processing VFISCHED sub-command subcmd. The VFISCHED sub-command subcmd contains invalid parameter values.

User response:
Check documentation on the syntax of VFISCHED command and correct the error.

VFISCMD06E mm/dd/yy hh:mm:ss – VFISCHED COMMAND FAILED, VFI/RTS IS NOT ACTIVE

Explanation:
Issued by VFISCMD while processing VFISCHED command. The Dynamic Dataset Scheduling subtask VFI/RTS is not active.

User response:
Verify that RTSMF started task is active and that the Dynamic Dataset Scheduling subtask VFI/RTS is active.

VFISCMD07I mm/dd/yy hh:mm:ss – VFISCHED REQUEST SUCCESSFUL

Explanation:
Issued by VFISCMD while processing a VFISCHED command. The VFISCHED request was successful.

User response:
None. Informational message.

VFISCMD08E mm/dd/yy hh:mm:ss – VFISCHED REQUEST FAILED

Explanation:
Issued by VFISCMD while processing VFISCHED command. The VFISCHED request failed with errors.

User response:
Check additional messages to see why the request failed.

VFISCMD09I mm/dd/yy hh:mm:ss – tablename REFRESH INDICATOR SET

Explanation:
Issued by VFISCMD while processing a VFISCHED REFRESH command. The REFRESH indicator for tablename was set and the table is refreshed.

User response:

VFISCMD10I mm/dd/yy hh:mm:ss – REBUILD INDICATOR SET FOR SCHEDULE

Explanation:
Issued by VFISCMD while processing a VFISCHED REBUILD command. The REBUILD indicator was set and the schedule is rebuilt.

User response:

VFISCMD11E mm/dd/yy hh:mm:ss – SCHDREQ – FAILED, NOT LOCATED

Explanation:
Issued by VFISCMD while processing VFISCHED REFRESH SCHDREQ command. The file containing the SCHDREQ parameters could not be located.

User response:
Verify that the SCHDREQ file exists and contains SCHDREQ parameters.

VFISCMD12W mm/dd/yy hh:mm:ss – SCHDREQ – BYPASSED, REFRESH IS ACTIVE

Explanation:
Warning message issued by VFISCMD while processing a VFISCHED REFRESH SCHREQ command. The REFRESH indicator is already set and the SCHDREQ table is refreshed.

User response:

VFISCMD13E mm/dd/yy hh:mm:ss – VFISCHED SUB-COMMAND subcmd FAILED, PROGRAM ENDED WITH RC=*rcode*

Explanation:

Issued by VFISCMD while processing VFISCHED subcommand subcmd. The program processing the subcommand failed with a return code *return_code*.

User response:
Contact Technical Support.

VFISDSP02E **mm/dd/yy hh:mm:ss – DISPLAY COULD NOT BE PROCESSED; VFI/RTS SCHEDULER TASK NOT ACTIVE**

Explanation:
Issued by VFISDSPL while processing VFISCHED DISPLAY command. The Dynamic Dataset Scheduling subtask VFI/RTS is not active.

User response:
Verify that RTSMF started task is active and that the Dynamic Dataset Scheduling subtask VFI/RTS is active.

VFISDSP03E **mm/dd/yy hh:mm:ss – DISPLAY COULD NOT BE PROCESSED; INVALID REQUEST**

Explanation:
Issued by VFISDSPL while processing VFISCHED DISPLAY command. The VFISCHED DISPLAY request is invalid.

User response:
Check documentation on the syntax of VFISCHED DISPLAY command and correct the error.

VFISDSP04E **mm/dd/yy hh:mm:ss – DISPLAY COULD NOT BE PROCESSED; SCHEDULE EVENT ARRAYS (SEA) NOT LOADED**

Explanation:
Issued by VFISDSPL while processing VFISCHED DISPLAY ARRAYS command. The VFI/RTS Schedule Event Arrays are not loaded.

User response:
Determine why the Schedule Event Arrays are not loaded.

VFISELI01I **mm/dd/yy hh:mm:ss - VFISELIG STARTING - OBTAINING DATASETS WHICH ARE ELIGIBLE FOR BACKUP**

Explanation:
Informational message issued by VFISELIG when obtaining data sets which are eligible for backup via Dynamic Dataset Scheduling.

User response:

VFISELI02I **mm/dd/yy hh:mm:ss - VFISELIG COMPLETED SUCCESSFULLY - OUTPUT WRITTEN TO DATASET dsname**

Explanation:
Informational message issued by VFISELIG when successfully obtained data sets which are eligible for backup via Dynamic Dataset Scheduling. Output is written to data set specified in the message.

User response:

VFISELI04E **mm/dd/yy hh:mm:ss - VFISELIG COMMAND IS MISSING REQUIRED PARAMETER(S)**

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing when a required parameter is missing.

User response:
Contact Technical Support.

VFISELI05E **mm/dd/yy hh:mm:ss - VFISELIG COMMAND CONTAINS UN-RECOGNIZED PARAMETERS – SEE 'SNAP' DATASET**

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing when an unrecognized parameter was found.

User response:
Contact Technical Support.

VFISELI06E **mm/dd/yy hh:mm:ss - VFISELIG COMMAND CONTAINS INVALID VALUE FOR PARAMETER(S): *parameter***

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing when an invalid *parameter* value was specified.

User response:
Contact Technical Support.

VFISELI07E **mm/dd/yy hh:mm:ss - VFISELIG FAILED, DYNALLOC ERROR/ INFO=*errc/info*, DSN=*dsname***

Explanation

Issued by VFISELIG during BKUPJCL mode=SCHED processing. VFISELIG failed with dynamic allocation error for data set dsname. The message displays the dynamic allocation error/information where:

- *errc* is the Dynamic Allocation error code
- *info* is the Dynamic Allocation Information code
- *dsname* is the Dataset name in error

User response:
Check IBM's MVS AUTH ASSEMBLER SERVICES GUIDE for description of Dynamic Allocation error and information codes.

VFISELI09E **mm/dd/yy hh:mm:ss - VFISELIG FAILED, *inventory error msg***

Explanation

Multi-line message where the second message is:

DSN/APPLID: dsname/applid

Issued by VFISELIG during BKUPJCL mode=SCHED processing. VFISELIG failed while trying to read the INVENTORY database to obtain data sets which are eligible for backup via Dynamic Dataset Scheduling. The message displays the error related to the Inventory database where *inventory_error_msg* is one of the following:

NO COMPONENT RECORDS IN FILE INVENTRY

FILE INVENTRY IS NOT OPENED

OPEN FAILED FOR FILE ddname, RC=cccc, ERROR=rrrrrrrr

GETKEY FAILED FOR FILE ddname, RC=cccc, ERROR=rrrrrrrr

GETNEXT FAILED FOR FILE ddname, RC=cccc, ERROR=rrrrrrrr

POINT FAILED FOR FILE ddname, RC=cccc, ERROR=rrrrrrrr

and dsname/applid is the data set name and application id of the record that failed.

User response:
Contact Technical Support.

VFISELI10E **mm/dd/yy hh:mm:ss - VFISELIG FAILED, FILE NOT ALLOCATED: DDNAME=*ddname***

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing. VFISELIG failed because file is not allocated for specified *ddname*.

User response:
Contact Technical Support.

VFISELI11E **mm/dd/yy hh:mm:ss - VFISELIG FAILED, UNABLE TO OPEN FILE: DDNAME=*ddname***

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing. VFISELIG failed because it was unable to open file for specified *ddname*.

User response:
Contact Technical Support.

VFISELI12E **mm/dd/yy hh:mm:ss - VFISELIG FAILED, GETMAIN ERROR FOR DCB AREA: DDNAME=*ddname***

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing. VFISELIG failed trying to acquire DCB storage for specified *ddname*.

User response:
Contact Technical Support.

VFISELI14E **mm/dd/yy hh:mm:ss - VFISELIG FAILED, UNABLE TO ACQUIRE ENQUEUE FOR FILE: DSNAME=*dsname***

Explanation:
Issued by VFISELIG during BKUPJCL mode=SCHED processing. VFISELIG failed because it could not acquire enqueue for output file. The message displays the *dsname* of the file.

User response:
Contact Technical Support.

VFISJRN02E **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL COULD NOT BE PROCESSED; VFI/RTS SCHEDULER TASK NOT ACTIVE**

Explanation:
Issued by VFISJRNC while processing VFISCHED JRNL command. The Dynamic Dataset Scheduling subtask VFI/RTS is not active.

User response:
Verify that RTSMF started task is active and that the Dynamic Dataset Scheduling subtask VFI/RTS is active.

VFISJRN03E **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL COULD NOT BE PROCESSED; INVALID REQUEST**

Explanation:
Issued by VFISJRNC while processing VFISCHED JRNL command. Invalid request was specified.

User response:
Check documentation on the syntax of VFISCHED JRNL command and correct the error.

VFISJRN04W **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL DISABLE COULD NOT BE PROCESSED; JOURNALIND ALREADY DISABLED**

Explanation:
Warning message issued by VFISJRNC while processing VFISCHED JRNL DISABLE command. Journaling is already disabled.

User response:

VFISJRN05W **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL CLOSE COULD NOT BE PROCESSED; JOURNALIND IS DISABLED**

Explanation:
Warning message issued by VFISJRNC while processing VFISCHED JRNL CLOSE command. Journaling is currently disabled.

User response:

VFISJRN06W **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL ENABLE COULD NOT BE PROCESSED; JOURNALIND ALREADY ENABLED**

Explanation:
Warning message issued by VFISJRNC while processing VFISCHED JRNL ENABLE command. Journaling is already enabled.

User response:

VFISJRN07W **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL CLOSE COULD NOT BE PROCESSED; JOURNAL SCHDJDS DATASET NOT OPEN**

Explanation:
Warning message issued by VFISJRNC while processing VFISCHED JRNL CLOSE command. Journaling SCHDJDS data set is not opened.

User response:

VFISJRN08E **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL ENABLE COULD NOT BE PROCESSED; UNABLE TO READ SCHDPARM**

Explanation:
Issued by VFISJRNC while processing VFISCHED JRNL ENABLE command. Unable to read SCHDPARM parameters.

User response:
Verify that SCHDPARM parameters are available.

VFISJRN09E **mm/dd/yy hh:mm:ss – SUB-COMMAND JRNL ENABLE COULD NOT BE PROCESSED; JOURNAL SCHDJDS DSN NOT ALLOCATED**

Explanation:
Issued by VFISJRNC while processing VFISCHED JRNL ENABLE command. SCHDJDS data set is not allocated.

User response:
Verify that SCHDJDS DDname is allocated.

VFISJRN11I **mm/dd/yy hh:mm:ss – VFI/RTS SCHEDULE JOURNALING IS DISABLED**

Explanation:
Informational message issued by VFISJRNC while processing VFISCHED JRNL DISABLE command. Journaling has been disabled.

User response:

VFISJRN12I **mm/dd/yy hh:mm:ss – JOURNAL SCHDJDS DATASET CLOSED**

Explanation:
Informational message issued by VFISJRNC while processing VFISCHED JRNL CLOSE command. Journaling SCHDJDS data set has been closed.

User response:

VFISJRN13I **mm/dd/yy hh:mm:ss – VFI/RTS SCHEDULE JOURNALING IS ENABLED: JOURNAL=journaldsn**

Explanation:
Informational message issued by VFISJRNC while processing VFISCHED JRNL ENABLE command. Journaling has been enabled.

User response:

VFISUBU01I **mm/dd/yy hh:mm:ss – VFISUBU STARTING SCHEDULE UPDATES WITH JOB NAME/NUMBER OF BACKUP JOB(S)**

Explanation:
Informational message issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. Dynamic Dataset Scheduling (VFI/RTS) updates Inventory database records with job name/number for all data sets being backed up.

User response:

VFISUBU02I **mm/dd/yy hh:mm:ss – VFISUBU SCHEDULE UPDATES COMPLETED – n,nnn,nnn INVENTORY RECORDS UPDATED; n,nnn,nnn SEA/J ENTRIES CREATED**

Explanation:
Informational message issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. Dynamic Dataset Scheduling completed schedule updates. Gives count of the number of INVENTRY records updated with backup job/number and the number of SEA/J entries created in the Schedule Event Array.

User response:

VFISUBU03E **mm/dd/yy hh:mm:ss – VFISUBU COMMAND CONTAINS UN-RECOGNIZED PARAMETERS – SEE SNAP DATASET**

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. The VFISUBU command that was dynamically generated by the backup job contains unrecognized parameters.

User response:
Contact Technical Support.

VFISUBU04E mm/dd/yy hh:mm:ss – VFISUBU COMMAND IS MISSING REQUIRED PARAMETERS

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. The VFISUBU command that was dynamically generated by the backup job is missing required parameters.

User response:
Contact Technical Support.

VFISUBU05E mm/dd/yy hh:mm:ss – VFISUBU FAILED, DYNALLOC ERROR/INFO = *errc/info*, DSN = *dsname*

Explanation

Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed with dynamic allocation error for data set dsname. The message displays the dynamic allocation error/information where:

- *errc* is the Dynamic Allocation error code
- *info* is the Dynamic Allocation Information code
- *dsname* is the Dataset name in error

User response:
Check IBM's MVS AUTH ASSEMBLER SERVICES GUIDE for description of Dynamic Allocation error and information codes.

VFISUBU06E mm/dd/yy hh:mm:ss – VFISUBU FAILED, FILE NOT ALLOCATED, DDNAME = ddname

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. Specified DDname is not allocated.

User response:
Verify that the specified DDname is allocated in the backup job JCL.

VFISUBU07E mm/dd/yy hh:mm:ss – VFISUBU FAILED, UNABLE TO OPEN FILE, DDNAME = ddname

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. Program was unable to open file with specified DDname.

User response:
Contact Technical Support.

VFISUBU08E mm/dd/yy hh:mm:ss – VFISUBU FAILED, VFI/RTS IS NOT ACTIVE

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. The Dynamic Dataset Scheduling subtask VFI/RTS is not active.

User response:
Verify that RTSMF started task is active and that the Dynamic Dataset Scheduling subtask VFI/RTS is active.

VFISUBU09E mm/dd/yy hh:mm:ss – VFISUBU FAILED, IN-STORAGE IRKA NOT INITIALIZED

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. The in-storage IRKA Scheduling Event Array is not initialized.

User response:
Contact Technical Support.

VFISUBU10E mm/dd/yy hh:mm:ss – VFISUBU FAILED, IN-STORAGE IRKA HAS NO ENTRIES

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. The in-storage IRKA Scheduling Event Array has no entries.

User response:
Contact Technical Support.

VFISUBU11E mm/dd/yy hh:mm:ss – VFISUBU FAILED, UNSUCCESSFUL INDEX BUILD FOR IRKA – REASON = 01-reas

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. Program was unsuccessful in building index for Scheduling Event Array IRKA.

User response:
Contact Technical Support.

VFISUBU12W mm/dd/yy hh:mm:ss – MATCHING INVENTORY RECORD NOT FOUND; REC:record

Explanation:
Warning message issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. No matching record was found in the Inventory database.

User response:

VFISUBU13W **mm/dd/yy hh:mm:ss – INVENTORY RECORD NOT UPDATED; REC: record**

Explanation:
Warning message issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. Matching record in Inventory database was not updated.

User response:

VFISUBU14W **mm/dd/yy hh:mm:ss – MATCHING IN-STORAGE SEA RECORD NOT FOUND; REC:record**

Explanation:
Warning message issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. No matching record was found in the Scheduling Event Array (SEA).

User response:

VFISUBU15W **mm/dd/yy hh:mm:ss – MATCHING IN-STORAGE SEA RECORD NOT RETRIEVED; REC:record**

Explanation:
Warning message issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted and a VFISUBU command is generated. Unable to retrieve the matching Scheduling Event Array (SEA) record.

User response:

VFISUBU16E **mm/dd/yy hh:mm:ss – VFISUBU FAILED, GETMAIN ERROR FOR DCB AREA, DDNAME = *ddname***

Explanation:
Issued by VFISUBU when an IBM Z Batch Resiliency backup job has been submitted. The VFISUBU command that was dynamically generated by the backup job failed. Program failed while issuing GETMAIN for DCB area for the specified *ddname*.

User response:
Contact Technical Support.

BRNVSAMI Messages

This section documents the BRNVSAMI messages.

VFI **BRNVSAMI CAN NOT OPEN ERROR MESSAGE FILE**

Explanation:
BRNVSAMI is unable to open the ERROR message file. Program ends.

User response:
Informational message: Console.

VFI **BRNVSAMI WAITING FOR BKUPTBL ENQUEUE**

Explanation:
BRNVSAMI is waiting to enqueue the BKUPTBL file resource. The inner loop waits five seconds and retry the enqueue up to twelve times. The outer loop waits twenty seconds, resend the message and retry the inner loop for up to four times.

User response:
Informational message: Console.

VFI **BRNVSAMI BKUPTBL ENQUE INUSE BY JOB (xxxxxxxx)**

Explanation:
BRNVSAMI sends this message to the operator console after repeated attempts to enqueue the BKUPTBL file resource. The message alerts the operator to the job using this resource.

User response:
Ensure that no other job is using the BKUPTBL file resource.

ERRVSAM01 **DD=<ddname> OPEN ERROR, PROGRAM ENDED.**

Explanation:
BRNVSAMI tried unsuccessfully to open the specified ddname. The program is ended.

User response:
Check the output listing DD=JESMSGLG for any IEC type messages related to the <ddname>. Follow the IBM guidance for the IEC type message. Rerun the job. If the problem persists, contact Technical Support.

ERRVSAM02 **CAN NOT GET CONTROL OF IBM Z Batch Resiliency ENQUEUE**

Explanation:
BRNVSAMI tried unsuccessfully to enqueue the BKUPTBL file. The program is ended.

User response:
Ensure that no other program is using this resource. If the problem persists, contact Technical Support.

ERRVSAM03 **CAN NOT READ JFCB FOR <ddname> FILE!**

Explanation:
An error occurred when BRNVSAMI tried to read the <ddname> file JFCB. The program is ended.

User response:
Contact Technical Support.

ERRVSAM04 **CAN NOT OPEN BKUPAUD FILE WITH NEW JFCB!**

Explanation:
An error occurred when BRNVSAMI tried to open the BKUPAUD file with DISP=old.

User response:
Ensure that no other program is using this file exclusively. If the problem persists, contact Technical Support.

ERRVSAM05 **<macro name> MACRO ERROR IN <routine>**

Explanation:
An error occurred when BRNVSAMI tried to execute the named macro in <routine>. The program ends.

User response:
Contact Technical Support.

ERRVSAM06 **TESTCB MACRO FOUND WRONG INFORMATION FOR: < keyname variable (CL70) > RETURN CODE = '##'**

Explanation:
An error occurred after BRNVSAMI was unable to point to a generic key in the BKUPTBL.

User response:
Contact Technical Support.

ERRVSAM07 **CAN NOT INSERT THE FOLLOWING <rectype> RECORD (BKPFLAG X"##") < keyname variable (CL70) >**

Explanation:
BRNVSAMI was unable to insert a record into the BKUPTBL.

User response:
Ensure that the BKUPTBL file is not out of space. Rerun the job. If the problem persists contact Technical Support.

ERRVSAM08 **CAN NOT UPDATE THE FOLLOWING <rectype> RECORD (BKPLAG X"##") < keyname variable (CL70) >**

Explanation:
An error occurred when BRNVSAMI tried to update the BKUPTBL file record.

User response:
Contact Technical Support.

ERRVSAM09 **INVALID BKPVCNT, NOT NUMERIC < keyname variable (CL70) >**

Explanation:
BRNVSAMI has found a non-numeric value for BKUPTBL record volume count. Processing continues.

User response:
Contact Technical Support.

ERRVSAM10 **RETURN CODE = ##**

Explanation:
This message specifies the return code which accompanies a prior error message.

User response:
Contact Technical Support.

ERRVSAM11 **INVALID BKPVCNT, MUST BE FROM 1 THRU 255 < keyname variable (CL70) >**

Explanation:
BRNVSAMI has found an invalid numeric value for BKUPTBL record volume count. Processing continues.

User response:
Contact Technical Support.

ERRVSAM13 **INVALID BKPFLAG, MUST =X"00" < keyname variable (CL70) >**

Explanation:
BRNVSAMI has found an invalid value for BKUPTBL record BKPFLAG. Processing continues.

User response:
Contact Technical Support.

ERRVSAM14 **INVALID BKPFSEQ, NOT NUMERIC < keyname variable (CL70) >**

Explanation:
BRNVSAMI has found a non-numeric value for BKUPTBL record file sequence number. Processing continues.

User response:
Contact Technical Support.

ERRVSAM15 **INVALID BKPFSEQ, MUST BE FROM 1 THRU 255' < keyname variable (CL70) >**

Explanation:
BRNVSAMI has found an invalid numeric value for BKUPTBL record file sequence number. Processing continues.

User response:

Contact Technical Support.

ERRVSAM16 **INVALID INPUT RECORD, REQUIRED <variable name> FIELD BLANK < keyname variable (CL70) >**

Explanation:
BRNVSAMI has found an invalid blank field in the BKUPTBL. Processing continues.

User response:
Contact Technical Support.

ERRVSAM17 **ATTENTION MESSAGES FOLLOW, ALSO SEE DD=JESMSGLG**

Explanation:
BRNVSAMI lists attention messages following a failed BKUPTBL open.

User response:
Contact Technical Support.

ERRVSAM18 **ACB CODE FUNCTYP DATASET NAME**

Explanation:
BRNVSAMI attention message list header line.

User response:
Contact Technical Support.

ERRVSAM19 **<acb code> <function> <dataset name>**

Explanation:
BRNVSAMI attention message list detail line.

User response:
Contact Technical Support.

ERRVSAM20 **VFI GETMAIN ERROR!**

Explanation:
An error occurred when BRNVSAMI tried to obtain main storage space for queue processing.

User response:
Ensure that the region parameter on the job card is at least 1M. Fix this and rerun the job. If the problem persists, contact Technical Support.

ERRVSAM23 **<ddname variable> GQSCAN MACRO ERROR. REASON CODE ####.**

Explanation:
An error occurred when BRNVSAMI tried to process a queue inquiry.

User response:
Contact Technical Support.

ERRVSAM51 **<keyname variable (CL70) >**

Explanation:
BRNVSAMI specifies the BKUPTBL record key in error.

User response:
Contact Technical Support.

ERRVSAM52 **REASON CODE X"nnnn"**

Explanation:
BRNVSAMI specifies the reason code for the associated error.

User response:
Contact Technical Support.

ERRVSAM99 **PROGRAM END**

Explanation:
BRNVSAMI reports that program processing ends because of the last listed error message.

User response:
Contact Technical Support.

BRNVSAMU Messages

This section documents the BRNVSAMU messages.

ERRVSAM00 **CAN NOT OPEN BKUPTBL FILE.**

Explanation:
BRNVSAMU tried unsuccessfully to enqueue the BKUPTBL file. The program waited and retried the enqueue 5 times in 20 second intervals.

User response:
Ensure that no other program is using the resource. If the problem persists, contact Technical Support.

ERRVSAM01 **CAN NOT OPEN BKUPTBL FILE. RETRY in 20 SEC**

Explanation:
BRNVSAMU tried unsuccessfully to enqueue the BKUPTBL file. The program waits and retries the enqueue in 20 seconds.

User response:

ERRVSAM02 **CAN NOT HAVE CONTROL ON IBM Z Batch Resiliency BKUPAUD FILE!**

Explanation:
BRNVSAMU tried unsuccessfully to enqueue the BKUPAUD file.

User response:
Ensure that no other program is using this resource. If the problem persists, contact Technical Support.

ERRVSAM03 **CAN NOT READ JFCB FOR BKUPAUD FILE!**

Explanation:
An error occurred when BRNVSAMU tried to read the BKUPAUD file JFCB.

User response:
Contact Technical Support.

ERRVSAM04 **CAN NOT OPEN BKUPAUD FILE WITH NEW JFCB!**

Explanation:
An error occurred when BRNVSAMU tried to open the BKUPAUD file with DISP=OLD.

User response:
Ensure that no other program is using this file exclusively. If the problem persists, contact Technical Support.

ERRVSAM06 **TESTCB MACRO FOUND WRONG INFORMATION FOR:> keyname variable (CL70) <RETURN CODE = 'rc'**

Explanation:
An error occurred after BRNVSAMU was unable to point to a generic key in the BKUPTBL.

User response:
Contact Technical Support.

ERRVSAM07 **DSNKEY MATCH NOT FOUND IN BKUPTBL:> keyname variable (CL70) <**

Explanation:
BRNVSAMU only updates records found in the BKUPTBL. It does not insert new records. This message identifies a full record key (cl70) not found in the BKUPTBL.

User response:

ERRVSAM08 **CAN NOT UPDATE THE FOLLOWING RECORD:> keyname variable (CL70) <RETURN CODE = 'rc'**

Explanation:
An error occurred when BRNVSAMU tried to update the BKUPTBL file record.

User response:
Contact Technical Support.

ICP Messages

This section documents the ICP messages.

CKKRC03601I **mm/dd/yy hh:mm:ss CKKRC036: ICP REQID(xxxxxxxx)**

Explanation:
This message indicates a successful update of the internal ICP table. The interval control element request id is documented.

User response:
None.

CKKRC03602E **mm/dd/yy hh:mm:ss - CKKRC036: ICP RC=x**

Explanation

This message indicates a failed ICP request. The RC value indicates the error:

1-Interval specification error

2-Timer specification error

3-Task specification error

Additionally, the ICP command in error is listed.

User response:
None.

CKKRC03603I **mm/dd/yy hh:mm:ss - CKKRC036: ICP RTDICPIN PROCESSED**

Explanation:
This message is issued at the end of the RTDICPIN member.

User response:
None.

CKKRC03604E **mm/dd/yy hh:mm:ss - CKKRC036: PPT ERROR TASK(xxxxxxx) PROCESS BYPASSED**

Explanation:
This message is issued when the there is a TASK/PPT mismatch.

User response:
Contact Technical Support.

CKKRC03605E **mm/dd/yy hh:mm:ss - CKKRC036: PPT LOAD ERROR:xxxxxxxx**

Explanation:
This message is issued when there is a load failure for the identified Process Program Table entry.

User response:
Contact Technical Support.

CKKRC04001E mm/dd/yy hh:mm:ss - CKKRC040: COMMAND LENGTH ERROR

Explanation:
This message indicates an error in the command length.

User response:
Contact Technical Support.

CKKRC04002E mm/dd/yy hh:mm:ss - CKKRC040: DDN:RTACMDQ NOT AVAILABLE

Explanation:
This message indicates the RTACMDQ DD is not in the JCL.

User response:
Correct the JCL and rerun.

CKKRC04003E mm/dd/yy hh:mm:ss - CKKRC040: RTACMDQ ENQ FAILED, COMMAND BYPASSED

Explanation:
This message indicates the command queue file is not available.

User response:
Resolve the enqueue issue, then retry.

CKKRC04004E mm/dd/yy hh:mm:ss - CKKRC040: RTACMDQ OPEN ERROR, COMMAND BYPASSED

Explanation:
This message indicates an error opening the command queue data set.

User response:
Correct the error, then retry.

CKKRC04005E mm/dd/yy hh:mm:ss CKKRC040: RTACMDQ GET ERROR, COMMAND BYPASSED

Explanation:
This message indicates a GET error on the command processor data set.

User response:
Correct the error, then retry.

CKKRC04006E mm/dd/yy hh:mm:ss CKKRC040: RTACMDQ PUT ERROR, COMMAND BYPASSED

Explanation:
This message indicates a PUT error on the command processor data set.

User response:
Correct the error, then retry.

CKKRC04007I mm/dd/yy hh:mm:ss - CKKRC040: RTACMDQ

Explanation:
This message indicates a successful command request.

User response:
None.

CKKRC04007I mm/dd/yy hh:mm:ss -CKKRC040: RTACMDQ RANGE ERROR(xxxxxxxx)

Explanation:
This message indicates an RRN error.

User response:
Contact Technical Support.

CKKRC04101I mm/dd/yy hh:mm:ss -CKKRC041: ICP DISPLAY REQUESTED

Explanation:
This message indicates an ICP display request.

User response:
None.

CKKRC04102I mm/dd/yy hh:mm:ss - CKKRC041: ICP DISPLAY RE-SCHEDULED

Explanation:
This message indicates an automatic ICP display scheduled.

User response:
None.

TimeLiner Messages

This section documents the TimeLiner messages.

BRNL001E XXXX FAILED, MACRO 'XXXX' NEAR LABEL 'XXXX' WITH RETURN CODE = XXXX REASON CODE = XXXX

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL002E XXXX FAILED, MACRO 'XXXX' NEAR LABEL 'XXXX' WITH ERROR NUMBER = XXXX

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL003I XXXX CALLED 'XXXX' NEAR LABEL 'XXXX' TO OBTAIN SOCKET DESCRIPTOR = XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL004I XXXX CALLED 'XXXX' NEAR LABEL 'XXXX' TO OBTAIN HOSTNAME = XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL005I XXXX CALLED 'XXXX' NEAR LABEL 'XXXX' TO OBTAIN XXXX TCPIP ADDRESS = XXXX.XXXX.XXXX.XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL006I XXXX XXXX = XXXX

Explanation:
This message is generic and is used for displays of variables and the like. Message text displayed varies, according to the data being shown.

User response:
None. Informational message.

BRNL007I XXXX ISSUED, MACRO 'XXXX' NEAR LABEL 'XXXX' FOR XXXX CHARACTER MESSAGE = XXXX

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL009E XXXX FAILED, MACRO 'XXXX' NEAR LABEL 'XXXX' WITH RETURN CODE = XXXX

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL013I XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH COMPLETION CODE XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL014S XXXX INVALID DATA SPACE FOUND NEAR LABEL XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL015E XXXX DATASET 'XXXX' DDNAME 'XXXX' FAILED TO OPEN NEAR LABEL 'XXXX'

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL016I XXXX SUBMITTED JOB 'XXXX' WITH JES IDENTIFICATION 'XXXX'

Explanation:
The stated job has been submitted. JES identification is displayed.

User response:
None. Informational message.

BRNL019E XXXX NEAR LABEL 'XXXX' INVALID VALUE FOUND XXXX = XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL020E **XXXX NEAR LABEL 'XXXX' FAILED TO FIND ACTIVE SMF IEFU84 EXIT**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL022I **XXXX READ XXXX XXXX RECORDS FROM DATASET 'XXXX'**

Explanation:
Displays the records read from the stated dataset.

User response:
None. Informational message.

BRNL024W **XXXX FAILED, MACRO XXXX NEAR LABEL XXXX WITH CODE = XXXX SENSE =**

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL026S **XXXX FAILED NEAR LABEL XXXX WITH INVALID TSO COMMAND ENVIRONMENT**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL028W **XXXX MACRO XXXX ENCOUNTERED AN INVALID DATASET NAME 'XXXX' NEAR LABEL XXXX**

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL029W **XXXX FAILED, MACRO XXXX NEAR LABEL XXXX WITH REG.15 = XXXX RECORD KEY = XXXX**

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL031I **XXXX SMS XXXX CONSTRUCT NAME XXXX WITH DESCRIPTION XXXX LAST UPDATED BY XXXX ON XXXX AT XXXX**

Explanation:
The date on which the SMS construct name with the stated description was last updated by the user displayed.

User response:
None. Informational message.

BRNL032I **XXXX FOUND DSNAME 'XXXX' CATALOGUED IN CATALOG 'XXXX' WITH VOLUME SEQUENCE XXXX ON XXXX WITH DEVICE TYPE XXXX**

Explanation:
The DSNAME stated was found to be catalogued in the catalog listed, with the volume sequence displayed and by the device shown.

User response:
None. Informational message.

BRNL039E **XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH RETURN CODE = XXXX, REASON CODE = XXXX ABEND CODE = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL040E **XXXX XXXX RECORD COULD NOT BE PLACED IN DATASPACE BUFFER BECAUSE THE DATASPACE WAS FULL AT THE TIME**

Explanation:
The stated record could not be placed in dataspace buffer because the dataspace was full.

User response:
Contact Technical Support with the details of the message.

BRNL045E **XXXX call to 'XXXX' near label 'XXXX' failed with return_value = XXXX return_code = XXXX reason_code = XXXX**

Explanation:

An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL048E XXXX PROBABLE CAUSE, CURRENT TASK ALREADY RUNNING ON THIS SYSTEM

Explanation:
TimeLiner is already running on this system. Only one instance of TimeLiner can be running at one time.

User response:
Cancel the second instance of TimeLiner.

BRNL049I XXXX XXXX SHUTDOWN DUE TO XXXX SHUTDOWN

Explanation:
An instance of TimeLiner has been shut down, because another was already active.

User response:
None. Informational message.

BRNL052E XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH RETURN CODE = XXXX, REASON CODE = XXXX MODULE ID = XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL053I XXXX COMMAND 'XXXX' COMPLETED WITH COMPLETION CODE XXXX

Explanation:
The issued command has been completed.

User response:
None. Informational message.

BRNL055E XXXX UNABLE TO FIND MEMBER XXXX IN DDNAME XXXX

Explanation:
The stated member cannot be found.

User response:
Correct the member name and retry.

BRNL056I XXXX COMMAND 'XXXX' ACCEPTED FROM DDNAME XXXX MEMBER XXXX FOUND IN XXXX

Explanation:
The stated command was successfully accepted.

User response:
None. Informational message.

BRNL057I XXXX SELECTED XXXX RECORDS WITH JULIAN DATES PRIOR TO OR EQUAL TO XXXX AND RECORDS WITH JULIAN DATES LATER THAN XXXX IS DELETED

Explanation:
The selected records which meet the date criteria stated is deleted.

User response:
None. Informational message.

BRNL058I XXXX JOB NUMBER XXXX SUBMITTED TO INTERNAL READER FROM DDNAME XXXX MEMBER XXXX FOUND IN XXXX

Explanation:
The stated job number which was submitted by DDNAME, was found in the stated data set.

User response:
None. Informational message.

BRNL059I XXXX NO VALID JOB SUBMITTED TO INTERNAL READER FROM DDNAMEXXXX MEMBER XXXX FOUND IN XXXX

Explanation:
The job submitted to the internal reader by the stated DDNAME was not found.

User response:
None. Informational message.

BRNL060I XXXX JOB XXXX ON SYSTEM XXXX CURRENTLY XXXX XXXX ENQUEUE OF QNAME 'XXXX' RNAME 'XXXX'

Explanation:
The status of the stated job on the system enqueue with the QNAME and RNAME listed.

User response:
None. Informational message.

BRNL061E XXXX Probable cause, LOADLIB is not APF authorized

Explanation:
TimeLiner is not authorized on this system.

User response:
Contact your System Programmer to reconfigure the system installation.

BRNL063E XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH SEVERITY CODE = XXXX, ERROR CODE = XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL067E **XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH RETURN CODE = XXXX, REASON CODE = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL068E **XXXX FAILED, MACRO 'XXXX' NEAR LABEL 'XXXX' WITH RETURN CODE = XXXX FOR DATASET 'XXXX' ON VOLUME XXXX**

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL069E **XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH COMPLETION CODE XXXX FOR DATASET 'XXXX' ON VOLUME XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL073I **XXXX CALLED 'XXXX' NEAR LABEL 'XXXX' TO OBTAIN XXXX TCPIP ADDRESS = XXXX.XXXX.XXXX.XXXX PORT NUMBER = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL076E **XXXX XXXX ERROR FEEDBACK NEAR LABEL XXXX ABEND CODE = XXXX ABEND REASON CODE = XXXX SECURITY CODE = XXXX SECURITY REASON CODE = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL078E **INVALID OR MISSING 'XXXX' KEYWORD OPERAND**

Explanation:
The stated keyword operand is invalid.

User response:
Correct operand and retry.

BRNL083E **Unable to locate license key input, product or feature use denied, please contact Technical Support to obtain license for further use**

Explanation:
A license required for access to the product or feature being used is unable to be found.

User response:
Contact your System Programmer for details on the available licenses.

BRNL084I **XXXX INPUT ACCEPTED FROM DDNAME XXXX MEMBER XXXX FOUND IN XXXX**

Explanation:
The stated input was successfully accepted by the DDNAME member and found in the displayed data set.

User response:
None. Informational message.

BRNL085E **XXXX CATALOG ENTRY FOR DSN = XXXX DOES NOT MATCH DATABASE ENTRY**

Explanation:
The stated catalog entry for the DSN does not match the database entry.

User response:
Take note of the message content. Contact Technical Support with contents of the message.

BRNL087E **XXXX CATALOG ENTRY FOR DSN = XXXX NOT FOUND**

Explanation:
The catalog entry for the stated DSN was not found.

User response:
Correct and retry.

BRNL088E **XXXX RECATALOG FOR DSN = XXXX FAILED**

Explanation:
The recatalog for the stated DSN failed.

User response:
Retry operation. If problem reoccurs, contact your System Programmer.

BRNL089E **XXXX XXXX INVALID LENGTH RECORD COULD NOT BE PLACED IN DATASPACE BUFFER**

Explanation:
The record for the dataspace buffer is of invalid length.

User response:
Rename record and retry.

BRNL090E **XXXX XXXX EXCESSIVE LENGTH RECORD COULD NOT BE PLACED IN DATASPACE BUFFER**

Explanation:
The record to be placed in the dataspace buffer has exceeded the valid length.

User response:
Correct record and retry.

BRNL111E **XXXX XXXX RECORD COULD NOT BE PLACED IN DATASPACE BUFFER BECAUSE DATASPACE WAS INVALID AT THAT TIME**

Explanation:
The selected record could not be placed in the dataspace buffer, as the dataspace was invalid at the time of attempt.

User response:
Retry. If the error reoccurs, contact your System Programmer.

BRNL112E **XXXX XXXX RECORD COULD NOT BE PLACED IN DATASPACE BUFFER BECAUSE VECTOR TABLE WAS INVALID AT THAT TIME**

Explanation:
The selected record could not be placed in the dataspace buffer, as the the vector table was invalid at the time of attempt.

User response:
Retry. If the problem reoccurs, contact your System Programmer.

BRNL113E **XXXX XXXX INVALID LENGTH SYSPLEX MESSAGE PLACED IN DATASPACE BUFFER**

Explanation:
The sysplex message is of an invalid length for successful placement in the dataspace buffer.

User response:
Correct the sysplex message and retry.

BRNL114E **XXXX XXXX INVALID DATA FOUND IN EMPTY DATASPACE BUFFER**

Explanation:
Invalid data has been found in the empty dataspace buffer.

User response:
Contact Technical Support with the contents of the message.

BRNL116I **XXXX ACCEPTED INPUT FROM 'XXXX' NEAR LABEL 'XXXX' FOR XXXX CHARACTER MESSAGE = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL117E **XXXX PROBABLE CAUSE, ENTERPRISE SCHEDULER ADDRESS SPACE IS NOT ACTIVE ON THIS SYSTEM**

Explanation:
An error has occurred, with the most likely explanation that the enterprise scheduler space is not active on the current system.

User response:
Contact your System Programmer with the details of the message.

BRNL118E **XXXX XXXX ON THIS SYSTEM, THE JOB ACTIVITY DATABASE IS NOT BEING UPDATED AND JOB SCHEDULING CLIENTS ARE NOT BEING NOTIFIED**

Explanation:
An error has occurred which is preventing the job activity database from being updated and job scheduling clients from being notificed.

User response:
Contact Technical Support.

BRNL123W **INSUFFICIENT STORAGE AVAILABLE FOR MODULE XXXX TO EXECUTE AT OPTIMUM EFFICIENCY. MINIMUM OF XXXX BYTES OF STORAGE NEEDED**

Explanation:

The module cannot perform at optimal efficiency because not enough storage is available. The minimum amount of storage required is listed in the message.

User response:
Contact your System Programmer with the contents of the message.

BRNL125E **XXXX PROBABLE CAUSE, INSUFFICIENT REGION SIZE**

Explanation:
An error has occurred, most likely due to insufficient region size.

User response:
Contact your System Programmer.

BRNL126E **XXXX FAILED, MACRO 'XXXX' NEAR LABEL 'XXXX' WITH RETURN CODE = XXXX & REASON CODE = XXXX ON VOLUME XXXX**

Explanation:
Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL127W **XXXX UNABLE TO MOUNT FILESYSTEM DATASET NAME 'XXXX' ON VOLUME 'XXXX'**

Explanation:
The filesystem dataset stated cannot be mounted on the displayed volume.

User response:
Contact Technical Support.

BRNL128E **XXXX UNABLE TO OBTAIN THE VVDS ENTRY FOR DATASET NAME 'XXXX' ON VOLUME 'XXXX' LOCATED IN CATALOG 'XXXX'**

Explanation:
The VVDS entry for the listed dataset name on the volume stated, located in the catalog, cannot be obtained.

User response:
Contact your System Programmer.

BRNL155I **XXXX XXXX BLDVRP BUFFERS=(XXXX(XXXX)),KEYLEN = XXXX,STRNO = XXXX,TYPE = GSR**

Explanation:

User response:

BRNL156I **XXXX XXXX BLDVRP BUFFERS=(XXXX(XXXX,XXXX)),KEYLEN = XXXX,SHRPOOL = XXXX,STRNO = XXXX,TYPE = (LSR,XXXX)**

Explanation:

User response:

BRNL157I **XXXX UNABLE TO LOOKUP STORAGE GROUP - PROCEEDING ON**

Explanation:
The storage group lookup could not be executed.

User response:
None. Informational message.

BRNL158E **XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' FAILED TO FIND DDNAME 'XXXX'**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of message content. Contact Technical Support with contents of message.

BRNL159E **XXXX DDNAME 'XXXX' FAILED TO OPEN NEAR LABEL 'XXXX'**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of message content. Contact Technical Support with contents of message.

BRNL160E **XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH RETURN CODE = XXXX, REASON CODE = XXXX, PROBLEM DETERMINATION CODES = XXXX,XXXX, FOR DSNAME = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of message content. Contact Technical Support with contents of message.

BRNL161E **XXXX FAILED, MACRO 'XXXX' NEAR LABEL 'XXXX' WITH RETURN CODE = XXXX & REASON CODE = XXXX FOR DSNAME = XXXX**

Explanation:

Stated macro has failed. Message is only displayed in debugging mode.

User response:
Take note of message content. Contact Technical Support with contents of message.

BRNL162I 'XXXX' ACCEPTED FROM DDNAME XXXX MEMBER XXXX FOUND IN XXXX

Explanation:
The stated element has been accepted from the listed DDNAME and member found in the module displayed.

User response:
None. Informational message.

BRNL164E XXXX XXXX XXXX (XXXX) XXXX FAILED IN XXXX, RETURN CODE XXXX, REASON CODE XXXX - XXXX

Explanation:
The stated task has failed in the module listed, with the return code and reason code displayed.

User response:
Contact Technical Support.

BRNL165I XXXX XXXX XXXX (XXXX) XXXX EXECUTED, COMPLETION CODE XXXX, RETURN CODE XXXX

Explanation:
The stated task has been executed, with the completion code and return code displayed.

User response:
None. Informational message.

BRNL166I XXXX XXXX XXXX (XXXX) XXXX JOBNAME XXXX SCHEDULED FOR XXXX

Explanation:
The jobname displayed is scheduled for the stated time.

User response:
None. Informational message.

BRNL167I XXXX XXXX XXXX (XXXX) XXXX JOBNAME XXXX NOT SCHEDULED WITHIN XXXX DAYS

Explanation:
The stated jobname was not scheduled within the listed number of days.

User response:
None. Informational message. Reschedule jobname if required.

BRNL169E XXXX INVALID COMMAND - XXXX

Explanation:
An invalid command has been sent.

User response:
Correct the command and retry.

BRNL170E XXXX MISSING COMMAND

Explanation:
An error has occurred. The stated module is missing a command.

User response:
Contact Technical Support.

BRNL171E XXXX INVALID OPERAND (XXXX) FOR XXXX COMMAND

Explanation:
The operand entered is invalid.

User response:
Correct the operand and retry.

BRNL172E XXXX MISSING OPERAND FOR XXXX COMMAND

Explanation:
A required operand for the entered command is missing.

User response:
Add the missing operand and retry.

BRNL181I XXXX TLNASCHD STARTED XXXX (ACTIVE XXXX)

Explanation:
States when the TLNASCHD task was started and for how long it has been active.

User response:
None. Informational message.

BRNL182I XXXX SHUTDOWN IN PROGRESS

Explanation:
Shutdown of the stated task is in progress.

User response:
None. Informational message.

BRNL186I XXXX XXXX XXXX (XXXX) XXXX SET FOR IMMEDIATE EXECUTION

Explanation:
The stated task is scheduled for immediate execution.

User response:
None. Informational message.

BRNL192E XXXX XXXX statement not accepted: XXXX - XXXX XXXX

Explanation:
The listed statement was not accepted for the stated reasons.

User response:
Contact Technical Support.

BRNL193E XXXX Non-COLLECT statement encountered

Explanation:
As stated, a non-collect statement has been encountered.

User response:
Contact Technical Support.

BRNL194E XXXX Device XXXX, XXXX failed, return code XXXX

Explanation:
The stated device failed with the return code displayed.

User response:
Contact your System Programmer with the contents of the message.

BRNL195E XXXX Device XXXX, XXXX failed, return code XXXX status XXXX

Explanation:
States the failed device with the return code shown along with the current status.

User response:
Contact your System Programmer with the contents of the message.

BRNL196I XXXX Device XXXX online, not ready status

Explanation:
The listed device is online, but currently displaying a status of Not Ready.

User response:
Wait for the device to be ready and retry.

BRNL197E XXXX Device XXXX, Volume XXXX reserve time-out, device stats not available

Explanation:
The reserve for the volume on the stated device has timed-out. The statistics for the device are unavailable.

User response:
Contact your System Programmer with the contents of the message.

BRNL198E XXXX Device XXXX, Volume XXXX XXXX failed, return code XXXX

Explanation:
The stated volume on the device shown has failed, with the return code as displayed.

User response:
Contact your System Programmer with the contents of the message.

BRNL200I XXXX TLN I/O error monitoring suppressed

Explanation:
I/O error monitoring for TimeLiner has been suppressed.

User response:
None. Informational message.

BRNL201E XXXX TLN I/O error monitor installation incomplete

Explanation:
Installation of TimeLiner I/O error monitoring was not completed.

User response:

BRNL203I XXXX IFS LOADED FROM XXXX:

Explanation:
States the data set from which IF statements have been loaded.

User response:
None. Informational message.

BRNL206I XXXX NO IF STATEMENTS LOADED

Explanation:
No IF statements have been loaded and could be required.

User response:
None. Informational message.

BRNL209E XXXX TLN I/O error monitor in use, cannot be freed

Explanation:
An I/O error has occurred and TimeLiner cannot be freed to execute.

User response:
Contact your System Programmer with the contents of the message.

BRNL210E XXXX TLN Task ID XXXX already active

Explanation:
A task which the user is attempting to restart is already active.

User response:
Close one task to allow the other instance to run.

BRNL211E XXXX requires an active TLN task which has completed volume data collection

Explanation:
The stated task requires an active TimeLiner task with volume data collection complete.

User response:

Contact Technical Support.

BRNL212E XXXX Automation requires a SYSEXEC data set

Explanation:
Automation of the stated task requires a SYSEXEC data set .

User response:
Contact your System Programmer with the details of the message.

BRNL214E XXXX Data refresh not initiated

Explanation:
Data refresh on the stated task has not been initiated.

User response:
Contact Technical Support.

BRNL215E XXXX Unable to refresh data: XXXX not active

Explanation:
Data cannot be refreshed because the stated task is not active.

User response:
Activate the stated task and retry.

BRNL216E XXXX requires APF authorization

Explanation:
The stated task requires APF authorization to run.

User response:
Contact your System Programmer for authorization.

BRNL218I XXXX Data collectors is detached

Explanation:
The stated data collectors is detached.

User response:
None. Informational message.

BRNL219I XXXX Expect 13E abends of TLNADCSB

Explanation:

User response:
None. Informational message.

BRNL224E XXXX NO INCLUDED VOLUMES FOUND

Explanation:
No included volumes were found for the stated task.

User response:
Contact Technical Support.

BRNL225I XXXX XXXX COMMAND RECEIVED FROM XXXX

Explanation:
The stated command has been received from the module or feature listed.

User response:
None. Informational message.

BRNL228E XXXX DATASET 'XXXX' UNAVAILABLE, CURRENTLY BEING UPDATED BY ANOTHER TASK

Explanation:
The required, listed data set cannot be accessed, as it is currently being updated by another task.

User response:
Wait for the other task to complete and then retry.

BRNL229E XXXX NEAR LABEL XXXX UNABLE TO FIND ACTIVE TLN ADDRESS SPACE CVT ADDRESS

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL230E XXXX NEAR LABEL 'XXXX' INVALID XXXX FOUND FOR VOLUME XXXX

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL231E XXXX NEAR LABEL XXXX INVALID TLN ADDRESS SPACE CVT TOKEN CHECKSUM

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL232E XXXX NEAR LABEL XXXX INVALID TLN ADDRESS SPACE CVT VERSION

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL233E **XXXX NEAR LABEL XXXX TLN ADDRESS SPACE CVT ADDRESS HAS CHANGED**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL234I **XXXX CALLED FROM XXXX AT OFFSET XXXX**

Explanation:

User response:
None. Informational message.

BRNL235W **XXXX DSNAME 'XXXX' ON VOLUME XXXX SPACE UTILIZATION CALCULATION INVALID, XXXX**

Explanation:
The stated DSNAME on the volume listed has an invalid space utilization calculation.

User response:
Contact your System Programmer with the contents of the message.

BRNL236W **XXXX CERTAIN SPACE UTILIZATION CALCULATIONS CANNOT BE PERFORMED ON XXXX DATASETS RESIDING ON VOLUME XXXX IN A COPYPOOL SMS STORAGE GROUP**

Explanation:

User response:
Contact your System Programmer.

BRNL237W **XXXX Data set 'XXXX' is cataloged on a volume other than XXXX**

Explanation:
The stated data set is cataloged on a volume that is unexpected.

User response:
Contact your System Programmer with the details of the message.

BRNL238E **XXXX Probable cause, syntax error in a TLNPRMxx member of the PARMLIB dataset**

Explanation:
There is a syntax error in the PARMLIB data set.

User response:
Correct the syntax error and retry.

BRNL240E **XXXX CALL TO 'XXXX' NEAR LABEL 'XXXX' COMPLETED WITH RETURN CODE = XXXX, ERROR CODE = XXXX, ABEND CODE = XXXX & REASON CODE = XXXX**

Explanation:
An error has occurred. Message is only displayed in debugging mode.

User response:
Take note of the message details. Contact Technical Support with contents of the message.

BRNL241E **XXXX Address Space failed to transition to non-swappable state within XXXX minutes**

Explanation:
The address space failed to transition to the non-swappable state within the state number of minutes.

User response:
Retry. If the problem reoccurs, contact Technical Support.

BRNL243E **XXXX Job with JES number XXXX has invalid log block log block ID**

Explanation:
The stated job with the displayed JES number has an invalid log block. The log block ID is shown.

User response:

BRNL244I **XXXX RESTART FOR JOB XXXX XXXX**

Explanation:
The stated job is restarting.

User response:
None. Informational message.

BRNL245I **XXXX STARTING RESTART PROCESSING TO RESOLVE JOBS THAT WERE ACTIVE WHEN TIMELINER ENDED**

Explanation:
The system is restarting TimeLiner again, as jobs were still active before TimeLiner was successfully closed.

User response:
None. Informational message.

BRNL246I **XXXX FINISHED RESTART PROCESSING XXXX JOBS WERE IDENTIFIED**

Explanation:
The restart processing has been completed. The jobs which were identified by the process are listed.

User response:

None. Informational message.

BRNL247I **XXXX JOB XXXX XXXX ON XXXX IS NO LONGER ACTIVE**

Explanation:
The interrupted job is no longer active. The job name, job ID, and system are given in the message.

User response:
None. Informational message.

BRNL248I **XXXX JOB XXXX XXXX ON XXXX IS STILL ACTIVE**

Explanation:
Interrupted job is still active. The job name, job ID, and system are given in the message.

User response:
None. Informational message.

BRNL249I **XXXX JOB XXXX XXXX RESTART PROCESSING BEGINNING FOR XXXX JOB**

Explanation:
Restart processing is starting. The job name and job ID are given in the message.

User response:
None. Informational message.

BRNL250I **XXXX JOB XXXX XXXX RESTART PROCESSING COMPLETED FOR XXXX JOB**

Explanation:
Restart processing completed. The job name and job ID are given in the message.

User response:
None. Informational message.

BRNL255E **XXXX Unable to write to logstream after XXXX attempts**

Explanation:
Writing of data to the log stream has timed out after the specified number of attempts.

User response:
Retry. If the problem reoccurs, contact Technical Support.

BRNL258I ***mod_name rtn_name* INPUT *Parm1 parm2 parm3 parm4***

Explanation:
Generic module entry debug message.

User response:
None. Informational message.

BRNL259I ***mod_name rtn_name* RC AT EXIT *return_code***

Explanation:
Generic module exit debug message.

User response:
None. Informational message.

BRNL260I ***mod_name rtn_name v1 v2 v3***

Explanation:
Generic module display message.

User response:
None. Informational message.

BRNL261E ***mod_name rtn_name service_name return_code RS***

Explanation:
Generic error message for unexpected failures.

User response:
None. Informational message.

BRNL262I ***mod_name* Job *jobname jobid sysid* was active when this TimeLiner last ended**

Explanation:
Indicates the job was active when TimeLiner went down.

User response:
None. Informational message.

BRNL263I ***mod_name rtn_name JINA|JINC status* built for *jobname jobid* on *sys***

Explanation:
Indicates JINA or JINC built for interrupted job.

User response:
None. Informational message.

BRNL264I ***ModName RtnName* TimeLiner Start TOD *time(hex)***

Explanation:
Show TimeLiner STC Start time.

User response:
None. Informational message.

BRNL265I ***Jobname* is TimeLiner eligible *optional-string(x datasets copied)***

Explanation:
Indicates the job is TimeLiner eligible and optionally, the number of data sets that were backed up.

User response:
None. Informational message.

BRNL266I **TimeLiner initialization complete**

Explanation:
Indicates TimeLiner STC has completed initialization. This message is written by WTO to JESMSGLG and not by BRNLMSGM.

User response:
None. Informational message.

BRNL267E **Error returned from VDB: *error_message***

Explanation:
VDB returned an error when it was called to update the VSAM backup table.

User response:
Contact Technical Support.

BRNL268E **TLNSVCDP Failed to capture dump - SDUMPX RETURN CODE *reason_code* REASON CODE *rs***

Explanation:
Indicates that the recovery routine for SVCs was unable to issue SDUMPX. This message is issued directly by TLNSVCDP via WTO.

User response:
Contact Technical Support.

BRNL269E **BRNLCTLP SVC *nn* FRONTEND *nnnnnn* DISABLED**

Explanation:
Indicates that TimeLiner disabled the code for one of its SVC frontend modules. This means that TimeLiner is not functioning properly. This message is issued as a hardcoded WTO and so is displayed in JESMSGLG.

User response:
Look at message 279S to determine which job experienced the error.

BRNL270E **XXXX CHECK JOBNAME *nnnnnnnn* JOBID *nnnnnnnn***

Explanation:
Indicates the job that experienced the abend that caused TimeLiner to disable its SVC front end module. This message is issued as a hardcoded WTO and so is displayed in JESMSGLG.

User response:
Check the system log and job log for information about dumps.

BRNL271E ***mmmmmmmm text***

Explanation

Indicates a TimeLiner SVC front end abended. *mmmmmmmm* is the name of the TimeLiner recovery module and *text* is a description of the error. This is a severe error because it indicates that TimeLiner may not have been able to collect diagnostic data such as a dump or LOGREC information, and may not have been able to disable the exit.

This error is written via WTO in the user job in which the abend occurred. In general this message is issued as a hard coded WTO rather than through BRNLMSGM.

User response:
Contact Technical Support.

BRNL272E ***mmmmmmmm* SVC *nn* FRONTEND ABENDED**

Explanation:
Indicates a TimeLiner SVC front end abended. *mmmmmmmm* is the name of the TimeLiner SVC front end module and *nn* is the number of the SVC. This is a severe error because it indicates that TimeLiner had to disable some of its data collection and therefore job information is incomplete. In general this message is issued as a hard coded WTO rather than through BRNLMSGM.

User response:
Contact Technical Support.

BRNL273I ***mmmmmmmm text***

Explanation:
Informational debug message issued by module *mmmmmmmm*. In general this message is issued as a hard coded WTO rather than through BRNLMSGM.

User response:
None. Informational message.

BRNL274I ***mmmmmmmm* MEMBER *token* WAS *former state***

Explanation:
IXCSETUS issued to change a member's state.

User response:
None. Informational message.

BRNL275I ***mmmmmmmm* MEMBER *token* IS NOW *new state***

Explanation:
IXCSETUS issued to change a member's state.

User response:
None. Informational message.

BRNL276I ***mmmmmmmm* CURRENT MEMBER TOKEN IS *token***

Explanation:
Sysplex token of the current TimeLiner STC.

User response:
None. Informational message.

BRNL277I **TimeLiner *command text***

Explanation:
Status of the indicated command issued by the user.

User response:
None. Informational message.

BRNL278I ***mmmmmmmm* $DIVSAVE *label text text rc rs***

Explanation:
Debug from $DIVSAVE macro.

User response:
None. Informational message.

BRNL279E **mmmmmmmm DIV SAVE FAILED REASON CODE = *reason_code* REASON CODE = *rs* $DIVSAVE AT *label text***

Explanation:
Error from $DIVSAVE macro.

User response:
Contact Technical Support.

BRNL280E **An error occurred attempting to add a job to the binary tree. See the BRNLCTRL log for more information.**

Explanation:
WTO message.

User response:

BRNL281E **XXXX Duplicate table entry found. Jobname XXXX bypassed**

Explanation:
The same jobname has been requested more than once.

User response:
Remove the duplicate entry and retry.

BRNL282E **The SWAREQ for the JFCB has failed. See the BRNLCT00 and BRNLCTRL log for more information.**

Explanation:
WTO message.

User response:
See the BRNLCT00 and BRNLCTRL log for more information, then contact Technical Support.

BRNL283E **BRNLSVCF Internal error - *text***

Explanation:
WTO message.

User response:
Contact Technical Support.

BRNL284E **BRNLSVCF TimeLiner disabled. A dump will be created for this error.**

Explanation:
WTO message.

User response:
Contact Technical Support.

BRNL285E **BRNLCTLP TimeLiner shutting down. An error occcurred in one of our exit or SVC modules**

Explanation:
WTO message. An abend (or unrecoverable error) occurred in an SMF exit or SVC front-end module.

System action:
TimeLiner shuts down.

User response:
Look at the system job log to determine which user job failed. Collect information and the dump, and contact Technical Support.

BRNL286E **XXXX TimeLiner module *mod_name* detected invalid *stg_type* at *address***

Explanation:
WTO message. An abend occurred in an SMF exit or SVC front-end module when attempting to validate user storage. This is a user error.

System action:
TimeLiner shuts down.

User response:
Check error, start TimeLiner and retry.

BRNL287E **XXXX TimeLiner module *mod_name* unable to collect XXXX data for a data set**

Explanation:
WTO message.

User response:
Contact your System Programmer with the details of the message.

BRNL288I **Application Table use count Shutdown delayed**

Explanation:
WTO message in CTLPR.

User response:
None. Informational message.

BRNL289I **The record buffer is not empty - Shutdown delayed**

Explanation:
WTO message in CTLPR.

User response:
None. Informational message.

BRNL290S **Attempt to acquire common storage for the XXXX failed with RETURN CODE: *return_code***

Explanation:

Note the return code and retry. If the problem reoccurs, contact Technical Support.

User response:

BRNL291E ***mod_name*** **STORAGE RELEASE OF xxxxxxxx FOR *nnnnnnnn* BYTES AT *label* FAILED RETURN CODE: *return_code***

Explanation:
Indicates a storage release of the indicated area.

User response:
Note the return code and retry. If the problem reoccurs, contact Technical Support.

BRNL292I ***mod_name*** **(RE)LOAD OF APPLICATION TABLE**

Explanation:
The application table is reloading.

User response:
None. Informational message.

BRNL293I ***modname*** **TLNAPPHD: *addr1* APPTABLE: *addr2* APTTINFOX: *addr3***

Explanation:
Shows storage locations of newly loaded application table related data areas.

User response:
None. Informational message.

BRNL294I ***mod_name*** **INIT APPTBL STRUCTS AT *label* CURRENT TLNA: *addr1* OLD TLNA: *addr2***

Explanation:
At TimeLiner STC startup, the structures are initialized that contains the app table data when loaded.

User response:
None. Informational message.

BRNL295E ***module*** **DISPLAY *subcommand+* TOO LONG - MAX LEN IS *len* CHARS**

Explanation:
A user has issued a DISPLAY command to TimeLiner, but the subcommand is too long.

User response:
Correct the command and retry.

BRNL296I **Reload APP Table in progress - STOP ignored**

Explanation:
WTO message issued by BRNLCTLP. Cannot STOP because the APP Table is reloading.

User response:
Informational message. Wait for the APP table to reload and then retry.

BRNL297E ***module*** **DISPLAY *subcommand* IS INVALID**

Explanation:
A user has issued a DISPLAY subcommand that is not valid.

User response:
Correct the subcommand and retry.

BRNL298E ***module*** **VALID DISPLAY SUBCOMMANDS ARE**

Explanation:
This message lists valid DISPLAY subcommands.

User response:
Take note of valid subcommands and select the relevant command to send.

BRNL299I ***module*** **COUNT OF ACTIVE JOBS BEING MONITORED: *count***

Explanation:
The number of jobs currently being monitored by TimeLiner.

User response:
None. Informational message.

BRNL300I ***module*** **COUNT OF JOBS USING CURRENT APPTABLE: *count***

Explanation:
The number of jobs examining the current Application Table to determine whether this job should be monitored by TimeLiner.

User response:
None. Informational message.

BRNL301I ***module*** **COUNT OF JOBS USING FORMER APPTABLE: *count***

Explanation:
The number of jobs examining an older version of the Application Table to determine whether this job should be monitored by TimeLiner. The count should only be greater than zero if a RELOAD was done and some jobs were still in the process of examining the older Application Table.

User response:
None. Informational message.

BRNL302I ***module*** **MEMBER: *membername* STATUS: *status***

Explanation:
There is one message for each member of the TimeLiner XCF group. This message gets the data from TimeLiner internal control blocks as opposed to XCF services. MEMBER is the member name. STATUS

indicates whether this is a PRIMARY or SECONDARY TimeLiner member.

User response:
None. Informational message.

BRNL303E ***module function*** **INVALID GROUPNAME:** ***name*** **LEN:** ***len***

Explanation:
The group name passed to *function* is invalid. It is either a null string or a name that is greater than 8 characters.

User response:
Correct the group name and retry.

BRNL304I ***module*** **MEMBER:** ***mbrname*** **JOBNAME:** ***job*** **STATE:** ***state*** **STATUS:** ***status***

Explanation:
Information about the *mbrname* in the XCF group.

User response:
None. Informational message.

BRNL305E ***module function*** **MALLOC FAILED**

Explanation:
The MALLOC issued in the indicated *module function* failed.

User response:
Examine the joblog for more information.

BRNL306E ***module*** **IXCQUERY** ***explanation***

Explanation:
IXCQUERY failed. The *explanation* in the message describes the reason.

User response:
Correct the error and retry. If the error reoccurs, contact your System Programmer with the contents of the message.

BRNL307E ***module*** **IXCQUERY RETURN CODE:** ***return_code*** **RESET:** ***reset***

Explanation:
IXCQUERY failed. The return code and reason code are displayed.

User response:
Correct the error and retry. If the error reoccurs, contact your System Programmer with the contents of the message.

BRNL308E ***module function*** **INVALID LEVEL** ***level***

Explanation:
The level (a string representation of the name token level) is invalid. Level can only be TASK, HOME, PRIMARY, or SYSTEM.

User response:
Correct the level value and retry.

BRNL309E ***module function*** **INVALID LEVEL** ***level***

Explanation:
The level (an integer representation of the name token level) is invalid. Level can only be 1, 2, 3, or 4.

User response:
Correct the level value and retry.

BRNL310E ***module function*** **IEANTRT RC:** ***return_code***

Explanation:
The Name Token query service failed with RC *return_code*.

User response:
Note the return code and retry. If the problem reoccurs, contact your System Programmer.

BRNL311E ***module function*** **INVALID NTTP EYECATCHER AT** ***address***

Explanation:
The NTTP entry at the indicated *address* is invalid.

User response:
Correct the address error and retry.

BRNL312E ***module*** **NO JOBS ARE CURRENTLY BEING MONITORED**

Explanation:
Unable to locate name token for active jobs.

User response:
Contact your System Programmer with the details of the message.

BRNL313I ***module*** **JOB ID** ***jobid*** **CURRENTLY BEING MONITORED**

Explanation:
Job id *jobid* is active and is being monitored.

User response:
None. Informational message.

BRNL314E **TimeLiner terminating due to abend in** ***module*** **JOB:** ***jobname jobid***

Explanation:
The TimeLiner exit module abended in *jobname jobid*.

System action:
TimeLiner terminates.

User response:
Contact your System Programmer with the contents of the message.

BRNL315E ***module*** **BUFFER WAIT LIMIT EXCEEDED AT LABEL** ***label***

Explanation:
The BUFRWAIT in BRNLCTLP exceeded 4.

System action:
TimeLiner remains active.

User response:
Debugging message. Contact your System Programmer for more information.

BRNL316I ***module*** **DS_CLEAR R14-R1** ***values***

Explanation:
TLNACTRT is clearing DS storage.

User response:
None. Informational message.

BRNL317I ***module label svc-name*** **SVC** ***svc-no*** **table entry updated with original value xxxxxxxx**

Explanation:
TimeLiner is terminated due to an abend in one of its SVC modules. If the associated TimeLiner entry is still in the table, TimeLiner replaces that value with the value that was there when TimeLiner started.

User response:
None. Informational message.

BRNL318I **TimeLiner forcing REINIT(YES) to add TimeLiner modules to SVC table**

Explanation:
TimeLiner is terminated due to an abend in one of its SVC modules. During termination, TimeLiner replaced one or more SVC table entries. TimeLiner is restarting with REINIT(YES) to ensure that TimeLiner SVC modules are re-added to the SVC table.

User response:
None. Informational message.

BRNL319E **TimeLiner unable to update job index RC =** ***retcode***

Explanation:
WTO message. An error occurred when trying to chain the OJX blocks. More detailed error messages appear in the TimeLiner STC output.

User response:

BRNL320E ***module routine*** **addToChain unable to add OJX to chain**

Explanation:
WTO message. An error occurred when trying to chain the OJX blocks. The OJX in question and the Header anchor fields are displayed.

User response:
Correct the error and retry.

BRNL321I ***module %d*** **job index records initialized**

Explanation:
WTO message. Prior to building the chain, TimeLiner first clears existing chain pointers.

User response:
None. Informational message.

BRNL322I ***module routine %d*** **job index records chained**

Explanation:
WTO message. *%d* number of index records were chained in start date time order.

User response:
None. Informational message.

BRNL323I ***module routine*** **no OJXs have been used**

Explanation:
WTO message. The scan of the OJXs indicate that no OJX records have job data. This is most likely a new DIV.

User response:
None. Informational message.

BRNL324I ***module routine*** **No jobs are on start date time chain**

Explanation:
WTO message. The scan of the OJXs indicate that no OJX records have job data. This is most likely a new DIV.

User response:
None. Informational message.

BRNL325E ***module routine*** **Possible error in scan** ***%d*** **records reported**

Explanation:
WTO message. The number of jobs scanned does not match the number of jobs in the DIV.

User response:
Check the number of jobs, correct the error and retry.

BRNL326E ***module %d*** **possible jobs in DIV**

Explanation:
WTO message. This message is paired with BRNL325E and contains the total possible jobs in the DIV.

User response:
See BRNL325E for details.

BRNL327E ***module routine*** **Invalid DIV: OHR eye is** ***%s***

Explanation:
WTO message. The DIV header does not contain a valid eyecatcher.

User response:
Correct the error and retry.

BRNL328I ***module routine* DIV HDR anchor fields for *low_jobid*, *high jobid***

Explanation:
WTO message. The newest job on the start date-time chain and the oldest job on the start date-time chain are shown.

User response:
None. Informational message.

BRNL329I ***BRNLCUSDT printOJX* zero OJX pointer passed**

Explanation:
WTO message. An OJX with an offset of 0 was passed to *printOJX*.

User response:
None. Informational message.

BRNL330I **OJX *ojxoffset*, *jobid*, *jobname*, *job_start_time prev next***

Explanation:
WTO message. Information about OJX at offset *ojxoffset*.

User response:
None. Informational message.

BRNL331I **TimeLiner beginning update of DIV job index**

Explanation:
WTO message. TimeLiner is chaining the OJX entries in start date-time order.

User response:
None. Informational message.

BRNL332I **TimeLiner completed update of DIV job index**

Explanation:
WTO message. TimeLiner chained the OJX entries in start date-time order.

User response:
None. Informational message.

BRNL333E **TimeLiner *module label* unable to obtain LATCH REASON CODE *reason_code***

Explanation:
TimeLiner could not obtain the latch it needed to process data records. The name of the module is shown, and *label* is the location where the error occurred.

System action:
TimeLiner shuts down.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL334E **TimeLiner *module label* unable to release LATCH REASON CODE *reason_code***

Explanation:
TimeLiner could not release a latch it had obtained to process data records. The name of the module is shown, and *label* is the location where the error occurred.

System action:
TimeLiner shuts down.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL335E **TimeLiner *module label* RECOVERY address of *area* is zero**

Explanation:
An abend occurred in a TimeLiner module. The associated recovery routine could not process correctly because the address of a TimeLiner control area is zero.

System action:
TimeLiner shuts down.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL336E **TimeLiner *module label* UNKNOWN ERROR in retry RC**

Explanation:
An abend occurred in a TimeLiner module. The associated recovery routine could not process correctly because the return code was not expected.

System action:
TimeLiner shuts down.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL337E **XXXX XXXX RECORD COULD NOT BE PLACED IN DATASPACE BECAUSE DATASPACE FULL AFTER WRAP**

Explanation:
The dataspace buffer has wrapped. There is not enough room between the next available byte and the last read byte to hold the new buffer.

User response:

BRNL338E **TimeLiner *module* recursive abend**

Explanation:
A recovery routine was entered recursively. The routine percolates.

User response:

Retry. If the error reoccurs, contact Technical Support.

BRNL339E ***module*** **Internal error - Abend occurred**

Explanation:
An abend occurred in the named module.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL340E **BRNLCT00 TimeLiner shutting down due to BRNLPXMR error**

Explanation:
An error occurred in the XCF message exit BRNLPXMR. Issued as WTO.

System action:
TimeLiner shuts down.

User response:
See TimeLiner and system log information for error information.

BRNL341I **BRNLATPC TimeLiner Common Data buffer wrap**

Explanation:
The data in the common data buffer has wrapped. Issued as WTO.

User response:
None. Informational message.

BRNL342E **BRNLATPC TimeLiner recovery did not receive expected parameters**

Explanation:
. An abend occurred in BRNLATPC. The recovery routine did not receive expected input parameters. Issued as WTO.

System action:
TimeLiner terminates.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL343I **BRNLATPC TimeLiner released LATCH**

Explanation:
An abend occurred in BRNLATPC. The recovery routine determined that a latch was held and it released the latch. Issued as WTO.

User response:
None. Informational message.

BRNL344E **BRNLPLMR TimeLiner released LATCH**

Explanation:
An abend occurred in BRNLPLMR. The recovery routine determined that a latch was held and it released the latch. Issued as WTO

User response:
Contact Technical Support.

BRNL345E **BRNLDFPC ETCRE ET token is *token* Routine index is *index***

Explanation:
Displays information about establishment of the PC used by TimeLiner. Issued as WTO.

User response:
None. Informational message.

BRNL346E **BRNLDLPC Entry Table does not exist**

Explanation:
The Entry Table token for the TimeLiner PC does not exist. Issued as WTO.

System action:
TimeLiner continues shutdown.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL347I **BRNLDLPC ETDES for ET *et* completed with *return_code***

Explanation:
The Entry Table is destroyed. Issued as WTO.

System action:
TimeLiner continues shutdown.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL348E **BRNLATCH createLatch ISGLCRT failed REASON CODE *reason_code***

Explanation:
TimeLiner is unable to create a latch set. Issued as WTO.

System action:
TimeLiner shuts down.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL349E **BRNLATCH getLatch ISGLID failed RC *rc hexrc***

Explanation:
TimeLiner is unable to identify the latch set. Issued as WTO.

System action:
TimeLiner shuts down.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL350E **BRNLATCH purgeAllLatches ISGLPBA failed RC *rc hexrc***

Explanation:

TimeLiner is unable to purge latch set. Issued as WTO.

System action:
TimeLiner continues shutdown.

User response:
Retry. If the error reoccurs, contact Technical Support.

BRNL351E BRNLCT00 PSW KEY not 4, Check PPT entry for BRNLCT00

Explanation:
TimeLiner is not running in key 4. Issued as WTO.

System action:
TimeLiner continues shutdown.

User response:
Verify that there is an entry in SCHEDxx for BRNLCT00 with KEY 4 specified.

BRNL996I *mod_name message*

Explanation:
WTO message. Generic debug messages for Metal C support.

User response:
None. Informational message.

BRNL997I *mod_name message*

Explanation:
WTO message. Generic debug messages for BRNLCT00/BRNLCTLP.

User response:
None. Informational message.

BRNL998I *mod_name message*

Explanation:
WTO message. Generic debug messages for exits.

User response:
None. Informational message.

BRNL999E XXXX XXXX XXXX XXXX

Explanation:
999 is reserved for hardcoded WTO messages.

User response:

Appendix I. IBM Z Batch Resiliency Abend Codes

RTSMF Abend Codes

Table 75. RTSMF abend codes

Code	Module	Abend	Reason/Response

U0401	CKKRC008	NO COMMAND ENTERED	RTDPRINT MESSAGE:
			RTS008003E - NO COMMAND WAS PASSED; PROGRAM TERMINATING ENTER COMMAND AS PARM FIELD; E.G.,
			//STEP1 EXEC PGM=CKKRC008,
			// PARM='DISPLAY TABLES'

U0402	CKKRC008	INVALID COMMAND ENTERED	RTDPRINT MESSAGE: RTS008004E - INVALID COMMAND WAS PASSED; PROGRAM TERMINATING COMMAND INPUT 8 OR LESS CHARACTERS
U0403	CKKRC008	COMMAND PROCESSOR BUSY	RTDPRINT MESSAGES: RTS008002I - COMMAND PROCESSOR BUSY; WAITING FOR THE COMMAND PROCESSOR TO BECOME AVAILABLE RTS008005E - COMMAND PROCESSOR STILL BUSY; PROGRAM TERMINATING
U0404	CKKRC008	COMMAND QUEUE DSN STILL IN USE	RTDPRINT MESSAGES: RTS008007I - COMMAND QUEUE DATASET IN USE BY ANOTHER JOB/USER; WAITING FOR DATASET TO BECOME AVAILABLE RTS008008E - COMMAND QUEUE DATASET STILL IN USE; PROGRAM TERMINATING
U0601	BRNVC006	UNABLE TO LOCATE APPTABLE OR APPTABLX TABLES	View messages in the RTDPRINT file then contact Technical Support if needed.
U0800	BRNVC008	DYNAMIC DATASET SCHEDULING NOT ACTIVATED	Verify that an entry for VFI/RTS is in VFISTASK and is not commented out.
U0801	BRNVC008	ERROR OBTAINING STORAGE	Contact Technical Support.
U0802	BRNVC008	ERROR ATTEMPTING TO LOAD MODULE	Contact Technical Support.

© Copyright IBM Corp. 2019, 2020

Table 75. RTSMF abend codes (continued)

Code	Module	Abend	Reason/Response
U0803	BRNVC008	ERROR READING SCHDPARM	Contact Technical Support.
U0804	BRNVC008	SYNTAX ERROR IN SCHDPARM	Contact Technical Support.
U0805	BRNVC008	SCHEDREQ TABLE NOT LOCATED	Verify that SCHEDREQ exists in USERMAC. and that an entry for SCHEDREQ exists in VFITABLS.
U0806	BRNVC008	SRA NOT LOADED	Contact Technical Support.
U0807	BRNVC008	ERROR LOADING SCHEDULE EVENT ARRAYS	Contact Technical Support.
U0808	BRNVC008	ERROR LOADING SCHEDULE	Contact Technical Support.
U0809	BRNVC008	OPEN FAILED FOR INVENTRZ	Verify that DDNAME INVENTRZ exists in RTSMF JCL.
U080A	BRNVC008	INVENTORY FILE NOT OPEN/AVAILABLE	Contact Technical Support.
U0999	CKKRC001	AUTH ERROR; C21LIC INVALID OR OUT OF DATE	Contact Technical Support.
U1001	CKKRC001	OPEN/PROCESSING ERROR ON PARAMETER DATASET; DDNAME=PARMLIB	
U1001	BRNVC012	INVENTRY FILE INIT ERROR (NON-RT/SMF) CONV RTN	Contact Technical Support.
U1002	CKKRC001	PARM MEMBER RTSRTD00 MISSING	ABEND 1002 ISSUED WHEN RTSRTD00 (OR MEMBER SPECIFIED ON EXEC PARM) IS NOT PRESENT IN THE PARMLIB DATASET
U1002	BRNVC012	INVENTRY FILE OPEN ERROR (NON-RA/SMF) CONV RTN	Contact Technical Support.
U1003	CKKRC001	SMFPARMS ERROR	ABEND 1003 ISSUED WHEN SMFPARMS (OR MEMBER SPECIFIED IN RTSRTD00) IS NOT PRESENT IN THE PARMLIB DATASET OR AN ERROR OCCURRED

Table 75. RTSMF abend codes (continued)

Code	Module	Abend	Reason/Response
U1004	CKKRC001	VFIPARMS OR MEMBER SPECIFIED IN RTSRTD00 MISSING	ABEND 1004 ISSUED WHEN VFIPARMS (OR MEMBER SPECIFIED IN RTSRTD00) IS NOT PRESENT IN THE PARMLIB DATASET OR AN ERROR OCCURRED
U1005	CKKRC001	CKKRTASK ERROR ENCOUNTERED	ABEND 1005 ISSUED WHEN CKKRTASK (OR MEMBER SPECIFIED IN RTSRTD00) IS NOT PRESENT IN THE PARMLIB DATASET OR AN ERROR OCCURRED
U1006	CKKRC001	MAXIMUM TASKS ENCOUNTERED	ABEND 1006 ISSUED WHEN CKKRTASK CONTAINS MORE START COMMANDS THAN ARE SPECIFIED IN &MAXTAS (&MAXTAS=9)
U1007	CKKRC001	ERROR READING SCHDREQ	ABEND 1007 ISSUED WHEN UNEXPECTED ERROR ENCOUNTERED WHILE READING THE SCHEDULING REQUEST ARRAY USERMAC. MEMBER (SCHDREQ)
U1008	CKKRC001	VFITABLS MEMBER MISSING	ABEND 1008 ISSUED WHEN VFITABLS (OR MEMBER SPECIFIED IN RTSRTD00) IS NOT PRESENT IN THE PARMLIB DATASET OR AN ERROR OCCURRED
U1009	CKKRC001	OPEN ERROR ON INVENTORY FILE	
U1010	CKKRC001	CIB ERROR; NOT STOP OR MODIFY COMMAND	
U1011	CKKRC001	CIB/QEDIT ERROR	
U1999	BRNVC001	TABLE LOCATE ERROR: APPTABLE; EXCLJOB; EXCLDSN	REVIEW VFITABLS USERMAC. MEMBER, ENSURE ENTRIES FOR APPTABLE, APPTABLX, EXCLDSN, EXCLJOB. Contact Technical Support.
U2000	BRNVC001	BRNVC020 OR BRNVC021 LOAD ERROR; INVENTRY NOT THERE	THIS ABEND ALSO OCCURS ON A BAD RETURN CODE FROM BRNVC021 , THE INVENTRY FILE MANAGER. REVIEW RTDPRINT FOR MESSAGE AND CODES RELATED TO OPENING THE INVENTRY FILE OR BUILDING THE LSR POOLS. Contact Technical Support.
U2001	CKKRC002	OPEN ERROR ON RTDPRINT DD STATEMENT OR GETMAIN FAILURE	REVIEW JOB LOG FOR ERROR MESSAGES, ENSURE //RTDPRINT DD STATEMENT IN JCL

Table 75. RTSMF abend codes (continued)

Code	Module	Abend	Reason/Response
U2001	CKKRS002	SPECIFIC/GENERIC PDS REQUEST ERROR	ABEND 2001 IS ISSUED IF USER CALLS CKKRS002: WITH GENERIC MEMBER NAME AFTER OPENING THE DATASET FOR SPECIFIC MEMBER INPUT REQUESTS RULE IS: IF YOU WANT TO USE GENERIC REQUESTS, YOU MUST USE ONE ON FIRST CALL FOR EACH DATASET. SUBSEQUENT CALLS MAY BE GENERIC OR SPECIFIC. Contact Technical Support.
U2002	CKKRS002	REQUEST NOT **CLOS** FOR UN-OPENED DATASET	ABEND 2002 IS ISSUED IF USER CALLS CKKRS002: FOR SAME DATASET AFTER PREVIOUS CALL RETURNED A R/C 12 (UNABLE TO OPEN). USER MUST CALL 'CKKRS002' WITH MBR_NAME = '**CLOS**' BEFORE CALLING FOR ANY OTHER ATTEMPT TO ACCESS THE UNOPENED DATASET. Contact Technical Support.
U2003	CKKRS002	CKKRS002 RE-ENTRY INVALID FOR REQUEST	ABEND 2003 IS ISSUED IF USER CALLS CKKRS002: FOR SAME MEMBER AFTER 'MEMBER EOF CONDITION. Contact Technical Support.
U2004	CKKRS003	ERROR RETURNED ON STOW	USER ABEND 2004 IS ISSUED IF STOW SETS A RETURN CODE INDICATING AN ERROR DUE TO 'CKKRS003' LOGIC. THE RETURN CODE IS IN REG-15.....R/C 00; 12; 16 & 24 IS RETURNED TO CALLER; OTHERS CAUSE U001..16 WITH A REASON CODE = 1 IS RETURNED AS R/C 16...16 WITH A REASON CODE = 2 IS RETURNED AS R/C 20.
U2005	CKKRS004	TOO MANY PARMS; KEYWD > 8 CHAR OR 0 CHAR	ABENDS: USER 2005 - PARM-2 KEYWORD VALUE DOES NOT CONTAIN EQUAL SIGN KEYWORD MUST BE IN FORM XXXXXXXX=WHERE XXXXXXXX IS 1 TO 8 CHARACTERS. Contact Technical Support.
U3101	CKKRC013	UNABLE TO OBTAIN CTL TABLE STORAGE	Contact Technical Support.
U3102	CKKRC013	INVALID FUNCTION SPECIFIED (PARM#1)	Contact Technical Support.
U3103	CKKRC013	INCORRECT NUMBER OF PARMS FOR FUNCN	Contact Technical Support.

Table 75. RTSMF abend codes (continued)

Code	Module	Abend	Reason/Response
U3104	CKKRC013	'OPEN' - UNABLE TO OBTAIN STORAGE	Contact Technical Support.
U3105	CKKRC013	'SORT' - UNABLE TO GETMAIN SWORT WRK AREA	Contact Technical Support.
U3106	CKKRC013	'CLOSE' - PROBABLE LOGIC ERROR	Contact Technical Support.
U3107	CKKRC013	'RESET' - 2ND PARM NOT "I" OR "O"	Contact Technical Support.
U3108	CKKRC013	'SORT' - QIOFSORT SET NON-ZERO R/C	Contact Technical Support.
U3109	CKKRC013	'SORT' - # SORT KEYS < 0 OR > 15	Contact Technical Support.
U3110	CKKRC013	'SORT' - KEY-OFFSET < 0 OR > 4096	Contact Technical Support.
U3111	CKKRC013	'SORT' - INVALID FORMAT TYPE SPEC'D	Contact Technical Support.
U3200	BRNVC007	BKUPTBL PROCESSING ERROR; CHECK RTDPRINT	RTDPRINT MESSAGES: RTSRTB006E - mm/dd/yy hh:mm:ss - BRNVC007: BKUPTBL QIO PUT ERROR; RC=> RTSRTB014E - mm/dd/yy hh:mm:ss - BRNVC007: LOAD ON CKKRS006 FAILED; RC= RTSRTB021E - mm/dd/yy hh:mm:ss - BRNVC007: INCORE GETMAIN FAILURE; TERMINATING RTSRTB022E - mm/dd/yy hh:mm:ss - BRNVC007: INCORE TABLE EXCEEDED; TERMINATING Contact Technical Support.
U3201	BRNVC010	GLOBAL STATUS PROCESSING ERROR	RTDPRINT MESSAGE: VFI010000E - mm/dd/yy hh:mm:ss - INVENTRY GLOBAL STATUS PROCESSOR ABENDING; CODE: Contact Technical Support.

Table 75. RTSMF abend codes (continued)

Code	Module	Abend	Reason/Response
U3202	BRNVC006	LOAD FAIL; BRNVC016 ; BRNVC009 ; BRNVC010	RTDPRINT MESSAGE: VFI006006E - mm/dd/yy hh:mm:ss - XXXXXXXX NOT AVAILABLE, PROCESS TERMINATING. REVIEW MESSAGE FOR MISSING PROGRAM, ENSURE PROGRAMS IN STEPLIB Contact Technical Support.
U3203	BRNVC006	INVENTRY VSAM ERRORS	RTDPRINT MESSAGES: BRNVC006006I - mm/dd/yy hh:mm:ss - :R15(XXXXXXXX)FEEDBACK (XXXXXXXX) EVALUATE RETURN AND FEEDBACK CODES ALONG WITH VARIABLE TEXT. Contact Technical Support.
U3208	BRNVC013	VARIOUS ERRORS; REVIEW RTDPRINT MESSAGES	RTDPRINT MESSAGES: BRNVC01302E - mm/dd/yy hh:mm:ss - TABLE LOAD FAILED: BRNVC01304E - mm/dd/yy hh:mm:ss - APPTABLE ENTRY NOT FOUND BRNVC01305E - mm/dd/yy hh:mm:ss - VFIOANDX NOT FOUND BRNVC01306E - mm/dd/yy hh:mm:ss - VFIOANDX COUNT IS ZERO BRNVC01307E - mm/dd/yy hh:mm:ss - VFIOANDX SLOTS EXCEEDED Contact Technical Support.
U3209	BRNVC013	INVENTRY VSAM ERRORS	RTDPRINT MESSAGES: BRNVC02001I - mm/dd/yy hh:mm:ss - :R15(XXXXXXXX)FEEDBACK(XXXXXXXX) EVALUATE RETURN AND FEEDBACK CODES ALONG WITH VARIABLE TEXT. Contact Technical Support.
U3210	BRNVC020	KEYDHIST/KEYJHIST VSAM ERROR	RTDPRINT MESSAGES: BRNVC0130121I - mm/dd/yy hh:mm:ss - :R15(XXXXXXXX)FEEDBACK(XXXXXXXX) EVALUATE RETURN AND FEEDBACK CODES ALONG WITH VARIABLE TEXT. Contact Technical Support.

Table 75. RTSMF abend codes (continued)			
Code	**Module**	**Abend**	**Reason/Response**
U3230	CKKRC030	DYNAMIC ALLOCATION ERROR	RTDPRINT MESSAGE: RTS030005E - mm/dd/yy hh:mm:ss - CATALOG ERROR - CSICCFLG = XX - DSN =XXXXXXXX.XXXXXXXX.XXXXXXXX. XXXXXXXX.XXXXXXXX Review CSICCFLG. Contact Technical Support.
U3231	CKKRC030	CHECKPOINT FILE SAVE ERROR	Contact Technical Support.
U3233	CKKRC030	QIO LOAD ERROR	View messages in the RTDPRINT file then contact Technical Support if needed.
U3301	BRNVL001	NO MORE OAT INDEX SLOTS	Contact Technical Support.
U3997	CKKRC003	QIO ERROR ON CHECKPOINT LOAD/ INFO	Contact Technical Support.
U3998	CKKRC003	SQE WITH NON-ZERO RTSSQUSR FIELD	Contact Technical Support.
U3999	CKKRC003	SMF DD/FILE MANAGER(CKKRS009) ERRORS	Contact Technical Support.

SMF Extractor Abend Codes

Table 76. SMF Extractor abend codes		
Code	**Program**	**Reason**
U0100	CKKSDRVR	DRIVER
U0200	CKKSRDR	READER SUBTASK
U0300	CKKSWTR	WRITER SUBTASK
U0400	CKKSFSUB	JCL SUBTASK
U0500	CKKSWAIT	SMFDUMP ENQ PROGRAM
U0600	CKKSCCAT	ALLOCATE, CONCATENATE EXTRACT FILES

VFIGENR/VTCCMP Abend Codes

Table 77. VFIGENR/VTCCMP abend codes

Code	Reason
U0101	Read I/O error. Control blocks are dumped. Synad messages are issued (WTO only). Dump is produced.
U0102	Write I/O error. Control blocks are dumped. Synad messages are issued (WTO only). Dump is produced.
U0103	BDW appears to be incorrect. First few bytes of a record are printed. Dump is produced.

Backup Grouping (BGMAIN) Abend Codes

Table 78. BGMAIN abend codes

Code	Reason
U0901	BRNVBGOUT DDname not allocated.
U0902	BGMAIN PARM ERROR – Error occurred either in parameter passed on the EXEC statement or parameter in BRNVBGIN file.
U0903	Error attempting to invoke IRXEXEC to execute interpreted REXX exec BRNBGPRE or BGPOSTDM.

DC/Optimize Abend Codes

Table 79. DC/Optimize abend codes

Code	Program	Reason
U0001	VFIDCOPT	Max symbolics exceeded.
U1002	VFIDCOPT	Processing error.
U1003	VFIDCO03	Card search loop error.
U1004	VFIDCO04	Symbolic extract error.
U1005	VFIDCO05	In stream proc extract error.
U1006	VFIDCO06	INVENTRY file error.

IZBR-TLMS Abend Codes

Table 80. TLMS abend codes

Code	Program	Reason
U0555	TLMSBVOL	Error in next-volume processing.

Table 80. TLMS abend codes (continued)		
Code	**Program**	**Reason**
U0777	TLMSBVOL	Error processing last volume.
U0888	TLMSBVOL	Error processing last volume.
U0999	TLMSBVOL	Volume count inconsistency.
U0012	TLMSLIST	Too many parameters specified or invalid parameter.
U0999	TLMSVER	TLMS return code greater than 4.
U0555	TLMSVOL	Error in next-volume processing.
U09xx	TLMSVOL	Error processing multi-volume chain, where xx is CA Dynam/TLMS return code.
U0999	TLMSVOL	Volume count inconsistency.
U0555	VOLTLMS1	Error in next-volume processing.
U0777	VOLTLMS1	Error processing last volume.
U0888	VOLTLMS1	Error processing last volume.
U09xx	VOLTLMS1	Error processing multi-volume chain, where xx is CA Dynam/TLMS return code.
U0999	VOLTLMS1	Volume count inconsistency.

BRNVSAMX Abend Codes

Table 81. VFIVSAMX abend codes		
Code	**Program**	**Reason**
U0101	BRNVVSMD	Get error.
U0103	BRNVVSMD	Put error.
U0111	BRNVVSMD	Error opening DDIN.
U0112	BRNVVSMD	Error opening BKUPTBL.
U0999	BRNVVSMD	Error obtaining BKUPTBL dataset name.
U0101	VFIVSAMR	Get error.
U0111	VFIVSAMR	Error opening DDIN.
U0112	VFIVSAMR	Error opening BKUPTBL.

Batch Utility Abend Codes

Table 82. Batch utility abend codes

Code	Program	Explanation
U0001	BRNVB001	Due to invalid parameter list or a program load failure for module CKKRC002 or CKKRC016.
U0001	BRNVB011	Sort failure.
U2001	BRNVB020	Load failure for BRNVC020.
U0001	BRNVB021	Program load failure for module CKKRC002, CKKRC016, BRNVC022, CKKRC013.
U2101	BRNVB021	VSAM processing error. Review RTDPORINT for associated error messages.
U2201	BRNVB022	Load failure for CKKRC002, CKKRC016. INVENTRY B record put error, INVENTRY open error. See associated messages in RTDPRINT.
U2301	BRNVB023	Load failure for module CKKRC002, CKKRS006 or VSAM error on BKUPTBL.
U0001	BRNVB024	INVENTRY file VSAM error.
U0001	BRNVB033	INVENTRY file VSAM error.
U0001	BRNVB034	CRITICAL/CRITLIST mismatch error.
U0001	BRNVB037	INVENTRY file VSAM error.
U0001	BRNVB038	INVENTRY file VSAM error.
U0001	BRNVB040	Cycle Management processing error – application not found or next job error.
U0001	BRNVB041	Parameter error.
U0001	BRNVB044	INVENTRY file VSAM error.
U0001	BRNVB048	Parameter error or INVENTRY file VSAM error.
U0001	VDIDRVSM	INVENTRY file VSAM error.

Miscellaneous IBM Z Batch Resiliency Abend Codes

Table 83. Miscellaneous abend codes

Code	Program	Explanation
U3203	INVMAINT	Error while deleting an Inventory record. See messages IMNTE026 and IMNTE059 issued before the abend for more information.
x'BAD'	VDUFDR	See character string after abend for more information.
U0100	BRNVBCAT8	Error processing IDCAMS SYSPRINT record.
U0101	BRNVBCAT8	Error processing IDCAMS SYSPRINT record.
U0333	BRNVBFVR	Error processing FAVER output.
x'BAD'	BRNVCSI	See WTO for more information.
x'BAD'	BRNVCSIV	See WTO for more information.
x'BAD'	BRNVCSIZ	See WTO for more information.
U0999	BRNVLOAD	Bad IBM Z Batch Resiliency license.
U0100	VFISCAT7	Error processing IDCAMS SYSPRINT record.
U0101	VFISCAT7	Error processing IDCAMS SYSPRINT record.
U2003	BRNIBRDD	Time stamp conversion errors.
U2004	BRNIBRDD	Time stamp conversion errors.
U1002	VTOCDUMP	GETMAIN failed for vol table.
U1006	VTOCDUMP	Maximum of 680 cards exceeded.

Notices

This information was developed for products and services offered in the U.S.A. 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 U.S.A.

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 might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes are incorporated in new editions of the publication. IBM may make improvements and changes in the product(s) and the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

© Copyright IBM Corp. 2019, 2020

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements are the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. 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.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at *Copyright and trademark information* at https://www.ibm.com/legal/copytrade

Glossary

Data-In-Virtual (DIV)
DIV allows applications that access large amounts of data to create, read, and update data without the I/O buffer, blocksize, and record considerations; potentially improving performance.

Data Facility Data Set Services (DFDSS)
DFDSS is an IBM-licensed program used to copy, move, dump, and restore data sets and volumes. For more details about DFDSS see *Data Facility Data Set Services Version 2 Release 5.0: General Information*, GC26-4123.

Open Application Table (OAT)

Open Data set Table Entries (ODTE)

snapset
A snapset is a single snapshot of each volume across a group of volumes.

snapshot
A pointer-based, point-in-time image of a single volume.

© Copyright IBM Corp. 2019, 2020

Index

Special Characters

%CNTL% 12
$DELETE$ 88, 202, 207–209
$INDEX 188

A

ABACKUP 86
ABARACC 81
ABARALL 81
ABARJCL 86
ABARS
    restart 206
ABARSLGN 86
ABARSPROCNAME 86, 207
access
    ISPF panels 177
    reports 188
ACCESS 24, 25
accessibility xxi
ACID 25
activate
    processing 159
active
    jobs 220
ADD 24, 25, 29
adding 75
addition 160
address space 22, 27
ADDRESS SPACE 22
ADMS 91
ADRDSSU 118, 119, 122, 125, 165
ADSS 80
AFAVER 89
AFDR 88
AFULLDSS 128
AFULLFDR 128
AGE 143
aggregate
    recovery 206
AGGREGATE 85
ALLOC 11, 17
allocate 10
allocated 11, 122
allocating 10, 17
allocation 26, 60, 81, 89, 91, 111
allotted 80
ALTER 24, 140
alternate 20, 67
ALTSUFX 68
ampersand 32
analysis
    report 156
APAR (Authorized Program Analysis Report) xxii
APF 20, 47, 50, 86, 118, 140
APPL1 86
APPLDEF 80
application 29, 31, 32, 34, 35, 40, 43, 45, 55, 69, 73, 74, 77, 80, 82, 96, 97, 99, 135, 136, 139, 141, 142
application table 29
APPLID 29, 32, 37, 42, 43, 76, 79–81, 88, 89, 91, 93, 95, 111–113, 125, 127, 142, 143, 146, 152, 155, 171, 185, 194, 201, 203–205, 207, 209
APPTABLE 38, 149, 175, 194, 215
ARCCMDxx 86
ARCEDEXT 86
ARCSKEXT 263
ARECOVER 206, 207
ARETPD1 149, 225
ARETPD2 225
arrays 98
ARSKEXT 207
ascending time 219
ASG-Zara
    environment 212
ASG-ZEKE 29
asterisk 12, 31, 32, 43, 75, 96, 97, 100, 112, 113, 129, 131, 135, 136, 140, 143, 160
asterisk, double 143
audit
    backup table 172
    Db2 environment 161
    IMS environment 164
    report 149
AUDIT 148, 149, 225
audit gaps 1
AUDITDB 148, 149, 229
AUDITL2 11, 229
AUDTDS 229
AUDTL2 148
authority 22, 24
authorization 25, 50
authorize 24, 86
authorized 140
Authorized Program Analysis Report (APAR) xxii
authorizing 20
available 20

B

BABARS 205
back out 119
BACKCMFT 187
BACKJCL 141
backup
    CA Disk 209
    grouping messages 268
    interface messages 263
    JCL 178
    methodology 201, 209
    Report File 148
    restore methodologies 204
    table 145, 146

backup *(continued)*
table inquiry 199
TAPECPY 210
VSAM 199
BACKUP 187
Backup Dynamic Data Set Scheduling 68, 141
backup method 114
backup-skeleton 129
Backup/Restore
processing 346
BackupMethod 100
backups 1, 163, 164
BACKWO 135, 187
balancing 80
base 143
batch
job 150
jobs 215
mode 251
process 145
Batch Requestor 86, 206, 207
BAUDDEFN 13, 15
BAUDFLAT 11, 229
BDMS 209
BDSDATEQ 125
BDSNAME 125, 146
binding
the plan 161
BKJCL 38
BKP$ 141
BKPDATE 201
BKPMETH 76, 146, 201
BKPVOL 146
BKUP 42
BKUPALC 111, 203
BKUPAUD 15, 128, 129, 145, 229
BKUPBARS 86
BKUPCNTL 78, 79
BKUPDATE 136
BKUPDD 125, 127
BKUPDEFN 13
BKUPFLAT 11, 229
BKUPHLQ 128
BKUPJCL 105, 209, 229
BKUPJCLS 97, 107
BKUPJOB 128
BKUPMETH 128, 129, 166, 168, 205, 209, 229
BKUPMNT 145
BKUPPRI 140
BKUPSEC 140
BKUPTBL 111, 113, 125, 128, 131, 179, 199–201
BKUPTIME 136
BKUPTYPE 74
BKUPUNLD 145
blanks 31
BLKSIZE 56
books xxi
BRN110 50
BRNESHST 40
BRNL013I 221
BRNL289I 222
BRNLPR00 27
BRNUUMAC 5
BRNVB052 328
BRNVFMM 79
BRNVFMMC 79
BRNVJOBR 40
BRNVLEGX 356
BRXLIST 156
build
critical file list 152
data set 203
building 29, 32
business impact analysis 258
BYPASS 188

C

CA ESP 29
CA Schedule 29
CA Top Secret 25
CA Workload Automation 29
CA-DISK 91
CA7 34, 35, 37
CA7JOB 36, 37
CA7RPT 35
CANCEL 22
capture 86
cards 12, 85
Cascade Impact Report 218
cascade report 219
CATALOG 125
cataloged 139, 141–143, 156
CATBACK 95
CFLDUPS 153
CFNEW 153
CFNOCAT 153, 156
CFOLD 153
CFOLDAPP 153
CFUPD 153
change
control 262
procedures 262
TCALTTBL 247
changes in this edition xxiv
character 31, 75, 142, 160
check 20, 79
checkpoint 252, 253
CHECKPT 68
CHKALLOC 252
CHKPOINT 48, 229
CISIZE 56
class 20, 24
CMDIN
input file 206
CMDPROC 62
CMDQDEFN 13, 16
CMDQUEUE 68
CNDL 97
CNTL
BKUPDEFN 172
BKUPTBL 172
BRN110 177
CONCATRP 195
CRITREMV 171
CRITREPT 194
CROSSDEP 195
CTTVOLR 211

CNTL *(continued)*
CYCLEMGT 196
DORMANT 197
HISTRPT 194
INVBDUPS 175
INVTYRPT 195
IZBRDCOL 173
KEYDEFN 183
MEMPAPLY 159
MIGJBSUB 165
MIGJOBB 165
RESTALC 203
RMMVOLR 212
TCALTTBL 247
TLMSVOLR 210
TLNDEFDV 216
TMSUPC01 210
TMSUPC02 210, 211
TMSUPDSN 210
TMSUPVOL 210
ZDMFALLC 173
ZDMFBSIZ 173
ZDMFIZBR 173
codes 40, 161, 188, 247
COLD 98
collection 139
COLLECTX 253
command 17, 31, 82, 85
COMMAND 134
command processor 19, 61, 104, 133, 134
commands 22, 24, 35, 45, 60, 86, 104, 123
commas 129
COMMENT 113
commentary 129
compare operators 199
component 143
compress 17
CONAME 225
CONCATRP 187
configuration 29, 50, 67
configure 111
console 60, 61, 123
CONSOLE 21
control
cards 78, 171–173, 210–212
CONTROL 22
control statements
CTT 211
Control-M 29, 38
CONTROL-M 34
Control-M/Tape 211
conventions
typeface xxiii
conversion routine 159
copy
disk data sets 360
tape data sets 360
copying 27, 93
COPYPRNT
messages 277
outputting reports using 188
COUNT 60
create
jobs to allocate 203
creating 26, 38, 40
creation 35, 40, 42, 45
critical
business functions 258
business units 259
data 156
data sets 194, 197, 210, 212
file 153, 155
file list 177, 179
file recovery process 201
files 156
report 155, 194
tapes 210
CRITICAL
file names 196
Critical File List 69
CRITLEV 98
CRITNEW 155
CRITREPT 187
CRITSW 225
cross-reference
Db2 163
CROSSAPP 74
CROSSDEP 187
CSRPQPL 221
CSV 189
CSVDYNEX 24
CTM 38
CTMBCHnn 38
CTMRFLW 38
customer support
contacting IBM Support xxii
Cutoff 100
cycle 73
CYCLEMGT 187
cylinders 20, 26, 56

D

DASD 1, 10, 13, 15, 25, 26, 75, 80, 81, 88, 91, 93, 113, 128, 166, 168, 173, 262
DASD-only 22, 24
DASDVOL 125
data
analysis 173
at risk 215
encrypt 172
flow 53
repository 152
Data Protector 215
data set
encryption 175
extended
non-extended 172
history 177
history table 183
name 215
staging 215
Data Set Timeline Viewer 218
data sets
add additional 159
corrupted 215
exclude from collection 46
RESTALC 203

Data-In-Virtual 22, 26, 215, 220, 221
See also DIV
database
History 251
Inventory 168, 178, 251
schedule 185
databases 13, 161
DATATYPE(S) 76
date 1, 32, 35, 55, 62, 77, 86, 95, 100, 105, 106, 111, 129, 141, 143, 146–149, 151, 152, 225
DAYSFROMCREATION 76
Db2 161, 164
DB2BACK 163
DB2READ 161
DBA 161, 164
DC/Optimizer 2000 233
DCOLLECT 125, 128, 202
DCOPTFLE 195
DD 27, 50, 51, 61, 68, 80, 81, 85, 86, 89, 95, 97, 105, 112, 122, 125, 127, 128, 135, 142, 146
DDname 104, 127, 166, 196
DDName 35, 119
DDNAME 56, 67, 78, 79
default 127
default return codes 247
define
BKUPTBL 172
exclusions 156
generation data groups 181
defined enqueues 245
defining 15, 16, 75
definition of terms 415
DELAGE 143
DELDATE 151, 152
delete
critical files 202, 209
Delete 29
deleted 69
DELETES 210
DELFLAG 146
DELIMIT 67
DEMO 38
dense 96
DESC 42
descending time 219
description 49, 76
desired volumes 168
device-type 129
DF/DSS 1, 75, 77, 80, 96, 118, 119, 160, 201, 204, 233
DFDSS 201
DFSMS 143
DFSMSdss 1
DFSMShsm 81, 86, 206, 207
diagnostic 125
direct edit 32
disaster 69
disaster recovery
plan 257, 258, 262
planning process 257
disaster recovery plan 259
disk 8
disk volume open 118
DISP 38, 127
dispatcher 47, 49–51, 53, 61, 62, 67, 68, 97, 131, 133
display
dynamic dataset scheduling 185
DISPLAY 21, 133, 221
distribution 120
DIV 22, 26, 215, 220, 221
See also Data-In-Virtual
DMS 75, 91
DMS information 209
DMS/OS 91, 209
document 136
dollar sign 32, 143
DORMANT 187
double asterisk 96, 160
DOWN 82
DROP 146
DSACTIVE 220
DSB 143
DSN 20, 38, 42, 51, 61, 105, 120, 135, 136, 142, 143, 155, 157
DSNAME 25, 75, 76, 143, 146, 152, 171
DSNBUPDT 210
DSNs 219
DSS 54, 75, 77, 103, 111, 125, 165
DSSPRINT 125
DSTEST 194
DUMMY 38, 42
DUMP 125, 221
dynamic
exit 122
Dynamic Exit Facility 54
DYNMAXRC 56
DYNPGM 56

E

ECSA 123
edit 17, 29
enable
RESTBARS
ARSKEXT 206
END 82
ENTER 86
entries 35, 112, 139
environment 29, 50
equal sign 56
ERROPTS 156
error 40, 123
ERRORC 125
ERRORPT 153
ESDS 15
ESP 34, 40
ESPJOB 40, 42
ESPLIST 42
ESPREAD 40
ESPRHIST 34
evaluate 85
Event1 100
Event2 100
Excel 189
exception 62, 69
EXCL 100, 127
exclamation mark 160
EXCLFILE 153
EXCLJOB 47

EXCLSTAC 96
EXCLUCAT 125
exclude 96, 100
EXCLUDED 153
EXCLUDES 47
excluding data sets and jobs from processing 46
exclusion
    list 156
EXCPFILE 153
EXEC 19, 42, 67, 78, 119, 122, 124, 127, 166, 172
EXECUTE 45, 161
Execute Utilities 157
execution 12, 19, 104, 125, 127
exit
    XRESTJCL 201
exit member 163
exits 37, 42, 48, 75, 118
EXPDT 95
extensions xxi
extract file 251
extraction 53, 68
extractor 47, 48, 53, 56, 60

F

facility 53
FACILITY 20, 21
Fast Dump Restore, *See* FDR
FAVER 89, 199, 208, 233
FDF 125
FDR 75, 111, 118, 119, 122, 125, 127
FDR/ABR 88, 127, 207
FDRDSF 118, 119
FDRFULL 127
FDRTYPE 127
FDRZAPOP 118
field 31, 56, 129
figure 32, 62, 73, 78, 85, 111, 139
file
    analysis 69
FILTER 113
filters 113
FIND 29
FIREME 202
FIRST JOB? 31
FIXLIST 140, 142, 143
flag 156
FLAGS 156
FLATFILE 77
flowchart 73
FORCE 22
FORCE.STC 22
FORCE.STC.mbrname.id 22
forecast 37
format 129
forward cascade 219
full volume 125, 127, 128
FULLDSS 125, 128, 129
FULLFDR 127–129
function 123

G

GDG 11, 46, 74, 77, 98, 143, 159, 181
generate
    output report 173
    reports 187
    RESTJCL 201
generated
    JCL 204
generating 12, 73
GLJOB 31
glossary 415
GOOD 202
graphs 189
grouping 159
groups 24, 82

H

hardware 20
hash 32, 96
header 134
HELP 82
high-level qualifier 114, 128, 131
HILEVEL 218
HISTDS 177, 183
HISTJOB 177, 183
    See also job history
historical 55, 139
historical data 215
history 34, 45, 62, 149
HISTORY 148
HISTRPT 187
HISTRYL2 11, 148, 229
HLQ
    .LIBRARY 216
    .STC.LIBRARY 216
    .USERMAC 40
    BKUPTBL 203
    RESTJCL
        ALLOCAT 203
    SBRNEXEC 173
HOSTID 67
hours back filter 217
how to use this book xxi
HSEND 86
HSM
    exit
        ARCSKEXT 206
HSMACT 229
HSMPARM 86

I

IBGENER 111
IBM documentation 24
IBM Service 12
IBMFAC 25
ICF 54
ICP 141, 142
IDCAMS
    output 202
    parameter list 202

identifier 22, 29, 32
IEBCOPY 1, 111, 113, 203
IEBGENER 1, 93, 111
IEFBR14 93
IEFU83 48, 53
IEFU84 48, 53
IEFU85 48, 53
IEFU86 53
IEFUJV 124
IF SUBSTR 42
image copies 161, 163, 164
implementation 29, 119
IMS 164
IMS database 164
INCFILE 171
INCL 100, 103
INCLDSN 100, 171, 177–179
include 75, 76
INCLUDE 74, 165
incremental
    processes 127
indicator 135
INFILE 46
INFO 26
information 8
Initialization 48
input
    control cards 165
    copy 247
inquiry 189
INQUIRY 125, 181, 183, 185
install xxi, 5
installation 5, 8, 10, 12, 17, 20, 26, 29, 47, 53, 118, 124, 172, 173
Installation Worksheet 5
InstaSnap 157
INTAPE 93, 95
integration 172
Interactive System Productivity Facility xxi, 50
interface
    installation
        customization 173
interfaces 29
intermediate 43
interval 73
Interval Control Program 141
INV$ 141
invalid condition 125
inventory 13, 62, 67, 69, 98, 105, 139–143
Inventory
    Update 346
INVENTRY 61, 139, 152, 157, 159
INVENTxxxx 177
INVLSRD 225
INVMAINT 139, 140, 142
INVR#### 141
INVREMOV 142
INVREPT 174
INVTENG 140
INVTNEG 139
INVTUNLD 139
INVTYRPT 187
IRXJCL 38
ISMF 81
ISPF
    environment 188, 199
    Inquiry 149
    inquiry panels 177
    panels 187, 188, 199
IXCMIAPU 20, 21, 24
IZBR 46
IZBR Profile 5
IZWS 29, 43

J

JCL
    assemble/link 247
    CA7JOB, modifying 36
JCLLIB 13, 14, 19, 73, 85, 89, 105, 106
JES2 21, 54
JES3 21
JESMSGLG 221
JGENS 151
JLLIB 12
job
    CRITREMV 197
    exclusion 149
    mask 32
    name 31, 38, 112, 119, 124, 129
    scheduler 34
    steps 123
    validation exit 119
Job Edit 31
Job Initiation Exit 215
    See also IEFUJI
job record viewer
    TimeLiner 217
jobcard 17
JOBCARD 12
JOBCOUNT 166
JOBDSN 105
JOBINFO 113
JOBMASK 32
jobname 22, 42, 85, 86, 105, 106, 134, 164
JOBNAME 32, 37, 43, 47, 217
JOBRPT 187
jobs
    exclude from collection 46
JOBS/DSN 194
journal 68
JOURNALLOGSTREAM 26
JS10 40
JS20 40
JS30 40
JS40 40

K

KEY 22
KEYDEFN 149, 150
KEYDHIST 149, 150, 225
KEYDSN 151, 152
keyed
    history maintenance 343
Keyed History 149–151
KEYJHIST 149, 150

KEYLOAD 150
KEYMNTD 150
keyword 56, 68
KSDS 13, 14

L

legacy
    program 328
LEGACY 111
LEGFILE 123, 125
LEGTABLE 111–113
libraries 10, 12, 17, 20, 229
library 5, 16, 19, 20, 42, 53, 86, 88, 123, 124
LIBRARY 38, 61, 118, 134, 140
line commands 217
line format 129
linear 20, 22, 26, 215, 216
LINKLIST 86, 207
list 76, 96
LIST 24
LISTALL 74
LISTCUR 74
listing 122
LISTTAB 74
load 20, 80
LOAD 118
load libraries 216
log stream 20–22, 24, 215
logical dump 119
LOGR
    support 254
logs 86
LOGSTRM 24, 25
LPAR 1, 48, 68
LPARS
    running 20
LRAGE 143
LSNAME 254

M

macro 120
MAIN1351 221
maintain
    plan 262
MAINTAIN 148
MAINTCRT 152, 156
maintenance
    Bypass Report File 148
    processing 364
MAINTENANCE 29
MAINTLST 152, 153, 156, 157
MAINTLSTP 155
MAINTRPT 155
MAN 122
MAN files 119
management
    commitment 257
manager 123
MANTLSTN
    sample report 156
MANTLSTP 155
MANTREMV 156
manuals xxi
MAP 221
mask 31, 46, 47, 75, 77, 96
masking 129
MAXFILES 166
MAXMEG 96
MAXQD 53
MAXSTAC 96
mbrname 22
member
    $DELETE$ 204, 205
MEMBER 42
members
    member name 34
MEMPLATE 142, 159, 160
merging
    records 152
message
    error 263
    identifiers 277
    informational 263
    warning 263
messages
    BRNVB020 343
    BRNVB031 343
    BRNVB032
        (TABLEVFY) 344
    BRNVB034 345
    BRNVB044 346
    BRNVBCAT8 346
    BRNVBDMS 350
    BRNVBDSS 350
    BRNVBFDR 352
    BRNVBFVR 356
    BRNVLEGX 359
    cycle management 279
    error 247, 282, 291, 352
    HISMAINT 364
    in-flight processing 280
    information 247
    informational 282, 291, 352
    INVMAINT 282
    INVSPLT 291
    READHSM 292
    RTS 365
    SCANSMF 328
    simulated recovery 334
    SMFDSSU 336
    VFIDYNEX 356
    VFIGENR 360
    warning 247, 282, 352
methodology 12, 17, 73–75, 80, 88
MGMTCLAS 95
MIGJOBB
    options 165
migration 75
mirror
    volume 134
MIRVOL 134, 136
MIRVOLS 134
mode 5
MODE
    SMART

MODE *(continued)*
SMART *(continued)*
DUMB 202
modify
RESTALC 203
MODIFY 22, 29, 38
modifying
CA7JOB 36
modules 22, 42, 122
monitor
application jobs 216
monitoring 111, 114, 118, 119
move
critical data sets 168
MOVEJCL 168
movement method 168
MSG 143
MSGLEVEL 26
MSGLVL 53, 206
multiple 20, 38, 40, 48
MVS 48, 156
MVSA 253
MVSADMIN.LOGR 20, 21
MVSB 253
MYLIB 56

N

NAME 80
names 46
NEGFILE 139–141
new program 112
NEWENTRY 153
NEWPGM 113
newvalu 60
NOALLOC 11, 17
NOBACKUP 136
NOCAT 156
NOCOMMENT 113
node 46, 75, 96
non-zero 125
NONCRITA 157
NONCRITD 157
NONCRITS 157
NONE 26
NONVSAM 76
NOSCRATCH 202
NOTFOUND 202
NOTONVOL 202

O

OBJECTDSN 26
ODTCKPT 68
offsite rotation 163
OLDAPPS 153
online
accessing xxi
publications xxi
operation 75, 134
operational recovery 199
Operational Status Display 12
option
modules 118
NOSCRATCH 202
Option 100
OPTIONAL 68
options 113, 120, 122
order 135
OTHER 202
OUTDSN 48, 56, 253
output
exception
administrative 161
files 164
multiple 164
reports 161
OUTPUT 113
OUTTAPE 93, 95
OVERLAY 95
overview 1

P

parallel 129
parameter 20, 26, 45, 67, 69, 76, 86, 106, 113, 128, 141, 142, 151
parameters 8, 10, 12, 43, 53, 56, 68, 74, 98–100, 105, 113, 120, 124, 127, 129, 142, 147, 148, 152, 153, 165, 225
PARM 38, 42, 67, 78, 86, 111, 113, 123, 133, 135, 151, 155, 188, 196
PARMLIB 21, 22, 27, 48, 54, 97, 123, 124, 134, 159, 254
pattern
field 173
PCT 142
PDF xxi
PDS 27, 29, 34, 38, 42, 75, 78, 79, 86, 128, 129, 134, 165, 168, 194, 202
PDSE 32
percent 31, 75, 142, 143
PERMIT 24
PGM 42, 49, 67, 88, 89, 95, 97, 118, 122, 124, 131, 166, 172
physical dump 119, 122
plex 22
POPTIONS 225
PPT 22, 123, 142
PPT PGMNAME 22
pre-allocated 122
preface xxi
preferred volumes 168
primary
panel 177
primary data
corruption
destruction 220
problem determination, IBM Support
determining business impact xxii
PROC 73, 85, 86
procedure 93
process
control tasks 142
processing 142
PROCLIB 19
PROCNAME 25
PROCVFI 19

PROD 46, 142
PROD01 134
PRODAP01 77
PRODAP02 77
PRODAP03 77
PRODAP04 77
PRODAP05 77
PRODAP06 77
PRODAP07 77
production 10, 77, 127, 139
profile 24
program
    processing tasks 142
Program Development Facility xxi
Program Properties Table 22, 123
PROGxx 21
project plan 258
protect xxi
PS010 125, 127
PS020 125, 127
PS030 125, 127
PS05 93
PS050 127
PS10 93, 95
PS20 93, 95
PS30 93
publications
    accessing online xxi
Pullist 233

Q

QNAMES 20
qualifiers 46
question mark 31, 32, 96, 129, 160
QUIESCE 222
    See also STOP

R

R/T 140
RACF 20, 24, 140
RACLIST 20, 24
RDEFINE 24
reader
    date
        start 151
READER 105
real-time
    command queue 13, 16
    job 159
    SMF 159
    SMF extraction 251
Real-Time Dynamic Dataset Scheduler 365
REBUILD 104
RECFM 38
RECONLST 229
RECONOUT 229
record
    types 49, 53
recover
    backups 210
    corrupted backup table 172
recover *(continued)*
    zDP snapsets 218
recovery
    environment 207, 209
    job flow 201
recovery option 219
recreate
    backup table entries 172
rectype 143
reference xxi
refresh 67
REFRESH 111
REINIT 224
rejected 103
reload
    application table 223
    backup table 172
    SVC
        SMF exit 224
RELOAD 123, 221
RELOAD COMMAND 223
removal 46, 47
remove
    critical entries 171
    obsolete records 156, 157
REMOVE 46, 142, 143
removing data sets and jobs from processing 46
rename 17
report
    critical file 155
    cycle management 196
    dormant 197
    FLAGS 152
    Inventory records 195
REPORT 73, 125
REPORT DD 356
reporting
    non-critical data sets 157
reports
    delete 188
    print 188
    rename 188
    view 188
REPORTS
    $INDEX 188
request 24, 143
requirements 20, 21, 26
reschedule 100, 103
resolve
    application assignments 175
resource 24
resource permits 25
restart 139, 142
RESTBARS
    parameters 206
    requirements 207
RESTJCL 199, 200, 203, 204, 207, 209–212
restore
    all critical data sets 200
    CA Dynam/TLMS 210
    critical data sets 199, 212
    DASD 204, 207, 209
    data sets 210–212
    interface messages 263

restore *(continued)*
JCL generator 199
online DASD 205
RESTTBL 203
snapsets 218
specific applications
specific data sets 201
specific data sets 199
table 207
updates 205
VSAM 208
RESTORE 61, 125
restructured xxi
RESTTBL 205, 207, 229
retention period 146, 148, 149
retesting 262
RETPD 95
RETPD1 35, 148, 156, 225
RETPD2 148
RETPDI 45, 225
retrieval 125
return code
4 282
8 282, 291
table
alternate 247
reverse cascade 219
REXX
EXEC 21
risk analysis
perform 258
RMM
database 212
RMMBKUP 212
RMMVER 212
routine 124
RPTGENS 147, 188, 225
RPTMAINT 147
RPTRETPD 147, 188
RRDS 16
RT/ICP 142
RT/SMF 47–49, 54, 61, 62, 67, 68, 97, 103, 104, 131, 133, 142, 149, 157, 344
RT/SMF Dispatcher 114
RT/VRS 133, 134
RTA 111
RTACKTP 229
RTCMDB 67, 133
RTCMDQ 16, 134, 229
RTDICPIN 140–142
RTDPRINT 142
RTINVPRM 140–142
rtpd 129
RTPRINT 103
RTSC039 142
RTSMF 252, 253, 328
RTSSPPRM 114
rules 22, 25
run
activation 135
MAINTLST 157
name 135
period 135
run-time 20, 134, 140
RUNHIST 40

S

SAF 25
Safeguarded Copy 220
SBJC.V420.MENU 135
SBRNUMAC 5
scan 111
SCANALC 111
SCANSMF 111, 113, 114, 119
SCANSMF process 328
SCHDALLC 97
SCHDJDS 97
SCHDPARM 97, 98
SCHDREQ 99, 100, 103
SCHDTBL 99
schedule 45, 98, 113, 139–141
scheduler 29, 35, 62, 99, 104, 106
scheduling 38, 40, 42, 73, 97, 107
SCHEDxx 22
SCHEDXX 123
SCHELIGR 97
SCHPENDR 97
SCOPE 20
scratch 11
SCRATCH 68, 202
SDSF 21
SDUMP 53
security 22, 24, 25, 118
SELECT
DD 195
selection criteria
deletion 151, 152
sequence 43
serial
number 26
SET 221
SETROPTS 24
SETSYS EXITON 207
SEVERE 26
severity
contacting IBM Support xxii
determining business impact xxii
SIM$ 135
SIM$IN 135
SIMLOGF 133
SIMLPRI 131, 133
SIMLSEC 131, 133
SIMNAME 131
SIMPFX 131, 135
SIMRCVRY 133, 135, 225
SIMTBKUP 133
SIMTSNAP 131
SIMULATE 143
Simulate mode 171
simulation 136
SIMVOL 131
SMF
analysis 148
audit data 149
CNTL
SMFP 215
data collection 215

SMF *(continued)*
exits 215
real-time 218
record type 193, 215
system
ID 215
VDG definitions 215
SMF data
exclude from collection 46
SMF extractor
running multiple 252, 253
SMF.MANx 48, 251
SMF$ 47, 53, 68, 159
SMF30PGM 113
SMFDATA 55
SMFDSN 48, 68
SMFDSSU 233
SMFDSSUS 119, 122
SMFDSSUT 122
SMFEXT 53, 60
SMFEXTP 47, 48
SMFP 48
SMFPRMxx 21, 48, 54, 124
SMFREAD 252, 253
SMFREC 56
SMFSORT 55
SMFWAIT 252
SMFXU83 53
SMODE 105
SMS 20
snap 129
snapset
metadata 215
software 20, 29
SORT 111, 203
sorting 55
source 22
space 20, 26, 56
special character 32
specifying 31
spreadsheet 189
SRFD0504 129
SSPCKPT 114
stack 129
stacking 96
STACVOL 96
STACVOLS 96
start
task 118
START
TASK 97, 141
started task 22, 25, 53, 62, 119, 123, 139, 140, 149
startup 98
STARTUP 67, 133
STARTVRS 133, 134
STATCHNG 153
statement 103, 142
status 133, 134
STC 21, 22, 27
step name 119, 124
STEPLIB 50, 86, 123, 207
STEPNAME 217
stop 134
STOP 22, 123, 222
STOPVRS 133
storage
group 168
STORCLAS 95
SUBAPPL 95
SUBFILE 105
SUBFILE2 105, 106
SUBMIT 45
subset 80
SUBSTR 37
SUBSYS 48, 54
subsystem 21, 40, 161
subtask 49, 50, 53, 61, 62, 69, 97, 103, 131, 133, 140
Support
contacting IBM xxii
describing problems xxii
determining business impact xxii
submitting problems xxii
SVC 53, 215, 222
SVC DUMP 206
SVCDUMP 206
SVSCPLEX 254
SWITCH 221
synchronization
point 262
syntax 76, 96, 99, 143
SYS 48, 124
SYS.IEFUJV 119
SYS1.MANx 56
SYSDA 10
SYSEXEC 19
SYSID 35, 48, 68
SYSIN 43, 89
SYSIND 95
SYSMVSB 253
SYSOUT 38, 42, 89, 106, 107, 118, 129, 134, 135, 166, 168, 171
SYSPLEX 215, 222
SYSPLEX.TIMELINR.JOURNAL 24
SYSPRINT 122, 127, 139
SYSPROC 19
SYSTABLESPACE 161
system
catalog 153
data 259
exit 119
ID 51, 53, 67
logger 251, 254
maintenance 29, 222
requirements 5
SYSTEM 60
System Administrator xxi
System Operators xxi
SYSTSPRT 106, 107, 129, 168
SYSUID 14

T

table
BKUPMNT 146
RPTMAINT 148
verification processing 344
tablespaces 161, 163
tape

tape *(continued)*
copy 75
copy utility 164
drive requirements 262
files 259
management 93
management catalog 201
management system 146, 163
restores 201
Tape Management System Vault Job 69
TAPE1 125, 127
TAPECOPY 153
TAPECPY 95, 210, 211, 247
TAPEPARS 93, 95
task
initial implementation 216
TCALTMSG 247
TCALTTBL 247
TCDFLT 247
TCMSG 247
TECH 38
Technical Support 34, 37
template 10, 89
TEMPLATE
MIGBKUP 165
terms defined 415
test
strategy 262
string 13
TEST 46, 78
testing
process 262
time
sharing xxi
TimeLiner
address space 223
commands 221
job record viewer 217
messages 221
starting 216
stopping 222
using 215
TLINER 222
TLN 50
TLN013I 221
TLN266I 221, 223
TLN289I 222
TLNCTL00 22
TMS
database 210
environment 210
TMSAPI 146
TMSUPC01 211
token services 119
TPUT 86
TRACE 53, 206
tracking 47, 122, 149
transaction 26
TRANSACTIONDUMPINDEX 26
TSO 19, 29, 50, 62, 86, 133, 161, 177
TSO EXEC 21
TSOSUB 105
TSS 25
TWS 43
TWS/OPS 34
TWSPROC 43
type 202 118, 119
typeface conventions xxiii
TYPERUN 206
TYPRUN=NORUN 125

U

U0888 120
UCB 134, 168
UFI 26, 50, 51
UFIFILE 26, 51
UMODE 105
uncatalog
data sets 202
UNCATMSG 168
unload 139
update
critical file list 152
UPDATE 22, 24
UPDATES 74
updating 25, 27
USECOUNTS 222
user
options 122
user ID 22, 25
USERCAT 125
USERID 24, 35
USERIN 85, 86
USERMAC
CRITRPTR 194
LIBRARY 196
member description 233
SCHDREQ 99
VFIPARMS 188, 225
XJOBRPTR 193
XRESTALC 203
XRESTJCL 201
YDELETE 202
USERPARM 127
users 24
using 31
utilities
CA7JOB, modifying 36
utility
function 34

V

valid 136
validate 125
validation 124
Value 56, 120
VDG group 215
VDUOPT 122
VDUPRINT 118, 122
VFIDCOPT 195
VFIDYNEX 123, 124
VFIFDSS 125
VFIIDR 127
VFIIFDR 127
VFIMAINT 148, 229

VFIPARMS 149
VFISCHED 104
VFISUB 105
VFISUBR 106
VFIUJV 124
view
  backup table 179
  critical file list 178
  cross dependencies 195
  databases
    reports 177
  history
    report 194
  history information 183
  Inventory database 181
  jobs report 193
Virtual Recovery Simulator 62, 131, 133–136
virtual storage 24
VOLSER 14, 16, 134
volume
  production
    work 168
VOLUPDT 210, 211
VRS 131, 133–136
VRSALLOC 131
VRSCKPT 131, 133
VRSMIROR 134
VSAM
  Keyed History 183
VSXMAI 47

W

WAIT 86
WARM 98
WARN 26
what's new xxiv
wildcards 31, 32, 34, 47, 142

X

XABARBKP 81, 86
XBACKMN2 146
XBACKMNT 146
XBKUP001 75
XBKUPJCL 75
XDMSBKUP 91
XDSSBKUP 80
XFDRBKUP 88
XFVRBKUP 89
XLEGACY 113
XTAPECPY 93, 96

Y

YABARS 205
YDELETE 202
YDMSREST 209
YDSSREST 204
YFAVER 208
YTAPECPY 212

Z

Z Systems 50, 51
z/OS 48
ZARAVOLR 212
zDMF
  DSNLISTX 173
zDP
  processes 215
  Restore 215
  snapsets 51
ZDPAUDIT 51, 216
ZDPMAINT 157
ZDR 119
ZDRDSF 119
ZEKE 34, 45
ZEKERPT 45

GC27-9554-01