IBM

z/OS Version 1 Release 3 and 4 Implementation

z/OS, z/OS.e, msys for Operations, JES2, JES3, WLM

UNIX System Services, zSeries File System, Parallel Sysplex

RACF, PKI, LDAP, Crypto, CF duplexing

Archived

Paul Rogers
Luiz Fadel
Erik Frederiksen
Hirokazu Kakurai
Frank Kyne
Franco Meroni
Henrik Thorsen
Yolanda Vivar

Redbooks

ibm.com/redbooks

Archived

International Technical Support Organization

z/OS Version 1 Release 3 and 4 Implementation

June 2003

Archived

SG24-6581-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xiii.

Archived

First Edition (June 2003)

This edition applies to Version 1 Release 3 and Release 4 of z/OS (5694-A01), and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2003. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices . . . xiii
Trademarks . . . xiv

Preface . . . xv
The team that wrote this redbook . . . xv
Become a published author . . . xvii
Comments welcome . . . xvii

Chapter 1. z/OS Version 1 Release 3 and 4 overview . . . 1
1.1 z/OS Version 1 Release 3 and Release 4 BCP . . . 2
1.1.1 IBM z/OS service policy . . . 2
1.1.2 Elements and features included in z/OS V1R3 . . . 2
1.1.3 Elements and features removed from z/OS V1R3 . . . 2
1.1.4 Elements and features removed from z/OS V1R4 . . . 2
1.2 z/OS V1R3 base control program (BCP) . . . 3
1.2.1 64-bit architecture . . . 3
1.2.2 WLM compatibility mode . . . 3
1.2.3 Access control lists (ACLs) . . . 4
1.2.4 Automount facility . . . 4
1.2.5 TSO/E dynamic broadcast data set . . . 4
1.2.6 Change to parmlib . . . 5
1.2.7 System Logger attributes . . . 5
1.2.8 ASID resource relief and monitoring . . . 6
1.2.9 Page data set protection . . . 7
1.3 z/OS V1R4 base control program (BCP) . . . 7
1.3.1 System symbols . . . 7
1.3.2 Automatic restart management . . . 7
1.3.3 Changes to parmlib . . . 8
1.4 Workload Manager . . . 8
1.4.1 WLM enhancements in z/OS V1R3 at a glance . . . 8
1.4.2 WLM enhancements in z/OS V1R4 at a glance . . . 8
1.4.3 Multilevel security . . . 9
1.5 UNIX System Services (USS) . . . 9
1.5.1 Shared HFS support for z/OS V1R4 . . . 10
1.5.2 Syslist support for shared HFS . . . 10
1.6 Distributed File Service (DFS) . . . 10
1.7 msys for Setup . . . 11
1.7.1 z/OS V1R4 components . . . 11
1.8 JES2 Version 1 Release 4 . . . 12
1.9 JES3 Version 1 Release 4 . . . 12
1.10 System Display and Search Facility . . . 13
1.10.1 MEMLIMIT column on SDSF DA Display . . . 13
1.11 Communication server . . . 13
1.11.1 Sysplex-wide dynamic VIPA . . . 14
1.12 Firewall Technologies . . . 14
1.13 LDAP Server . . . 14
1.14 Network Authentication Service . . . 14
1.15 Open Cryptographic Enhanced Plug-ins (OCEP) . . . 15
1.16 RACF . . . 15

Archived

1.17 Enterprise Identity Mapping (EIM) . . . 15
1.18 z/OS.e . . . 15
1.18.1 Installing z/OS.e . . . 16
1.18.2 z/OS and z/OS.e differences . . . 16
1.18.3 Differences in element, features, and functions to z/OS . . . 18
1.18.4 New e-business workloads supported by z/OS.e (and z/OS) . . . 19
1.18.5 Required parmlib customization for z/OS.e . . . 19
1.18.6 LPAR update . . . 19
1.18.7 z/OS.e Web site . . . 19

Chapter 2. Installation of z/OS Version 1 Release 3 and 4 . . . 21
2.1 Processor requirements . . . 22
2.1.1 z.OS.e and z.OS release cycle synchronization . . . 23
2.1.2 z/OS.e V1R3 and V1R4 MSU capacity . . . 23
2.2 DASD space requirements . . . 23
2.3 ServerPac installation changes for z/OS V1R4 . . . 25
2.3.1 RESTFS job . . . 25
2.3.2 Select JES at install . . . 26
2.3.3 Master catalog flag . . . 26
2.3.4 Restructured ALLOCDS job . . . 26
2.3.5 msys for Operations support . . . 26
2.4 Coexistence, migration, fallback policy . . . 27
2.4.1 z/OS V1R4 coexistence . . . 27
2.4.2 Rolling IPLs . . . 28
2.4.3 Fallback . . . 28
2.4.4 JES3 coexistence . . . 28
2.4.5 JES2 coexistence . . . 29
2.4.6 msys for Setup coexistence . . . 30

Chapter 3. Base control program (BCP) . . . 31
3.1 z/OS V1R3 BCP enhancements . . . 32
3.1.1 TSO/E broadcast data set . . . 32
3.1.2 ASID resource relief and monitoring . . . 35
3.1.3 Page data set protection . . . 36
3.1.4 Binder changes . . . 41

Chapter 4. msys for Operations enhancements . . . 43
4.1 msys for Operations overview . . . 44
4.1.1 Foreground msys for Operations command dialogs . . . 45
4.1.2 Background msys for Operations automated recovery routines . . . 46
4.1.3 msys for Operations business value . . . 46
4.2 msys for Operations implementation checklist . . . 47

Chapter 5. JES3 Version 1 Release 4 enhancements . . . 49
5.1 JES3 enhancements . . . 50
5.1.1 JES3 MAINPROC refresh . . . 50
5.1.2 New Inquiry command . . . 55
5.1.3 Main processor status enforcement . . . 58
5.1.4 JES3 checkpoint protection . . . 61
5.2 JES3 BCP compatibility . . . 63
5.2.1 JES3 coexistence maintenance . . . 63

Chapter 6. JES2 Version 1 Release 4 enhancements . . . 65
6.1 JES2 health monitor . . . 66

Archived

6.1.1 JES2MON address space . . . 66
6.1.2 Sampler processing . . . 67
6.1.3 Probe processing . . . 68
6.1.4 Command processing . . . 71
6.2 End of memory . . . 80
6.2.1 TSO/E multiple logons . . . 80
6.3 Checkpoint data corruption . . . 81
6.4 HAM I/O improvements . . . 81
6.5 Enhanced INCLUDE statement externals . . . 81
6.5.1 Enhancement to INCLUDE statement . . . 82
6.5.2 Using default PARMLIB . . . 83
6.6 XMIT JCL card externals . . . 85
6.7 SDSF enhancements . . . 85

Chapter 7. z/OS Workload Manager (WLM) . . . 87
7.1 Removal of WLM compatibility mode . . . 88
7.1.1 Sample service definition with z/OS 1.3 . . . 88
7.1.2 Service policy description . . . 90
7.1.3 IPLing z/OS V1R3 . . . 91
7.1.4 Sample service definition IWMSSDEF . . . 92
7.1.5 WLM install definition utility . . . 94
7.2 PAV dynamic alias management for paging devices . . . 99
7.2.1 Globally enabling WLM I/O priority management . . . 100
7.2.2 Individually enabling dynamic alias management on a per volume basis . . . 101
7.2.3 Enabling dynamic alias management for paging devices . . . 101
7.3 WLM independent enclave service class reset . . . 101
7.3.1 Enclaves . . . 102
7.3.2 Resetting independent enclaves . . . 103
7.3.3 Multisystem enclave support . . . 104
7.4 WLM support for sub-capacity pricing . . . 104
7.5 WLM enqueue management enhancements . . . 105
7.5.1 ERV parameter in IEAOPTxx . . . 105
7.6 WLM WebSphere performance enhancement . . . 107
7.7 WLM batch initiator balancing enhancements . . . 107
7.7.1 WLM batch initiator management . . . 108
7.7.2 Queueing jobs for execution . . . 108
7.7.3 Classifying jobs . . . 109
7.7.4 Service classification rules . . . 110
7.7.5 Steps required for batch initiator management . . . 111
7.7.6 Batch initiator management limitations . . . 112
7.7.7 z/OS V1R4 enhancements . . . 112
7.8 Performance block application state reporting for enclaves . . . 113
7.8.1 RMF support . . . 113
7.8.2 New classification qualifiers for WebSphere (CB) subsystem . . . 116
7.9 WLM msys for Setup enhancement . . . 116
7.10 WLM support for ESS FICON and I/O priority management . . . 116
7.11 WLM temporal affinity for WebSphere Application Server . . . 117
7.11.1 SDSF support . . . 117
7.11.2 RMF support . . . 118
7.11.3 Vary command . . . 118
7.12 WLM velocity goals . . . 118
7.13 Enterprise workload management: eWLM . . . 119

Archived

Chapter 8. UNIX System Services enhancements in z/OS V1R3 121
8.1 Access control list (ACL) support for V1R3 122
8.1.1 File access authorization checking 122
8.1.2 New UNIXPRIV profiles with z/OS V1R3 123
8.1.3 ACL overview 125
8.1.4 Security product and ACLs 126
8.1.5 Creating ACLs 127
8.1.6 Access checking with ACLs 128
8.1.7 File and directory access with ACLs 130
8.1.8 ACL inheritance 131
8.1.9 Defining ACLs from OMVS 132
8.1.10 Using ACLs from the ISHELL 136
8.1.11 Create an ACL using the ISHELL 138
8.1.12 Example of ACL inheritance 138
8.1.13 Other setfacl command options 141
8.1.14 Modified commands with ACL support 142
8.1.15 USS Logical File System ACLs support 144
8.1.16 z/OS UNIX REXX support for ACLs 145
8.1.17 LE Callable Services support for ACLs 146
8.2 ISHELL enhancements 148
8.2.1 Directory list enhancements 148
8.2.2 Using the cursor on the directory list panel 150
8.2.3 Displaying colors on the Directory List panel 154
8.3 Shutting down z/OS UNIX without re-IPLing 154
8.3.1 Registration support 155
8.3.2 Shutting down z/OS UNIX 156
8.3.3 Restarting z/OS UNIX 159
8.4 Automount enhancements 160
8.4.1 Display current automount policy 160
8.4.2 Support “#” as comment delimiter in map file 160
8.4.3 Dynamic HFS allocation in automount 161
8.4.4 Generic match on lowercase names 162
8.4.5 Support system symbols in map file 164
8.5 Copytree utility 165
8.6 Shared HFS unmount option 165
8.6.1 New UNMOUNT option 166
8.7 Mount table limit monitoring 168
8.7.1 Shared HFS support for confighfs command 172

Chapter 9. UNIX System Services enhancements in z/OS V1R4 173
9.1 Automove system list 174
9.1.1 Automove system list specification 174
9.1.2 Changing an automove system list 177
9.2 Byte-range locking in a shared HFS environment 178
9.3 Shared HFS availability enhancement 179
9.4 Enhancements for UID/GID support 179
9.4.1 RACF database and AIM 180
9.4.2 Search enhancements to map UIDs and GIDs 181
9.4.3 Shared UID prevention 182
9.4.4 Automatic UID/GID assignment 183
9.4.5 Group ownership option 187

Chapter 10. zFS file system enhancements 189
10.1 zFS file systems 190

10.1.1 zFS supports z/OS UNIX ACLs. 190
10.2 Dynamic configuration. 191
10.2.1 New zfsadm commands 191
10.3 Dynamic aggregate extension. 195
10.3.1 Implementing dynamic aggregate extension. 195
10.3.2 Dynamic file system quota increase 196
10.3.3 Displaying dynamic aggregate and quota extensions. 197
10.4 New -grow option 198
10.4.1 Formatting aggregates with -grow. 199
10.5 Duplicate file system names 201
10.6 System symbols in the IOEFSPRM file. 202
10.7 Metadata backing cache and log file cache 202
10.7.1 Metadata cache storage 203
10.7.2 Log file cache 203

Chapter 11. Security Server RACF enhancements 205
11.1 Security Server components 206
11.2 RACF enhancements in z/OS V1R3 206
11.2.1 Access control lists (ACLs) for UNIX System Services. 206
11.2.2 Policy Director Authorization Services for z/OS and OS/390 support. 207
11.3 Security Server RACF enhancement in z/OS V1R4 209
11.3.1 Enterprise Identity Mapping Services (EIM) 209
11.3.2 z/OS UNIX Security Management Usability enhancements 212
11.3.3 Program control and program access to data sets (PADS) 215

Chapter 12. Security Server PKI Services 219
12.1 Security Server PKI Services in z/OS V1R3 220
12.1.1 New component of z/OS Security Server 220
12.1.2 Digital certificate 221
12.1.3 Certificate life cycle 221
12.1.4 Browser certificates. 222
12.1.5 Server certificates 223
12.1.6 z/OS PKI Services architecture. 224
12.1.7 Prerequisite products 225
12.2 Security Server PKI Services enhancement in z/OS V1R4. 226
12.2.1 Sysplex support 226
12.2.2 Event notification via e-mail 227
12.2.3 Additional distinguished name (DN) qualifier support 228
12.2.4 LDAP password encryption. 229
12.2.5 PKCS#7 certificate chain support 229
12.2.6 Key generation via PCICC 230
12.2.7 Additional default CERTAUTH 230

Chapter 13. Security Server LDAP server. 231
13.1 Security Server LDAP Server enhancements in z/OS V1R4 232
13.1.1 DIGEST-MD5 and CRAM-MD5 authenticate support. 232
13.1.2 Transport layer security (TLS) support 233
13.1.3 IBM-entryuuid support. 233
13.1.4 Modify DN operation 234
13.1.5 Access control list (ACL) enhancement 235
13.1.6 Activity logging 235
13.1.7 RDBM and JNDI removal 236

Chapter 14. Security Server Network Authentication Service 237

14.1 Security Server Network Authentication Service . . . 238
14.2 Enhancements in z/OS V1R4 . . . 238
14.2.1 IPv6 support . . . 238
14.2.2 New support for Kerberos registry in NDBM . . . 239

Chapter 15. Cryptographic services . . . 241
15.1 Cryptographic services components . . . 242
15.2 ICSF enhancements in z/OS V1R3 . . . 242
15.2.1 TSO panel enhancement . . . 242
15.2.2 Unique key per transaction (UKPT) and PKCS#1V2 support . . . 242
15.2.3 Advanced Encryption Standard (AES) support . . . 243
15.2.4 ICSF setup and CSFEUTIL utility enhancements . . . 243
15.2.5 Hardware requirements . . . 243
15.3 System SSL enhancements in z/OS V1R4 . . . 243
15.3.1 System SSL gskkyman utility . . . 244
15.3.2 Certificates with private keys in ICSF . . . 245
15.3.3 Shared SAF key rings . . . 245
15.3.4 Advanced Encryption Standard (AES) support . . . 245
15.3.5 IPv6 Network Address support . . . 246
15.3.6 Performance enhancements . . . 246
15.3.7 Session ID caching across a sysplex . . . 246
15.3.8 Serviceability enhancements . . . 247

Chapter 16. Parallel Sysplex enhancements . . . 249
16.1 Parallel Sysplex enhancements for z/OS V1R3 . . . 251
16.1.1 System Logger enhanced logstream attribute support . . . 251
16.1.2 GRS enhancements . . . 259
16.1.3 Sysplex-wide confighfs command scope . . . 261
16.1.4 Sysplex mount table limit monitoring . . . 261
16.1.5 Sysplex mount/unmount performance improvements . . . 261
16.1.6 DFSMShsm™ common recall queue (CRQ) . . . 262
16.1.7 Caching of larger than 4 KB CIs in VSAM RLS cache CF structure . . . 264
16.1.8 VSAM RLS lock structure duplexing enhancement . . . 266
16.1.9 DFSMS enforced data set separation for high availability . . . 267
16.1.10 OAM multiple object backup . . . 268
16.2 Parallel Sysplex enhancements for z/OS V1R4 . . . 271
16.2.1 XES DB2 Data sharing performance improvement . . . 271
16.2.2 XES CFRM performance enhancements . . . 271
16.2.3 RRS multisystem cascaded transaction enhancement . . . 273
16.2.4 System Logger offload monitor function . . . 274
16.3 Hardware and CFCC sysplex enhancements . . . 277
16.3.1 Cascaded FICON director switch support . . . 278
16.3.2 CF Request Time Ordering (Sysplex Timer connectivity to CFs) . . . 278
16.3.3 Enhanced Parallel Sysplex support in CFLEVEL 11 and CFLEVEL12 . . . 283
16.3.4 zSeries GDPS/PPRC hyperswap function . . . 284

Chapter 17. System-Managed CF Structure Duplexing . . . 289
17.1 Introduction . . . 290
17.1.1 System-Managed versus User-Managed Duplexing . . . 290
17.2 How System-Managed CF Structure Duplexing works . . . 292
17.2.1 Starting duplexing . . . 292
17.2.2 Maintaining a duplexed structure . . . 293
17.2.3 Stopping duplexing . . . 295
17.2.4 Error recovery . . . 296

17.3 Performance considerations . . . 297
17.3.1 CF CPU utilization . . . 298
17.3.2 Distance between CPC and CFs . . . 299
17.3.3 Effect of synch/asynch conversion . . . 299
17.3.4 Distance between CFs . . . 300
17.3.5 Link speeds . . . 301
17.3.6 CP speeds . . . 302
17.3.7 Estimating the impact of System-Managed CF Structure Duplexing . . . 302
17.3.8 Comparison to alternatives . . . 303
17.4 Capacity planning considerations . . . 304
17.4.1 Operating system CPC . . . 304
17.4.2 CF CPC capacity . . . 304
17.4.3 CF link capacity and subchannels . . . 307
17.5 Planning for System-Managed CF Structure Duplexing . . . 308
17.5.1 Hardware prerequisites . . . 308
17.5.2 Software prerequisites . . . 309
17.6 Setting up System-Managed CF Structure Duplexing . . . 311
17.6.1 Hardware changes . . . 311
17.6.2 HCD . . . 314
17.6.3 Operating system . . . 314
17.6.4 RMF . . . 314
17.6.5 CFRM changes . . . 315
17.6.6 LOGR CDS changes . . . 315
17.6.7 CF selection changes . . . 316
17.7 Exploiters of System-Managed CF Structure Duplexing . . . 317
17.7.1 CICS structures . . . 317
17.7.2 DB2 SCA . . . 318
17.7.3 IRLM . . . 318
17.7.4 IMS Shared Message Queue Structure . . . 319
17.7.5 IMS Shared Fast Path Database . . . 320
17.7.6 JES2 . . . 321
17.7.7 MQSeries . . . 322
17.7.8 MVS System Logger . . . 323
17.7.9 VTAM Generic Resources . . . 325
17.7.10 VTAM Multi Node Persistent Sessions . . . 325
17.7.11 WLM Multisystem Enclaves . . . 326
17.7.12 WLM Intelligent Resource Director LPAR Cluster structure . . . 327
17.7.13 VSAM/RLS Lock structure . . . 327
17.7.14 DFSMShsm Common Recall Queue . . . 328
17.8 Operations procedures . . . 329
17.8.1 Recovery . . . 329
17.8.2 Shutting down a CF . . . 330
17.8.3 Changes affecting duplexed structures . . . 330

Chapter 18. z/OS V1 DFSMS Transactional VSAM Services (DFSMStvs) . . . 331
18.1 DFSMStvs overview . . . 332
18.2 DFSMStvs . . . 332
18.3 Why DFSMStvs . . . 332
18.4 The batch window problem . . . 333
18.5 Recoverable data sets . . . 333
18.6 Non-recoverable data sets . . . 333
18.7 Transactional recovery . . . 334
18.8 VSAM RLS . . . 334

18.9 Parallel Sysplex CICS VSAM RLS . . . 335
18.10 Objective of DFSMStvs . . . 336
18.11 Accessing a data set with DFSMStvs . . . 338
18.12 Using DFSMStvs . . . 338
18.12.1 DFSMStvs logging . . . 339
18.12.2 Context and Unit of Recovery (UR) . . . 341
18.13 SYS1.PARMLIB changes . . . 342
18.13.1 DFSMStvs-related parameters . . . 343
18.14 Application considerations . . . 344

Appendix A. msys for Operations implementation checklist . . . 347
Step 1: Create VSAM and non-VSAM data sets . . . 348
Step 2: Copy additional PROCs into PROCLIB data set . . . 354
Step 3: Data sets for LNKLST and LPALST . . . 356
Step 4: Add a PPT entry . . . 358
Step 5: Update MPFLST . . . 359
Step 6: Define application major nodes for VTAM . . . 361
Step 7: Make determined security definition changes . . . 362
Step 8: Alter msys for Operations NVSS style sheet . . . 371
Step 9: Enable msys for Operations functions . . . 373
Step 10: Build the VTAM logon mode table AMODETAB . . . 383
Step 11: REXX environment table entries . . . 384
Step 12: Perform hardware customization on SEs . . . 387

Appendix B. Sample GRS exit ISGNQXIT . . . 401
B.1 Sample exit ISGNQXIT download . . . 402
B.1.1 Sample exit zip file . . . 402
B.2 Sample exit description . . . 403
B.2.1 GRS sample ISGNQXIT exit logic . . . 403
B.2.2 Scanning the RNL exclusion table examples . . . 404
B.2.3 RNL example . . . 406
B.2.4 Exit restrictions . . . 406
B.2.5 Recommendations . . . 407
B.2.6 Exit compatibility . . . 407
B.2.7 Exit installation and activation . . . 408
B.2.8 EXIT process verification . . . 408
B.2.9 Sample exit code . . . 410

Appendix C. Sample GRS exit ISGNQXITFAST . . . 421
C.1 Sample exit ISGNQXITFAST download . . . 422
C.1.1 Sample exit zip file . . . 422
C.2 Sample exit description . . . 423
C.2.1 GRS exit points . . . 423
C.2.2 GRS sample ISGNQXITFAST exit logic . . . 423
C.2.3 Scanning the RNL exclusion table examples . . . 424
C.2.4 RNL example . . . 426
C.2.5 Exit restrictions . . . 427
C.2.6 Recommendations . . . 428
C.2.7 Exit compatibility . . . 428
C.2.8 Exit installation and activation . . . 428
C.2.9 Installation exits ISGNGXITFAST and ISGNQXIT considerations . . . 429
C.2.10 EXIT process verification . . . 429
C.2.11 Sample exit code . . . 431

Related publications 441
IBM Redbooks 441
Other resources 441
Referenced and other relevant Web sites 442
How to get IBM Redbooks 444
IBM Redbooks collections. 444

Index 445

Archived

Archived

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.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

© Copyright IBM Corp. 2003. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX®
BatchPipes®
BookManager®
CICS®
Domino™
DB2 Universal Database™
DB2®
DFS™
DFSMSdfp™
DFSMShsm™
Encina®
Enterprise Storage Server®
eserver ™
FICON™
Geographically Dispersed Parallel Sysplex™
GDDM®
GDPS®
IBM®
IMS™
Infoprint®
iSeries™
Language Environment®
Lotus Notes®
Lotus®
Multiprise®
MQSeries®
MVS™
NetView®
Notes®
OS/390®
OS/400®
Parallel Sysplex®
pSeries™
QMF™
Redbooks™
Redbooks (logo)™
RACF®
RMF™
S/390 Parallel Enterprise Server™
S/390®
SecureWay®
Sysplex Timer®
SystemPac®
SLC™
SOM®
Tivoli®
VisualAge®
VM/ESA®
VTAM®
WebSphere®
World Registry™
xSeries®
z/Architecture™
z/OS®
z/VM®
zSeries®
1-2-3®

The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Archived

Preface

In this IBM® Redbook, we highlight many enhancements made in z/OS® Version 1 Release 3 and Release 4, and show how to use this document to help you install, tailor, and configure these releases.

First we provide a broad overview of z/OS Version 1 Release 3 and Release 4, and then we discuss how to install and tailor z/OS and the many components that have been enhanced: the z/OS base control program (BCP), Parallel Sysplex® and System-Managed CF Structure Duplexing, msys for Operations, Workload Manager (WLM), and zSeries® File System (zFS). Security functions such as Security Server (RACF®), PKI Services, LDAP server, Network Authentication service and Cryptographic services are also covered.

This redbook is intended for systems programmers and administrators responsible for customizing, installing, and migrating to these newest levels of z/OS.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Paul Rogers is a Consulting IT Specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on various aspects of z/OS and OS/390®. Before joining the ITSO 14 years ago, Paul worked in the IBM Installation Support Center (ISC) in Greenford, England providing OS/390 and JES support for IBM EMEA.

Luiz Fadel is a Senior Consulting IT Specialist, Certified from Brazil. He has 33 years of experience in Large Systems. Luiz holds a degree in electronic engineering from the Aeronautic Institute of Technology. His areas of expertise include Large System Hardware and Software, Parallel Sysplex, and Large System Design. His current responsibilities include zSeries pre-sale support for all countries in Latin America. He worked on assignment at the ITSO from 1977 through 1980, responsible for Large System hardware, and again from 1988 through 1990, responsible for Large System operating systems.

Erik Frederiksen is a Systems Programmer in Denmark with DMdata, which operates one of the larger set of zSeries sites in the Nordic area. He has more than 15 years of mainframe experience, encompassing both VM and z/OS. His areas of expertise include Workload Management, zSeries performance including DASD and JES2 aspects, general Parallel Sysplex implementation and performance and capacity evaluation.

Hirokazu Kakurai is an IT specialist with IBM Japan Systems Engineering Co. He has nine years of experience with IBM. His areas of expertise include z/OS, Parallel Sysplex, and WLM.

Henrik Thorsen is a Consulting IT Specialist, Certified from Denmark. He has 20 years of experience in Large Systems and has worked with IBM since 1985. He holds a Master's degree in Science from the Technical University of Denmark, as well as a Bachelor's degree in Economics from the Copenhagen Business School. His areas of expertise include a variety

© Copyright IBM Corp. 2003

of zSeries technical topics. Henrik worked on assignment at the ITSO from 1995 through 1997, responsible for Parallel Sysplex. He has written extensively on Parallel Sysplex, Workload Management, OS/390 and z/OS, as well as on various topics including implementation, capacity planning, and performance.

Frank Kyne is a Consulting IT Specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of Parallel Sysplex. Before joining the ITSO five years ago, Frank worked in IBM Global Services in Ireland as an MVS™™ Systems Programmer.

Yolanda Vivar is a Systems Programmer in La Caixa, Spain. She has five years of mainframe experience and worked at EDS Barcelona before assuming her present job. Her areas of expertise in the mainframe environment include DFSMS, VSM, and Storage Management-related products, as well as general UNIX System Services implementation.

Franco Meroni works for IBM Italy as a consultant with the EMEA South Region Technical Support team for S/390® and z/OS. He also cooperates with PSSC in Montpellier.

Thanks to the following people for their contributions to this project:

Richard Conway
International Technical Support Organization, Poughkeepsie Center

Robert Haimowitz
International Technical Support Organization, Poughkeepsie Center

Robert Vaupel
IBM Germany

Peter Baeuerle
IBM Germany

Scott Fagen
IBM US

Matthias Gubitz
IBM Germany

Sim Schindel
IBM Turkey

Juergen Holtz
IBM Germany

Georgette Kurdt
IBM US

Ron Northrup
IBM US

David Raften
IBM US

David Surman
IBM US

Soeren Understrup
IBM Denmark

Archived

Doug Zobre
IBM US

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Archived

Archived

1

z/OS Version 1 Release 3 and 4 overview

This chapter provides an overview of the functional enhancements made to z/OS V1R3 and z/OS V1R4.

This chapter describes the following functional enhancements to:

- BCP
- Workload Manager
- UNIX System Services
- Distributed File Service (DFS™)
- JES2 V1R4
- JES3 V1R4
- System Display and Search Facility (SDSF)
- Communication Server
- Firewall Technologies
- LDAP Server
- Network Authentication Service
- Open Cryptographic enhanced plug-ins (OCEP)
- RACF
- msys for Operations
- z/OS.e

© Copyright IBM Corp. 2003

1.1 z/OS Version 1 Release 3 and Release 4 BCP

The Base Control Program (BCP) provides essential operating system services. The BCP includes the following:

- I/O configuration program (IOCP)
- Workload Manager (WLM)
- System management facilities (SMF)
- The z/OS UNIX System Services (z/OS UNIX) kernel
- Support for Unicode

Beginning with z/OS V1R3 and z/OS.e V1R3, the BCP also includes the program management binder, which was formerly in the DFSMSdfp™ base element.

1.1.1 IBM z/OS service policy

At the time of writing, IBM tends to provide service support for each release of z/OS for three years following its general availability date. IBM may occasionally choose to support a release for more than three years. For updated information about z/OS service, marketing, and service withdrawal dates, refer to:

http://www.ibm.com/servers/eserver/zseries/zos/support/zos_eos_dates.html

1.1.2 Elements and features included in z/OS V1R3

A new feature of z/OS V1R3 is the binder.

Note: As of z/OS V1R3, the BCP also includes the binder, which was formerly in the DFSMSdfp base element.

1.1.3 Elements and features removed from z/OS V1R3

The following elements and features have been removed from the BCP:

- LANRES diskettes for base element.
- WLM compatibility mode. For more information, refer to 7.1, “Removal of WLM compatibility mode” on page 88.
- KEYRANGE support in DFSMS. For more information, refer to informational APAR II12896.
- STEPCAT/JOBCAT support in DFSMS. By installing APARs similar to OW25632 and OW41955, you may provoke warning messages allowing you to examine the use of STEPCAT/JOBCAT for possible removal.

1.1.4 Elements and features removed from z/OS V1R4

The host LANRES code is withdrawn in the z/OS V1R4. For migration information, see the white paper at:

http://www.ibm.com/eserver/zseries/library/whitepapers/gm130035.html

Optional feature Communications Server NPF is removed. The function is now part of the Communications Server base element.

Note: z/OS Infoprint® Server is IBM's strategic method for providing rerouting of print data to an IP network.

From the optional feature Security Server, the RDBM DB2® backend function of the LDAP Server component is removed. You are encouraged to migrate to the enhanced TDBM DB2 backend because of its improved scalability and availability.

Note: For instructions, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923.

1.2 z/OS V1R3 base control program (BCP)

When migrating to z/OS V1R3, there are many changes to the base control program, as follows:

- 64-bit binder and loader support
- WLM compatibility mode removal
- UNIX System Services
  - Access control lists (ACLs)
  - Automount facility enhancements
- TSO/E dynamic broadcast data set
- Parmlib changes
- System Logger logstream attributes
- ASID resource relief and monitoring
- Page data set protection

1.2.1 64-bit architecture

In z/OS V1R3, the 64-bit system infrastructure supports Addressing Mode 64, to allow applications to exploit 64-bit data addressability. This support includes the binder and the z/OS loader. This allows software vendors and customers to build, load, and execute AMODE 64 High Level Assembler programs.

This provides the capability to port applications that rely on 64-bit architecture. 64-bit Real memory improves overall price performance by making more efficient use of the installed processor memory. Data in memory above 2 GB can be accessed faster and with byte-level granularity. For programs that need large virtual data addressability, this support facilitates building and executing these programs in z/OS. The combination of the AMODE 64 function and the z/OS V1.2 64-bit virtual storage system service facilitates assembler language programs for broader 64-bit virtual exploitation.

1.2.2 WLM compatibility mode

The WLM compatibility mode option has been removed with z/OS V1R3. When migrating from a previous release in WLM compatibility mode, you have the following options:

1. If you are currently running in compatibility mode on an earlier operating system release, you can migrate directly to z/OS R3. When the z/OS V1R3 system is IPLed, this results in the system coming up in WLM goal mode. In this case, the system uses the default service definition that has been supplied.
2. If you are currently running in compatibility mode on an earlier operating system release, you can, as an intermediate step, migrate to WLM goal mode on the current release (that is, before upgrading to z/OS R3). You need to set up the appropriate service definition

using response time and velocity data. This would then be followed by the migration to z/OS V1R3 using the previous service definition established. This is the IBM recommended migration path.

Note: z/OS provides a job in SAMPLIB called IWMINSTL that loads and activates a sample policy for the WLM component. You can use this job to ease the migration to goal mode from compatibility mode.

For implementation considerations, see 7.1, “Removal of WLM compatibility mode” on page 88

1.2.3 Access control lists (ACLs)

Beginning with z/OS V1R3, you can use access control lists (ACLs) to control access to HFS and zFS files and directories by individual UIDs and GIDs. However, you must meet one of the following conditions in order to manage an ACL for a file:

- You are the file owner.
- You are a superuser (UID=0).
- You have READ access to SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class.

Currently, ACLs are supported by RACF for HFS and zFS. You must know whether your security product supports ACLs, and what rules are used, when determining file access.

For implementation considerations, see 8.1, “Access control list (ACL) support for V1R3” on page 122.

1.2.4 Automount facility

The following enhancements have been made to the automount facility to help administrators better manage their systems:

- Display current automount policy
- Support the number (#) character as a comment delimiter in the map file
- Allocate an HFS dynamically, if needed
- Generic match only on lower case names
- Support system symbolics in the map file

For implementation considerations, see 8.4, “Automount enhancements” on page 160.

1.2.5 TSO/E dynamic broadcast data set

The TSO/E broadcast data set contains notices and messages for TSO/E users in an z/OS system. There are current limitations in the use of the broadcast data set that could lead to system problems or multisystem outages, as follows:

- The broadcast data set must be named 'SYS1.BRODCAST'. Currently no other name is supported.
- Master JCL contains an entry for the broadcast data set which can only be changed with an IPL.
- Problems involving the broadcast data set can lead to slow performance when attempting to send messages and notifications to users.

With z/OS V1R3, the broadcast data set is allocated by TSO/E just after IPL and will no longer be included in the master JCL. The name of the broadcast data set is now specified in the IKJTSOxx parmlib member.

A new system command, called **D IKJTSO,SEND**, is added to display information that is currently displayed by the TSO/E **parmlib list** command which is enhanced to display the name and volume of the current broadcast data set.

Set ikjtso command

A new **SET IKJTSO=xx** command is added to process the IKJTSOxx parmlib member, including possibly switching to a new broadcast data set.

TSO/E logon enhancement

TSO/E logon is changed to allow users to log on even if the broadcast data set is not allocated. All the modules in TSO/E that hard code the name of the broadcast data set will be changed to allocate the broadcast data set name that was specified in the IKJTSOxx PARMLIB member.

For implementation considerations, see 3.1.1, “TSO/E broadcast data set” on page 32.

1.2.6 Change to parmlib

The following is a change to the parmlib members for z/OS V1R4.

IEASYSxx changes

The IEASYSxx member of parmlib is enhanced to support the new IKJTSO system parameter:

```
IKJTSO=xx
```

This parameter specifies the parmlib member, IKJTSOxx, from which TSO/E settings are obtained. The two-character identifier, represented by xx, is appended to IKJTSO to identify the parmlib member.

1.2.7 System Logger attributes

Currently, many MVS System Logger log stream attributes cannot be updated if there are any outstanding connections to the log stream. This is true when there are active or “failed-persistent” connections to the logstream. When a change needs to be made to most of the log stream attributes, all affected subsystems or applications must first be stopped (or quiesced) and successfully disconnected from the logstream. After there are no connections to the log stream in the sysplex, then the attribute updates can be made. The affected subsystems or applications can then be restarted (or resumed), and when they reconnect to the log stream, the new attributes are finally in effect. The same disruptive procedure has to be followed when these log stream resource attributes need to be changed back to their previous setting.

With z/OS V1R3, in a 7x24 operational environment, logger provides the capability to make, or at least initiate, log stream attribute updates even when there are current connections to a log stream. For example, an installation should be able to change the size of a log stream offload data set without having to follow the disruptive procedure just described.

Logstream policy attributes

Logger is enhanced to allow most of the logstream policy attributes to be accepted as “pending updates” when the log stream is in use at the time of the change request. The point at which the pending updates will take effect is based on the type of log stream change. Some of the pending update attributes will take effect during the next offload activity. Other attribute updates take effect at the next structure rebuild or on the next first connection to the log stream in the sysplex. This latter set provides the advantage of not having to disconnect, make the updates, and then cause the reconnection to the log streams. The updates to the log stream can be made at an earlier point, and then the pending updates will take effect at the next (scheduled or unscheduled) disconnect/reconnect activity.

Logger couple data set

Additionally, logger allows changes to structure definitions in the LOGR couple data set (CDS). Currently, no logger structure attributes can be updated without following a very disruptive procedure. In order to make a change to any logger structure definition, all the logger resources associated with the structure need to be deleted and then defined again with the updated attributes.

The logger support in z/OS V1R3 now allows a log stream definition to be associated with a different logger structure. The log stream cannot have any connection (active or failed-persistent) in order to be updated and be associated with a different logger structure. This support greatly reduces the need to provide specific logger structure attribute updates, since the affected log streams can be remapped to a logger structure that has the desired attributes.

New extended high level qualifier (EHLQ)

A new extended high level qualifier (EHLQ) parameter is supported on a log stream definition in the LOG CDS. This new parameter provides greater flexibility in naming the high level qualifier(s) to be used for the log stream's data sets. This increased flexibility aids in the setup and systems management of the MVS System Logger.

For implementation considerations, see 16.1.1, “System Logger enhanced logstream attribute support” on page 251, “Dynamic System Logger structure definition updates” on page 255, and “Extended high level qualifier support for System Logger” on page 255.

1.2.8 ASID resource relief and monitoring

z/OS V1R3 addresses the following problems when an address space becomes unusable:

- Minimizing the storage impact of specifying a large RSVNONR value in the IEASYSxx parmlib member.

  This encourages customers to specify a larger value, and thus allows them to run longer without exhausting their ASIDs.

  - RSVNONR is now excluded when determining the size of the CSCB/CSXB storage pool. This is a savings of up to 36 bytes of CSA per ASID specified with RSVNONR. This reduces the total storage required by each reserved ASID by about one-third. The new RSVNONR default is now 100.
  - Each ASID specified with RSVNONR now costs:
    - 4 bytes of SQA
    - 68 bytes of ESQA

- Minimizing the storage impact of “dead”, non-reusable ASIDs.

  Currently SQA storage remains allocated for every non-reusable ASID, and this can exhaust SQA and force an IPL to occur.
- Providing notification messages when resource exhaustion and replenishment occur.

  This allows installations to be more proactive on dealing with these situations, rather than waiting until all resources are totally gone.

For additional considerations, see 3.1.2, “ASID resource relief and monitoring” on page 35.

1.2.9 Page data set protection

With z/OS V1R3, page data set protection prevent two systems from accidentally using the same physical data set. When this occurs, it is very difficult to detect and diagnose and can cause data integrity and/or system failures. This support prevents the same data set from being used when possible, and where this is not possible, it protects the integrity of the system by causing system termination when it is detected. Full detection requires that the support be installed on all systems.

For implementation considerations, see 3.1.3, “Page data set protection” on page 36.

1.3 z/OS V1R4 base control program (BCP)

When migrating to z/OS V1R4, there are many changes to the base control program.

- System symbols
- Automatic restart management (ARM)
- Changes to parmlib

We describe these changes in the following sections.

1.3.1 System symbols

The number of static system symbols that can be defined is not sufficient for some environments. Therefore, the maximum size of the static system symbol table is increased in z/OS V1R4 to relieve this constraint. This increase in needed due to the increase in the use of symbols by msys for Setup.

z/OS V1R4 now provides support for defining additional installation-defined static system symbols, in addition to the system symbols that MVS provides, for each system in a multisystem environment. Previously, the limit was 99. This change guarantees the ability to define 800 symbols, but you can actually define as many as you can fit into the symbol table.

1.3.2 Automatic restart management

If you want to use Automatic Restart Manager on your V1R4 system, you must reformat the Automatic Restart Manager Couple Data Set to the z/OS V1R4 level. In V1R4, the automatic restart Couple Data Set has been expanded in size to accommodate the larger symbol table.

You can reformat the Couple Data Set as follows:

- From a z/OS V1R4 test system
- From a pre-V1R4 system by having a STEPLIB or JOBLIB DD in the IXCL1DSU utility point to the V1R4 SYS1.MIGLIB data set

1.3.3 Changes to parmlib

Following are changes to the parmlib members for z/OS V1R4:

BPXPRMxx member

The new parameter AUTHPGMLIST lets you specify the pathname of a hierarchical file system (HFS) file that contains the lists of APF-authorized pathnames and program names.

The new second NETWORK statement lets you define an Inet or Cinet configuration.

The new AUTOMOVE/NOAUTOMOVE parameter lets you specify where you want ownership of a file system to move if the owning system goes down.

IEASYMxx member

You can now define up to 800 static system symbols (for each system) in your environment using the SYMDEF parameter. Previously, you could define 99 static system symbols.

IECIOSxx member

There is now a default of **ALL** for the **BOX_LP** statement on the **HOTIO** and **TERMINAL** parameters, which specifies that all devices be boxed in the event that hot I/O conditions are detected.

1.4 Workload Manager

This section contains an overview of Workload Manager (WLM) enhancements in z/OS V1R3 and z/OS V1R4. For further information, refer to Chapter 7, “z/OS Workload Manager (WLM)” on page 87.

1.4.1 WLM enhancements in z/OS V1R3 at a glance

WLM is enhanced in z/OS V1R3 with the following support:

- Removal of WLM compatibility mode
- Policy activation from commands or batch programs
- Paging subsystem availability enhancement
- Introduction of WLM Enclave Service Class Reset
- WLM support for sub-capacity pricing

1.4.2 WLM enhancements in z/OS V1R4 at a glance

WLM is enhanced in z/OS V1R4 with the following support:

- Initiator balancing
  - New function is provided in WLM to rebalance batch initiators across the systems in a sysplex as the mix of workloads on these systems changes.
- msys for Setup
  - The msys for Setup exploitation by WLM provides support for new users migrating from WLM compatibility mode to goal mode to help administrators to quickly create and efficiently maintain WLM configurations with a minimum of input and knowledge of the component.
- WLM provides multilevel security that lets a system administrator establish a set of security criteria for a business workflow.

Let's look at the z/OS V1R4 enhancements in more detail.

Initiator balancing

Support in z/OS Version 1 Release 4 allows WLM to rebalance batch initiators across the systems in the sysplex as the mix of workloads on these systems changes.

WLM msys for Setup support

To customize WLM in goal mode, there are particular parmlib members, Couple Data Sets, and Coupling Facility structures that must exist. The setup and maintenance of such configurations requires detailed knowledge of z/OS and the WLM component.

Support in z/OS Version 1 Release 4 allows the WLM component for msys for Setup to provide easier and more efficient creation and maintenance of WLM configurations. The msys for Setup exploitation by WLM includes the following:

- WLM setup to run in goal mode
- Migration to new z/OS releases
- Enlarging the size of the WLM Couple Data Sets
- Resizing of the Coupling Facility structures for multisystem enclave support
- Resizing of the WLM LPAR cluster Coupling Facility structures

Using msys for Setup can make it easier to customize WLM in goal mode for the following:

- Particular parmlib members
- Couple Data Sets
- Coupling Facility structures

Modification of service definitions within the msys for Setup environment is not provided. Service definitions still have to be maintained by the existing ISPF WLM administrative application on the host.

1.4.3 Multilevel security

Workload Manager provides a new level of security that includes the definition (or labeling) of common access criteria for the data, programs, users and network that will process this workflow. This criteria has been adopted by many branches of the US government, and provides a security control that goes beyond what is currently available as part of z/OS.

1.5 UNIX System Services (USS)

z/OS UNIX has the following enhancements in z/OS V1R4 that simplify UNIX administration on z/OS and provide options that are consistent with other UNIX platforms:

- Administrators will have the ability to automatically assign an unused UID/GID value to a user/group.
- A system-wide setting prevents the assignment of a UID or GID value that is already in use. (With the proper authorization, assignment of a shared UID/GID is possible, however.)
- The SEARCH command is enhanced to allow an administrator to determine the set of users/groups that are assigned a UID/GID.
- Administrators can optionally assign a group owner of a new HFS file from the effective GID of the creating process.

1.5.1 Shared HFS support for z/OS V1R4

The BPXMCDS Couple Data Set is changed to hold additional data. If you want to use UNIX System Services shared HFS services on your z/OS V1R4 system, you must reformat the BPXMCDS Couple Data Set on a z/OS V1R4 level system.

Note: This must be done from a V1R4 system.

After an IPL of z/OS V1R4, define new primary and alternate version 2 type BPXMCDS Couple Data Sets. Use the sample job, BPXISCDS, in SYS1.SAMPLIB to do this. You can then enable the new version 2 type BPMCDS Couple Data Set as the primary Couple Data Set. UNIX System Services initialization then resumes.

Define the required user ID (uucp), and the required groupIDs (uupg and TTY) in the security database. If the names conflict with your current naming convention, you must create and activate a user ID alias table UNIX before z/OS 1.4 is installed.

1.5.2 Syslist support for shared HFS

This new function in z/OS V1R4 provides the ability to indicate *where* file systems should be moved to when a system leaves the sysplex, instead of being moved in an unpredictable fashion. Currently, when a system goes down, dead system recovery moves file systems that have been defined as automove=yes to another system in the sysplex. This is done in a random way. Because of performance issues and workload balancing, customers want the ability to specify where a file system can be moved to, as well as where it cannot be moved to.

To address this concern, now a system list can be added to the automove parameter for MOUNT. The list will be in priority order of systems, 1 to n, where n is the number of systems in the sysplex. The number of systems in the list can vary from 1 to n.

An indicator is used to define the list as an include (systems where the file system can be moved to in priority order) or exclude (do not move here) system list. If a syslist is specified and a file system cannot be moved to another system using the syslist rules during dead system recovery, it will be unmounted instead of being turned into a black hole.

1.6 Distributed File Service (DFS)

The Distributed File Service z/OS V1R4 support includes both SMB and zSeries Files Systems (zFS) enhancements.

The SMB enhancements include:

- Additional work station domain userid to MVS userid mapping options
- Reduction of system event notification overhead

zFS enhancements include:

- Improvements to the file system configuration support
- Additional administration capabilities
- Performance enhancements

1.7 msys for Setup

Managed System Infrastructure for Setup (msys for Setup), introduced in z/OS V1R1, offers a new approach for configuring z/OS, z/OS.e, and products that run on z/OS and z/OS.e. The configuration process is driven by a graphical user interface that greatly facilitates the definition of customization parameters. Updates are under the control of the msys for Setup administrator, and are made directly to the system.

msys for Setup in V1R4 delivers multiple user support allowing easier setup and configuration of z/OS products, which enables users to easily identify with specific roles and independent tasks running in parallel by different users.

For usability, the msys for Setup workplace tree view and update task have been improved. Improved package handling makes products known to msys for Setup easier to manage. Streamlining the Update process makes it easier for users to trigger the update of host resources. In addition, services allow for PROCLIB handling and additional parmlib members are supported.

The msys for Setup exploiters with z/OS V1R4 are:

- Parallel Sysplex
- Base sysplex
- TCP/IP
- ISPF
- USS
- Language Environment®
- LDAP Server

A suitable Java Runtime environment is shipped with the msys for Setup workplace. Java 1.3 is required on the z/OS host for msys for Setup exploitation.

Some framework highlights are as follows:

- Multiple user support allows more than one user at a time to work with msys for Setup on the same configuration.
- A progress indicator has been provided to indicate to the users the progress of long-running host jobs.
- Many ease-of-use enhancements to the msys for Setup workplace will make working with msys for Setup easier and more intuitive.
- msys for Setup provides messages, panels, and help text translated into Japanese.

1.7.1 z/OS V1R4 components

The following additional components use z/OS msys for Setup for their configuration:

- The LDAP plug-in provides the ability to configure multiple LDAP servers. The plug-in will make all the necessary configuration updates to create usable LDAP servers. In addition, configuration parameters (in slapd.conf, DSNAOINI, and slapd.envars) are discovered automatically for servers that are established.
- The TCP/IP plug-in provides customization of the TN3270 server as well as customization and refresh support for TCP/IP port reservations.

Archived

1.8 JES2 Version 1 Release 4

JES2 V1R4 provides support for the WLM balancing of initiators across the systems in the sysplex as the mix of workloads on these systems changes. The rebalancing of initiators across the sysplex should improve the performance across the system.

JES2 provides support for an Infoprint Server requirement to be able to suppress blank truncation for a SYSOUT data set. Currently, when JES2 detects an AFPDS (MO:DCA-P) data stream being written to the spool, JES2 does not do blank truncation even if the output class is defined to have blank truncation.

The JES2 resource monitor is intended to greatly improve overall JES2 RAS by giving service and the customer a window into what JES2 is actually doing when it cannot respond to commands. These problems occur frequently and generally involve a large amount of level 2 time to diagnose.

A common problem with diagnosis is that the customer dumps and restarts the member that is issuing the messages, and then calls IBM service. However, the messages are reporting a probable error on *another* member, and thus the dumps obtained are useless. When good documentation is obtained and a cause is found, it is often a user exit.

There is a new address space for JES2. The address space is where the new JES2 monitor runs. The name is based on the JES2 subsystem name and an installation-specifiable option. For a normal case, the name is JES2MON. For a JES2 subsystem called JESA, the name would be JESAMON.

JES2 no longer checks for multiple TSO logons with the same user ID. IBM has always recommended that an ENQ issued by TSO be used to prevent multiple TSO users from logging on in a sysplex. However, if an ENQ was not set up, JES2 would ensure that a TSO user only logged on once in a MAS.

JES2 has enhanced the way it reads its initialization stream. In earlier releases there was an INCLUDE statement that allowed a specific data set to be imbedded as part of the default PARMLIB concatenation. In z/OS V1R4, JES2 now supports reading members from the default PARMLIB concatenation. A default member name was also established for the JES2 initialization stream in the default PARMLIB concatenation.

JES2 checkpoint data error recovery is intended to improve recovery from corrupted checkpoint (and spool) data sets, thereby doing additional cold start prevention. Warm start processing is changed so that no physical writes to the checkpoint data sets will occur until after warm start processing completes. Therefore, if the first checkpoint data set is corrupted but the second checkpoint data set is still valid, the installation will have the opportunity to restart using the valid checkpoint data set.

The number of SPOOL I/O errors found during warm start processing as a result of errors in CBIO read processing of SPOOLed control blocks will be counted. If the count is greater than 10, the installation operator will be asked if they wish to continue or terminate the warm start. These changes will not prevent *all* cold starts, but it is one more step in addressing areas known to cause cold starts.

1.9 JES3 Version 1 Release 4

JES3 V1R4 provides the capability to add or change a MAINPROC statement during a hot start with refresh. This allows customers to add systems to their sysplex and not have to IPL. This avoids a warm start in order to add systems which require a sysplex-wide IPL.

Toleration for initialization errors

In previous releases, JES3 initialization errors caused JES3 to fail during initialization. This leaves the base operating system without a JES, and the JES3 initialization deck must be updated and JES3 must be restarted.

Some of these errors should be tolerated by providing a default value in the event a problem is detected. This would allow JES3 to initialize properly. The following initialization statements will be provided defaults:

- COMMDEFN
- OUTSERV
- STANDARDS
- SYSID
- SYSOUT

The OPTIONS statement is changed only to detect the presence of a duplicate statement. INQUIRY command support is provided for the following statements: OPTIONS, OUTSERV, STANDARDS and SYSOUT.

1.10 System Display and Search Facility

System Display and Search Facility (SDSF) provides you with information to monitor, manage, and control your z/OS or z/OS.e system. Although prior levels of JES2 may be used with z/OS V1R4, prior levels of SDSF may not.

Note: In z/OS.e, BookManager® help is not available for SDSF SYSLOG messages. This is because BookManager READ is not available in z/OS.e

z/OS V1R4 SDSF requires z/OS V1R4, and JES2 in the MAS at OS/390 R8 or higher. z/OS V1R4 has the following support.

1.10.1 MEMLIMIT column on SDSF DA Display

SDSF has been enhanced to show additional 64-bit information for address spaces for sysplexes in which at least one system is in 64-bit mode. A new MEMLIMIT column is added to the SDSF DA display. The data for this display and the corresponding SMF type 73 record is provided by RMF™.

1.11 Communication server

z/OS Version 1Release 4 is the first release to support IPv6 functions. z/OS V1R4 includes support for the basic IPv6 functions as follows:

- Stateless auto configuration
- Static IPv6 routing tables
- Static Virtual IPv6 Addresses (VIPA)
- Basic sockets API enhancements for IPv6
- Selected applications enabled for IPv6

Networking enhancements in Communication server support include:

- A new Simple Network Protocol (SNTP) server allows computers in a network connected to z/OS to synchronize their clocks with the zSeries Sysplex Timer®.

- TN3270 support adds new capabilities in the areas of security, configuration simplification, and flexibility. The support includes Transport Layer Security (TLS) in addition to the existing Security Sockets Layer (SSL) support.
- There is a more consistent interface to various security-related exit points, including a capability for the exit points to exchange data with each other, that enhances FTP in the area of improved activity logging.
- FTP adds codepage conversion capabilities for the new Chinese codepage known as GB18030.
- By enhancing addressing for lines and PUs, there are improvements in DIAL processing and enhanced HPR diagnostics
- Enterprise Extender (EE) is improved in the areas of usability, availability, scalability and serviceability.
- Improvements to VARY, ACT, UPDATE, SDUMP, and CSALIMIT processing.

1.11.1 Sysplex-wide dynamic VIPA

z/OS V1R4 allows sysplex-wide dynamic VIPAs for TCP/IP connections to have the same single IP address appearance for application instances initiating outbound connections within a sysplex as Sysplex Distributor provides for inbound connections. This capability simplifies IP address and workload management, improves security, and allows greater scalability and flexibility.

1.12 Firewall Technologies

The Internet Security Association and Key Management Protocol (ISAKMP) server and the configuration server of Firewall Technologies are packaged with the Security Server but licensed with the base operating system, and can be used without licensing or enabling the Security Server.

The ISAKMP server implements the required elements of Internet Key Exchange (IKE) as defined by Request for Comments (RFC) 2409. The Configuration server communicates with the firewall configuration graphical user interface (GUI) that is shipped within Firewall Technologies. Firewall Technologies uses the DES algorithm for encryption.

1.13 LDAP Server

The RDBM DB2 backend function of the LDAP Server is removed as of z/OS V1R4. You are encouraged to migrate to the enhanced TDBM DB2 backend because of its improved scalability and availability. For instructions, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923.

1.14 Network Authentication Service

Network Authentication Service is licensed with the base operating system and can be used without ordering or enabling Security Server. Network Authentication service uses the DES algorithm for encryption. Prior to z/OS V1R2, this component was named Network Authentication and Privacy Service.

1.15 Open Cryptographic Enhanced Plug-ins (OCEP)

As of z/OS V1R3 and z/OS.e V1R3, OCEP is licensed with the base operating system and can be used without ordering or enabling Security Server.

1.16 RACF

RACF uses the limited DES and CDM algorithm, and the RC2 40-bit algorithm, for encryption.

New in z/OS V1R3 and z/OS.e V1R3 is Public Key Infrastructure (PKI) Services, a component that uses RACF, the OCSF Base component of base element Cryptographic Services, and the ICSF component of base element Cryptographic Services, for encryption.

As of z/OS V1R3 and z/OS.e V1R3, the name “SecureWay®” is dropped.

1.17 Enterprise Identity Mapping (EIM)

Enterprise Identity Mapping (EIM) is an IBM @server infrastructure that allows administrators and application developers to solve the problem of managing multiple user registries across their enterprise. This infrastructure provides a common set of APIs that can be used across platforms to develop applications that look up the relationships between user identities and a single EIM identifier that represents a user in the enterprise.

The z/OS implementation of EIM consists of the following:

- The EIM C/C++ APIs for administering EIM and performing lookups
- The z/OS EIM admin utility, a z/OS UNIX System Services shell command for populating an EIM domain with enterprise identifiers and mappings between the identifiers and registry users
- A guide and reference manual

EIM will be downloadable from the Internet and will be SMP/E-installable.

Note: IBM is making EIM available on OS/400®, z/OS, AIX®, LINUX, and Windows 2000.

The following are EIM prerequisites:

- z/OS V1R4
- z/OS V1R4 Security Server LDAP
- z/OS V1R4 Security Server Enterprise Identity Mapping

z/OS V1R4 Security Server RACF or its equivalent is optional.

1.18 z/OS.e

z/OS.e is a new product for the IBM zSeries 800 that provides selected function at an attractive price. z/OS.e is intended to help you exploit the fast-growing world of next generation e-business by making the deployment of new workloads on the z800 competitively priced.

z/OS.e shares the same code base as z/OS, and will only run in an LPAR on a z800. The general availability date for z/OS.e V1 R3 was March 29, 2002.

z/OS.e is for new, next generation e-business workloads. Specifically, z/OS.e is an attractive solution when you need an Enterprise Application Server (EAS) or database server for customer relationship management (CRM), supply chain management (SCM), enterprise resource planning (ERP), business intelligence, and DB2-based solutions that are written in programming languages of Java or C/C++. z/OS.e is not for traditional workloads.

For certain workloads, z/OS.e on the z800 offers the ability to:

- Obtain z/OS qualities of service for these new workloads on the mid-range zSeries 800 server at a reduced cost.
- Potentially reduce the total cost of ownership (TCO) of hardware, software, resources, and environmentals when consolidating workload from other alternative non-IBM platforms.
- Implement a low-cost database server for Linux application servers which can run on other zSeries or other IBM processors such as the IBM pSeries™ servers or IBM xSeries® servers.

1.18.1 Installing z/OS.e

z/OS.e is only available as either a ServerPac or a SystemPac®. The program number is 5655-G52. Note that z.OS.e is available on ShopzSeries.

z/OS.e is ideal for customers who like the qualities of service that the mainframe has to offer and have considered running more applications on the mainframe. z/OS.e allows new workloads to run on the mainframe more easily due to its reduced total cost of ownership and exceptional robustness and functionality; specifically:

- z/OS.e is economical.

  At only a fraction of the cost of regular z/OS, the deployment of new e-business workloads on the mainframe has become very competitive against alternative platforms. z/OS.e is a low monthly charge per z800 engine. The extraordinary pricing is applicable to all engines licensed to run z/OS.e, regardless of the computing environment or whether z/OS.e LPARs occupy the full capacity of a z800 server or just one engine of a z800. In addition, z/OS.e pricing is maintained even when z/OS.e is part of a Parallel Sysplex cluster.
- z/OS.e equals z/OS with restrictions.

  z/OS.e provides an operating environment that is comparable to z/OS in qualities of service, management, reporting, and skills. z/OS.e does not replace z/OS. Instead, z/OS.e has the same code base as z/OS, but customized with new system parameters. z/OS.e activates restrictions against running traditional workloads, while it provides an attractive price for running new, next generation workloads.
- z/OS.e embraces e-business solutions.

  z/OS.e is designed specifically for today's hot applications like WebSphere®, Enterprise Java (J2EE), C/C++, WebSphere Commerce DB Serving, Domino™ and other applications. Contact your ISV for support information. z/OS.e is not for traditional workloads. Customers will have contractual and technical restrictions on such workloads.

1.18.2 z/OS and z/OS.e differences

z/OS.e shares the same code base as z/OS; however, this code base is customized with new system parameters and has a new product number different from that of z/OS. z/OS.e, however, is different from z/OS.

The new system parameters activate restrictions against running traditional workloads. z/OS.e can integrate with traditional data transaction processing workloads on z/OS or OS/390, or collaborate with Linux for zSeries, providing the flexibility to optimize your traditional and new workloads on the zSeries platform.

z/OS.e is for the z800 only. And like z/OS on a z800, you get such technology as:

- Intelligent Resource Director
- Workload Manager
- HiperSockets
- FICON™ Express
- Parallel Sysplex
- Capacity Backup
- 64-bit real support

z/OS.e IPLs only in LPAR mode (not basic mode). z/OS.e can also run in an LPAR as a guest under z/VM®, but not VM/ESA® because VM/ESA cannot support 64-bit z/Architecture™ guest operating systems. Like z/OS on a z800, z/OS.e can run in a whole z800 processor, or run in only one engine of a z800 processor. Like z/OS, z/OS.e can participate in a Parallel Sysplex cluster.

In a z/OS.e environment, IBM requires that you submit Transmit System Availability Data (TSAD) for:

- A z800 with a z/OS.e license, or
- A z800 with both a z/OS.e license and sub-capacity Workload License Charges

You can submit TSAD by using the IBM Remote Support Facility (RSF) available on the z800, or by mailing a diskette or DVD cartridge to IBM.

If you are using RSF, an IBM service representative can enable it for you when your z800 is installed. If RSF is not enabled, contact your IBM service representative.

JES differences

You cannot use a prior JES2 or JES3 level with the current level of the rest of z/OS.e. With z/OS, it is allowable to use certain prior JES levels as a means of easing a migration to the current level of z/OS.

TSO/E differences

An additional restriction of z/OS.e is that only eight concurrent TSO sessions will be allowed. These are intended to support system programmers and administrators with activities to maintain the system. You may have more than eight userids defined, but only eight can be concurrently logged on.

Language Environment considerations

Run-time library services (RTLS) is not supported with z/OS.e. With z/OS, RTLS allows you to access different levels of the Language Environment run-time libraries, controlled by run-time options. In addition, other mechanisms such as using STEPLIB DD statements or the link list (whereby an older level of Language Environment is used for application execution) are not allowed. Using these mechanisms violates the z/OS.e license agreement.

The Language Environment library routine retention (LRR) function is not supported with z/OS.e. (With z/OS, LRR can improve the performance of applications and subsystems.)

Archived

Compatibility pre initialization for C and PL/I is not licensed with z/OS.e. In z/OS.e, you cannot use the C/C++ Run-Time Library functions __osname() or uname() to find out if the operating system is z/OS.e. The functions do not return a result unique to z/OS.e; they return the same result as in z/OS.

1.18.3 Differences in element, features, and functions to z/OS

Installations who choose z/OS.e are not licensed for the traditional workloads. This includes:

- CICS®
- IMS™
- COBOL

> **Note:** However, precompiled COBOL DB2 stored procedures, and other precompiled COBOL applications using the Language Environment preinitialization interface (CEEPIPI), are supported.

- Fortran
- Compilers for COBOL, PL/I, and VisualAge® PL/I, and Fortran

> **Note:** However, z/OS.e supports execution of precompiled PL/I and VisualAge PL/I applications

Base elements and optional features

Also not licensed and not functional are selected z/OS base elements and optional features, as well as certain selected functions within those elements and features that support traditional workloads.

In addition, selective applications and certain DB2 features will not be available to you when using z/OS.e. z/OS.e does not support the QMF™ Host and QMF HPO features of DB2 Universal Database™ Server for OS/390 V6 (5645-DB2) and DB2 Universal Database Server for OS/390 and z/OS V7 (5675-DB2), because they require the GDDM® traditional base element which is not functional in z/OS.e.

Not functional and not licensed elements, features, and functions include:

- BookManager READ/BUILD
  - If you want to upload BookManager softcopy and create softcopy repositories to support BookManager BookServer on your z/OS.e host, the SoftCopy Librarian is the strategic tool for uploading and managing BookManager files on z/OS and z/OS.e and on LANs and workstations.
- GDDM
- GDDM PGF feature
- GDDM REXX feature
- DCE Application Support
- LANRES (removed as of z/OS V1R3)
- BDT File to File

Also not licensed in the z/OS.e environment are:

- Encina® Toolkit Executive
- MICR/OCR

Any application written with these products, using these products, or anything requiring any of these products as a prerequisite will not run and will generate an error.

Note: Non-IBM ISV and custom-written applications, which use the base elements, features, products, and are written in the programming languages mentioned, may not be supported. Consult your ISV for more details. You can refer to the following Web site for a list of vendors who support z/OS.e:

http://www.ibm.com/servers/eserver/zseries/solutions/s390da/r13e.html/

1.18.4 New e-business workloads supported by z/OS.e (and z/OS)

The following is not an exhaustive list of the “new workloads” that z/OS.e (and z/OS) support, but it does include the most common ones:

- All service supported levels of WebSphere Application Server, WebSphere Commerce, Lotus® Domino.
- Any industry or enterprise application written in Java, Enterprise Java (J2EE specification), and C/C++.
- Most IBM middleware, like DB2, MQSeries®, and most ISV middleware and tools
- Most Independent Software Vendor (ISV) and Utility Software Vendor (USV) products that support new workloads, as long as they do not invoke any of the fenced products on z/OS.e

1.18.5 Required parmlib customization for z/OS.e

Very few items need to be changed:

- IEASYSxx - change LICENSE=z/OS to LICENSE=z/OSe
- IFAPRDxx - change product number from 5694-A01 to 5655-G52

1.18.6 LPAR update

Using HCD, change LPAR name to ZOSExxxx

1.18.7 z/OS.e Web site

You can find the z/OS.e Web site at the following URL:

www.ibm.com/eserver/zseries/zose/

Archived

Archived

2

Installation of z/OS Version 1 Release 3 and 4

This chapter provides installation details necessary to install z/OS V1R3 and z/OS V1R4. If you have a choice, it is strongly recommended by IBM that you install z/OS V1R4, as this release is considered a positioning release for future enhancements.

This chapter discusses the following:

- Processor requirements
- DASD space requirements for install
- ServerPac installation changes for z/OS V1R4
- Coexistence, migration, and fallback policy

Archived

© Copyright IBM Corp. 2003

2.1 Processor requirements

Both z/OS V1R3 and z/OS V1R4 run on IBM or comparable zSeries servers, shown in Figure 2-1, as follows:

- IBM zSeries - 800 (z800) and 900 (z900)
- IBM S/390 Parallel Enterprise Server™ - Generation 5 (G5) and Generation 6 (G6)
- IBM S/390 Multiprise® 3000 Enterprise Server

Note: The G5, G6, and Multiprise 3000 servers are now considered zSeries servers. However, z/Architecture functions, including HiperSockets and IRD, are not available on these servers.

Further, observe that z/OS V1R5, expected in 1Q2004, is planned to run on the same IBM servers as listed for z/OS V1R3. Also observe that IBM plans to deliver new releases on a a yearly basis every September after the September 2004 release.

Driver 26 (LIC V1.6.2) and upwards is required for z/OS to support architectural enhancements.

For further information, refer to:

http://www.ibm.com/servers/eserver/zseries/zos/

Figure 2-1 Hardware supporting operating systems

Both z/OS.e V1R3 and z/OS.e V1R4 run on the following IBM server:

- IBM zSeries 800 (z800)
  - The z800 must run in LPAR mode (not basic mode) on the z800 server, and it must run at a particular maximum capacity that you define. You define the capacity in terms of millions of service units per hour (MSUs).

2.1.1 z.OS.e and z.OS release cycle synchronization

z/OS and z/OS.e are synchronized in terms of release availability; that is, they have the same release number and become available at the same time. For example, the z/OS.e release that corresponds to z/OS V1R4 is z/OS.e V1R4. When z/OS V1R5 becomes available, so does z/OS.e V1R5.

z/OS and z/OS.e are also synchronized in terms of new functions. When new functions are added to a release of z/OS, they are also added to the same release of z/OS.e. However, not every element and feature in a given release will contain new functions. Those elements and features that do not receive new functions will, nevertheless, be delivered with all applicable service. The level of service included is described in the current version of *z/OS and z/OS.e Planning for Installation*, GA22-7504.

2.1.2 z/OS.e V1R3 and V1R4 MSU capacity

Use z/OS.e HCD to define the processor, I/O resources, and logical partitions (LPARs) you will use. For guidance, refer to *z/OS Hardware Configuration Definition User's Guide*, SC33-7988. When you define the name of the LPAR, you must use the form ZOSExxxx, where xxxx is any valid combination of zero to four characters.

Use the z800 hardware management console (HMC) to create an activation profile. In the profile you specify the processors and memory allocated to each LPAR, as well as the MSU capacity available for each LPAR. For information about using the HMC application, see *Hardware Management Console Operations Guide*, an online book that ships with the HMC. For information about z800 MSU capacity, see the z800 Software Pricing Configuration Technical Paper at:

http://www.ibm.com/eserver/zseries/library/techpapers/pdf/gm130121.pdf

Configure your system to send Transmit System Availability Data (TSAD) using the Remote Support Facility (RSF) available on the z800 processor. For more information about sending TSAD, see the Customize Scheduled Operations task for the Hardware Management Console or the Scheduled Operations task for the support element in the Hardware Management Console Operations Guide.

2.2 DASD space requirements

The installation of z/OS V1R4 base elements and optional features requires 20 cylinders on a 3390 device. This uses the SMP/E SMPPTS compaction function first provided in OS/390 V2R5. The total space required by the new and changed FMIDs in SMPTLIB data sets is 2,656 cylinders on a 3390 device.

If you are migrating from levels of OS/390 prior to z/OS, you will see increased need for DASD; how much more depends on what levels of products you are running. Keep in mind the DASD required for z/OS includes:

- All elements
- All features that support dynamic enablement, regardless of your order
- All un-priced features that you ordered

Figure 2-2 on page 24 shows the space requirements for the releases that coexist with Z/OS V1R4.

Figure 2-2 OS/390 and z/OS target data set sizes (3390 cylinders) by release

The total storage required for all the target data sets listed in the space table in the z/OS Program Directory is 4,840 cylinders on a 3390 device. The total storage required for all the distribution data sets listed in the space table is 6,446 cylinders on a 3390 device.

The total HFS storage is 2,250 cylinders on a 3390 device: 2,200 cylinders for the ROOT HFS, and 50 cylinders for the /etc HFS.

The total storage required for all the SMP/E data sets listed in the space table is 65 cylinders on a 3390 device, not including the SMPLTS (which uses 1,318 3390 cylinders) and the SMPPTS.

Refer to Figure 2-3 on page 25 for the storage requirements by release.

Archived

Figure 2-3 OS/390 and z/OS HFS data set sizes (3390 cylinders) by release

2.3 ServerPac installation changes for z/OS V1R4

During the install process there are several changes that need to be considered, as follows.

2.3.1 RESTFS job

The RESTFS job (and corresponding subsystem jobs) previously required the submitting user ID to have UID(0) set in its OMVS segment. Access to the BPX.SUPERUSER profile in the FACILITY class would provide the required authority without the need for UID(0). The ServerPac for z/OS V1R4 now sets the effective UID to zero when the user ID has access to BPX.SUPERUSER. This eliminates the need for the user who is executing the restore of the ServerPac to have UID(0) authority.

> **Note:** This does *not* affect the pax utility. You still require UID(0) in order to execute the pax command outside of the ServerPac processing. This change is for ServerPac processing only.

RESTFS is historically the number one source of service calls. To help eliminate some of the problems, logic checking for common setup problems, new error messages, and more specific error messages for better diagnosis, have been added. RESTFS calls pax and BPXISETS from with the exec to eliminate bypass checking and job elimination. A new status message indicator shows the progress of the job.

BPXISETS job eliminated

The RESTFS changes have eliminated the need to run BPXISETS from the ServerPac dialogs. An element change has removed the need to run the FOMISCHO job.

2.3.2 Select JES at install

Previously, both JES2 and JES3 were installed by running the installation job stream. The customer would then delete the JES element not wanted, and then optionally merge the JES used with the BCP zones and reallocate the DASD space used by the deleted element.

With the z/OS V1R4 ServerPac, the customer can choose one JES element or both in the dialog, and specify whether they are to be merged. Then the installation job stream is run.

> **Note:** The new ServerPac function of JES selection has removed the J2MERG, J2DELETE, J3MERG, J3DELETE, and UPDCSI jobs.

2.3.3 Master catalog flag

ServerPac provides a new **CHange** command (**CH MCAT Y | N**) which allows the required setting in the master catalog flag to be overridden. This allows an installation to catalog all data sets in user catalogs if they choose to.

Display support will be added for the “overridden” status in the View and Change option of the Modify System Layout. The changed option is not saved and has to be done for each ServerPac order.

2.3.4 Restructured ALLOCDS job

Allocation in the ALLOCDS job used IEFBR14 and DD statements. This always resulted in a return code zero whether it worked or not. This forced the user to read thousands of lines of allocation messages to detect failures. ALLOCDS also ran at the initiator address space priority, which could sometimes cause system performance problems. If the job needed to be restarted, the user had to search through all allocation messages, fix the problems, edit many lines of JCL, and resubmit. The alternative was to reinitialize all the DASD volumes and start over.

With z/OS V1R4, the common problems causing failures are not detected before allocation. A majority of the allocations are now done with IDCAMS ALLOCATE commands, where possible. All allocation failures now cause a nonzero return code and the job stops at the first problem. All the pertinent error messages are found in SYSPRINT output. ALLOCDS can be restarted in the appropriate step without modification when there is a problem. Instructions on how to restart are now provided. The changes now allow ALLOCDS to be WLM-managed.

2.3.5 msys for Operations support

In z/OS V1R4, ServerPac provides customization support of the msys for Operations element by:

- Creating some of the msys for Operations VSAM and non-VSAM data sets. One of the data sets, a user DSIPARM data set, contains customized values of the msys for Operations settings. The sample job to do this is in INGALLC0 of SINGSAMP data set.
- Setting up three sample PROCs (in CPAC.PROCLIB), which are provided in SINGSAMP.
- Modifying the supplied MPFLSTxx members in CPAC.PARMLIB to support msys for Operations.
- Defining the msys for Operations application major nodes to VTAM®.
- Making changes to the ServerPac jobs RACFTGT and RACFDRV to support the msys for Operations security definitions, which includes class, autotasks, and operator definitions.

- Copying the msys for Operations style sheet into the DSIPARM data set.
- Copying the AOFCUST member to the DSIPARM data set after customization.
- Supplying a VTAM logon mode table called AMODETAB in the VTAMLIB (in CPAC.VTAMLIB), that has the required changes for msys for Operations.

2.4 Coexistence, migration, fallback policy

Coexistence, migration, and fallback are so closely related that a single policy governs all three. The policy is the same for both z/OS and z/OS.e. The policy is as follows.

- For all elements and features except JES2 and JES3:
  - Four consecutive releases of z/OS (including its predecessor, OS/390) and z/OS.e are supported for coexistence, migration, and fallback. This means that three consecutive releases may coexist with the current release, that the earlier-level release to which you back out must be within three consecutive releases of the current release, and that the release from which you migrate must be within three consecutive releases of the current release. Thus, the normal period for coexistence, migration, and fallback is a maximum of two years, based on the current six-month release cycle.

    However, there is an exception. Due to a special provision, OS/390 R10 and z/OS V1R1 are treated as a single coexistence-migration-fallback level rather than two levels, because of the unique relationship between OS/390 R10 and z/OS V1R1.
- The four consecutive releases that may coexist (either within a sysplex or elsewhere) with z/OS V1R4 are:
  - z/OS V1R4 or z/OS.e V1R4
  - z/OS V1R3 or z/OS.e V1R3
  - z/OS V1R2
  - OS/390 R10 and z/OS V1R1 (both are treated as a single level).
- For JES2 and JES3:
  - While the four-consecutive-release policy also applies to JES2 and JES3, the way in which four consecutive releases is determined is different than for the rest of the operating system.

    If a JES2 or JES3 release is functionally equivalent to its predecessor (that is, its FMID is the same), then from a coexistence-migration-fallback standpoint the release is considered to be the same JES release.

2.4.1 z/OS V1R4 coexistence

Coexistence and fallback play an important part in planning for migration to the latest release. They are very much related in that both deal with an earlier level of a system being able to tolerate changes made by a later level.

Coexistence occurs when two or more systems at different software levels share resources. The resources could be shared at the same time by different systems in a multisystem configuration, or they could be shared over a period of time by the same system in a single system configuration.

Examples of coexistence are as follows:

- Two different JES releases sharing a spool
- Two different service levels of DFSMSdfp sharing catalogs

- Multiple levels of SMP/E processing SYSMODs packaged to exploit the latest enhancements
- An older level of the system using the updated system control files of a newer level (even if new function has been exploited in the newer level)

JES coexistence

JES2 and JES3 are treated a bit differently than the rest of the operating system for the coexistence policy. While the four-consecutive-release coexistence policy applies to both JES2 and JES3, the fact that JES installation can be staged has been taken into account in determining which are the four consecutive JES releases that may coexist. If a JES2 or JES3 release is functionally equivalent to its predecessor (that is, its FMID did not change), then from a coexistence standpoint this is considered the same JES release.

2.4.2 Rolling IPLs

A rolling IPL is the IPL of one system at a time in a multisystem configuration. You might stage the IPLs over a few hours or a few weeks. The use of rolling IPLs allows you to migrate each z/OS or z/OS.e system to a later release, one at a time, while allowing for continuous application availability.

z/OS V1R4 and z/OS.e V1R4 systems can coexist with specific prior releases of z/OS, z/OS.e, and OS/390 systems. This is important because it gives you flexibility to migrate systems in a multisystem configuration to z/OS V1R4 or z/OS.e V1R4 using rolling IPLs rather than requiring a systems-wide IPL. The way in which you make it possible for earlier-level systems to coexist with z/OS V1R4 or z/OS.e V1R4 is to install coexistence service (PTFs) on the earlier-level systems.

2.4.3 Fallback

Fallback (backout) is a return to the prior level of a system. Fallback can be appropriate if you migrate to z/OS V1R4 or z/OS.e V1R4 and, during testing, encounter severe problems that can be resolved by backing out the new release. By applying fallback PTFs to the “old” system before you migrate, the old system can tolerate changes that were made by the new system during testing.

Fallback and coexistence are alike in that the PTFs that ensure coexistence are the same ones that ensure fallback, and the releases to which you can fall back are the same ones that can coexist.

2.4.4 JES3 coexistence

During JES3 processing on the global, downlevel processors identify (in a data area shared between JES3 processors) what their level of JES3 is, so that a global at the z/OS 1R4 level can make decisions about that processor based on its level.

The following PTFs for APAR OW52172 *must* be installed for coexistence, migration, and fallback irrespective of whether MAINPROCs are being added, deleted, or changed:

- UW86764 for OS/390 V2R8
- UW86765 for OS/390 V2R9
- UW86766 for OS/390 V2R10 and z/OS V1R1
- UW86767 for z/OS RV12 at PUT0210

Therefore, this support allows future JES3 releases to coexist with HJS7705, HJS7703, or HJS6609; and provides the capability to fall back to HJS7705, HJS7703, HJS6609, or HJS6608. This service can be installed on all processors in any order.

To activate the PTF, use JES3 restart (HOT,local). If you have any usermods referencing TVTMAINA (IATYTVT), you must change this code to reference TVTMAINJ instead, even though TVTMAINA is not being deleted in this APAR. If you do not make this change, and it becomes necessary to fall back from a JES3 release higher than HJS7705, TVTMAINA will contain all zeroes and your code may function incorrectly because of this value. If you have any usermods referencing MPMSGCLS (IATYMPE) you must rework or delete this code, even though MPMSGCLS is not being deleted in this APAR.

If you do not make this change, and you either have a mixed complex with a JES3 release higher than HJS7705 or it becomes necessary to fall back from that release, MPMSGCLS will contain all zeroes and your code may function incorrectly because of this value.

JES3 coexistence policy

As of z/OS V1R2, compliance to the coexistence-migration-fallback policy for JES3 is enforced. A migration to a JES3 release level that is not supported by the policy results in the following:

- If the JES3 release level for a local is not compatible with the global in a JES3 multisystem complex, message IAT2640 is issued and the JES3 local is not allowed to connect to the global.

The following JES3 releases may coexist with z/OS V1R3 JES3 in the same multisystem complex:

- OS/390 V2R8 JES3
- OS/390 V2R9 JES3
- OS/390 V2R10 JES3 and z/OS V1R1 JES3 (both are functionally equivalent)
- z/OS V1R2 and V1R3 JES3 (both are functionally equivalent)

2.4.5 JES2 coexistence

The JES2 coexistence support PTFs, which are needed for all new JES2 and pre-Release 4 JES2 to coexist in the same MAS, are only required on downlevel releases if the systems will be in the same MAS as z/OS 1R4 level JES2. The PTFs for APAR OW52833 are as follows:

- UW87739 for OS390 R8 and R9
- UW87740 for OS/390 R10 and z/OS V1R1
- UW87741 for z/OS V1R2 and V1R3

JES2 Coexistence policy

As of z/OS V1R2, compliance to the coexistence-migration-fallback policy for JES2 is enforced. A migration to a JES2 release level that is not supported by the policy results in the following:

- If the JES2 release level for a system that is initializing is not compatible with the other active systems in the JES2 MAS, message HASP710 is issued and the JES2 address space for the initializing system is terminated.

The following JES2 releases may coexist with z/OS V1R3 JES2 in the same multi-access spool (MAS):

- OS/390 V2R8 and V2R9 JES2 (both are functionally equivalent)
- OS/390 V2R10 JES2 and z/OS V1R1 JES2 (both are functionally equivalent)
- z/OS V1R2 and V1R3 JES2 (both are functionally equivalent)

2.4.6 msys for Setup coexistence

After migration to z/OS V1R4, then z/OS V1R1, z/OSV1R2, and z/OS V1R3 systems running msys for Setup code are required to install the download msys for Setup coexistence support FMID JMSI743. The coexistence is SMP/E-installable.

Important: If at least one z/OS V1R4 or coexistence msys for Setup workplace is used, then the msys for Setup workplace of the previous z/OS releases must *not* be used anymore.

FMID JMS1743

This new Web-downloadable FMID provides coexistence with z/OS V1R4 msys for Setup. The z/OS msys for Setup contains a new level of the workplace. Once the R4 workplace is downloaded and started, the user is prompted to convert their LDAP tree (where the configuration data is stored). If they choose to convert the tree, then the downlevel systems will no longer be usable with the converted LDAP tree. If they choose *not* to convert the tree, the z/OS R4 system cannot be used for msys for Setup.

Note: JMSI743 will be packaged and installed on z/OS V1R1, z/OS V1R2, and z/OS V1R3.

Archived

3

Base control program (BCP)

This chapter describes enhancements made to the base control program (changes to the BCP components are described in other chapters).

It describes changes to z/OS V1R3 as follows:

- TSO/E broadcast data set
- ASID resource relief and monitoring
- Page data set protections

Archived

© Copyright IBM Corp. 2003

3.1 z/OS V1R3 BCP enhancements

The following changes to BCP functions have been made in z/OS V1R3:

- TSO/E broadcast data set
- ASID resource relief and monitoring
- Page data set protection

3.1.1 TSO/E broadcast data set

For z/OS V1R3, allocation of TSO/E broadcast data was changed to prevent system problems or multisystem outages caused by previous limitations in its use.

Prior to z/OS V1R3, if the volume on which the SYS1.BRODCAST data set resides failed, a sysplex-wide IPL was required for TSO/E to use a new SYS1.BRODCAST data set. In addition, lack of SYS1.BRODCAST data sets could cause systems to crash due to CSA exhaustion.

New parameters are added to the IKJTSOxx parmlib member as follows:

BROADCAST Indicates the broadcast data set processing options

SYSPLEXSHR Indicates whether the broadcast data set is shared outside systems in the sysplex

These options allow more flexible broadcast data set processing. The name of the broadcast data set is no longer hardcoded as SYS1.BRODCAST. Instead, you can specify the broadcast data set name that you wish to use on the BROADCAST keyword on the SEND statement of the IKJTSOxx parmlib member.

Note: You can switch to a different broadcast data set dynamically, without an IPL. The entry for the broadcast data set is no longer included in the master JCL. Instead, TSO/E allocates the broadcast data set during IPL.

As a consequence, you may remove your MSTJCLxx broadcast reference (SYSLBC DD), since it is ignored in z/OS V1R3 and higher releases.

Archived

Broadcast parameter details:

BROADCAST(DATASET(ds-name) VOLUME(volume-name) TIMEOUT(time-out) switch-prompt)

ds-name specifies the fully qualified data set name. The use of quotes in the data set name is ignored; that is, 'SYS3.BRODCAST' is equal to SYS3.BRODCAST. Note that DATASET is a required keyword.

volume-name specifies the volume serial on which the broadcast data set resides. VOLUME is an optional keyword.

time-out specifies the number of seconds a switch request will wait for resources before timing out. Valid values for TIMEOUT are integers in the range of 0 to 999, inclusive. TIMEOUT is an optional keyword. The default value is 5 seconds.

switch-prompt specifies whether TSO/E should prompt before switching the broadcast data set. Valid values for switch-prompt are PROMPT and NOPROMPT. switch-prompt is an optional keyword. The default value is PROMPT.

If the BROADCAST keyword is not specified, the default values are: SYS1.BRODCAST for data-set-name, no volume, five second time-out, and PROMPT for switch-prompt.

The TSO/E LOGON function allows you to log on if the broadcast data set has not been allocated.

IKJTSOxx parmlib member

You can specify the IKJTSOxx parmlib member on the IPL parameters. The new keyword is added to the SEND statement of IKJTSOxx.

The LOGNAME keyword on the SEND statement in IKJTSOxx is changed to allow specifying that user's mail should be kept in the broadcast data set by an asterisk (*). This can be used in place of SYS1.BRODCAST. Both SYS1.BRODCAST and * are used to indicate that mail should be kept in the current broadcast data set.

For compatibility reasons, if LOGNAME(SYS1.BRODCAST) is specified, the user's mail will go into the current broadcast data set even if this data set is not named SYS1.BRODCAST.

The name of the broadcast data set can now be specified in the IKJTSOxx parmlib member, as follows:

```
BROADCAST(DATASET(SYS1.ALT.BRODCAST)
  VOLUME(SBOX01) TIMEOUT(5) PROMPT)
```

New display ikjtso command

This new command is also enhanced to display the name and volume of the current broadcast data set; see Figure 3-1 on page 34.

```
D IKJTSO,SEND
IKJ738I TSO/E PARMLIB SETTINGS : 867
   SYS1.PARMLIB(IKJTSO00) on volume 037CAT
   Activated by **IPL** on 2002-08-24 at 11:52:23 from system SC65
   Applies to :    SC65
           THE FOLLOWING ARE THE PARMLIB OPTIONS FOR SEND:
           OPERSEND(ON)
           USERSEND(ON)
           SAVE(ON)
           CHKBROD(OFF)
           LOGNAME(BRODCAST)
           USEBROD(ON)
           MSGPROTECT(OFF)
           SYSPLEXSHR(ON)
           OPERSEWAIT(ON)
           USERLOGSIZE(1,2)
           BROADCAST(DATASET(SYS1.BRODCAST)
                     VOLUME(SBOX01) TIMEOUT(5) PROMPT)
```

Figure 3-1 Display ikjtso,send command output

New set ikjtso command

The new **set ikjtso=xx** command allows an installation to dynamically switch from the active TSO/E parmlib member to the specified IKJTSOxx member. This includes the ability to dynamically switch to a different cataloged broadcast data set.

When entering the **set ikjtso** command, and the IKJTSOxx parmlib member specifies a new broadcast data set, it is necessary to confirm the switch to a new broadcast data set, as follows:

```
set ikjtso=xx
```

The following message is issued to verify the switch.

```
*nn IKJ717A CURRENT BRODCAST DATA SET = current-data-set-name, VOLUME =current-volser,
NEW BRODCAST DATA SET = new-data-set-name, VOLUME = new-volser, REPLY YES TO SWITCH
```

```
r nn,YES
```

Migration actions

You can now optionally specify the particular IKJTSOnn member you want to use on the IKJTSO= parameter in IEASYSxx. The default is IKJTSO00. No change is required unless you want to use a member other than IKJTSO00. In IKJTSOxx, you specify the broadcast data set on the BROADCAST parameter of the SEND statement. TSO/E will default to SYS1.BRODCAST, so no action is required unless you want to use a different data set.

The MSTJCLxx member is no longer used to indicate the broadcast data set. Instead, you specify the broadcast data set in the IKJTSOxx member if you want to use a broadcast data set other than SYS1.BRODCAST. The master JCL will no longer allocate the broadcast data set. Instead, TSO/E will use either the default (SYS1.BRODCAST) or the BROADCAST parameter in IKJTSOxx to allocate the broadcast data set.

For more information, refer to *z/OS MVS Initialization and Tuning Reference,* SA22-7592 and *z/OS TSO/E General Information,* SA22-7784.

Archived

3.1.2 ASID resource relief and monitoring

Before z/OS V1R3, when an address space becomes non-reusable either permanently or temporarily due to cross-memory binds, the ASCB and ASSB remain allocated and queued to the memory delete queue. When an address space becomes reusable, the ASID is added back to the ASVT and then the ASCB and ASSB are freed. If a large number of address spaces become non-reusable, this can cause an excessive amount of SQA and ESQA to remain allocated for those spaces. This limits the value of specifying a large number for RSVNONR, as more that the 4-byte ASVT entry remains allocated in storage.

Once the number of non-reusable ASIDs reaches the limit, a re-IPL z/OS to reset the “non-reusable” count back to zero must be done. So as cross-memory subsystems (such as DB2 or MQ) get recycled over time, eventually a re-IPL is necessary. This new function removes the requirement to do periodic IPLs to reset the non-reusable ASID count back to zero.

To reduce storage impacts, both virtual and real, the ASCB and ASSB need to be freed as soon as an ASID becomes non-usable. A new protocol is defined between cross-memory and memory delete to allow this. Freeing these blocks minimizes the SQA and ESQA impacts that a non-usable ASID has on the system, and this will allow a large number of replacement ASIDs to be defined with minimal storage impact.

Memory delete now frees the ASCB and ASSB once cross-memory has accepted responsibility for the address space. If the address space is permanently non-reusable, due to a bind to a system LX, then the virtual storage impact of the lost ASID is limited to the ASVT entry. If the address space is potentially reusable when binds have terminated, then cross-memory adds the XMSE for the address space to a new reuse queue.

The reuse queue will be processed when an address space terminates. When all binds have terminated, cross-memory will remove the XMSE from the reuse queue and add the ASID back to the ASVT. The ASID reuse code will now exist in both memory delete and cross-memory, as both may make an ASID reusable.

Note: The default value for RSVNONR in the IEASYSxx parmlib member is increased to 100.

Resource monitoring

Monitoring of an ASID and linkage indexes (LXs) usage limits is now done on a timed basis every two minutes. When this limit is crossed, a message is issued. The limits for the resources that are monitored are fixed and cannot be changed.

There is nothing that can be done to extend the resources. In general the messages are warnings to key automation on, so that actions can be taken, but that probably means scheduling an IPL. Some resources may be able to be recovered by terminating the “proper” address space or correcting a “hang”, but most of the time there is not much that can be done.

New messages

The following messages about ASIDs and LX usage are issued when limits are crossed (as mentioned, the limits cannot be changed):

```
IEA059E ASID SHORTAGE HAS BEEN DETECTED

Explanation:  The number of ASIDs available for allocation to new address
spaces has dropped below 5% of the value specified in IEASYSxx with the
MAXUSER specification.
```

```
IEA060I ASID SHORTAGE HAS BEEN RELIEVED

Explanation:  The number of ASIDs available for allocation to new address
spaces has risen above 10% of the value specified in IEASYSxx with MAXUSER
specification.

IEA061E REPLACEMENT ASID SHORTAGE HAS BEEN DETECTED

Explanation:  The number of ASIDs available for replacing non-reusable
address spaces has dropped below 5% of the value specified in IEASYSxx
with the RSVNONR specification.

IEA062I REPLACEMENT ASID SHORTAGE HAS BEEN RELIEVED

Explanation:  The number of ASIDs available for replacing non-reusable
address spaces has risen above 10% of the value specified in IEASYSxx with
the RSVNONR specification.

IEA063E SYSTEM LX SHORTAGE HAS BEEN DETECTED

Explanation:  The number of system LXs available for allocation has
dropped below 15% of the value specified in the IEASYSxx with the NSYSLX
specification.

IEA065E NON-SYSTEM LX SHORTAGE HAS BEEN DETECTED

Explanation:  The number of non-system LXs available for allocation has
dropped below 15% of the number available for allocation. The number
available for allocation is 2048 minus the value specified in IEASYSxx
with the NSYSLX specification and any LXs reserved by IBM for internal
use.

IEA066I NON-SYSTEM LX SHORTAGE HAS BEEN RELEIVED

Explanation:  The number of non-system LXs available for reallocation has
risen above 30% of the number available for allocation.

IEA067I CROSS-MEMORY RESOURCE MONITORING HAS FAILED

Explanation:  A fatal error occurred in the cross-memory monitoring task.
Cross-memory resource monitoring will no longer occur until the system is
re-IPLed.
```

3.1.3 Page data set protection

Page data sets are protected with a two-level mechanism in z/OS V1R3. This provides protection for systems both within and outside of a common serialization scope. If multiple use of a data set is detected, both systems (assuming they both have the data set protection support) are not allowed to continue execution in order to protect system integrity. Instead, messages are issued with information about why integrity was lost and a wait state occurs.

Two levels of page data set protection are provided, as follows:

- The first is a SYSTEMS level ENQ. A GRS resource name with SCOPE=SYSTEMS is obtained as follows:
  - During master scheduler initiation for IPL specified data sets
  - Whenever a page data set is added or replaced after IPL

This is very effective at providing protection within a GRS Ring or GRS Star configuration or alternative serialization environments. It cannot protect against isolated systems. It provides the best protection because it is continual.

- The second is a signature record within the page data set, along with validation of the catalog information. While it does provide protection, its primary purpose is to allow incorrect usage by isolated systems to be detected. Since the information is checked on a periodic basis, a small period of time may elapse before the incorrect usage is detected.

GRS SYSTEMS level ENQ

Page data sets are now protected with a SYSTEMS level ENQ that contains the page data set name and the volume serial it resides on. For the protection to be effective, it is important that the data set name and volume serial be unique.

The ENQ is issued at **PAGEADD** or **PAGEDEL REPLACE** time, as well as by IDCAMS DEFINE and DELETE processing. This prevents use of a data set still being formatted, and the deletion of a data set that is in use. IDCAMS no longer checks the UCB to determine if it can delete a page data set, so inactive page data sets on the same volume as active data sets can be deleted.

Note: PAGEDEL processing will release the ENQ. Previously, the TSO **DEFINE** command did not have to be authorized to define a PAGESPACE. Now the **DEFINE** command must be defined as authorized for it to work, just as the **DEFINE RECATALOG** required it to be authorized.

Figure 3-2 Sysplex with three systems and page data sets

The Qname for the SCOPE=SYSTEMS ENQ is SYSZILRD, and the Rname is the dsname+volser. The dsname+volser must be unique for the page data set protection to work.

You can display the ENQ protection using a display GRS command, as shown in Figure 3-3 for the configuration shown in Figure 3-2 on page 37.

```
D GRS,RES=(SYSZILRD,*)
ISG343I 16.09.39 GRS STATUS 036
S=SYSTEMS SYSZILRD PAGE.SC63.COMMON1+SBOX98
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC63      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC63.LOCAL1+SBOX01
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC63      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC63.LOCAL3+SBOX47
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC63      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC63.LOCAL4+SBOX54
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC63      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC63.LOCAL5+SBOXA3
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC63      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC63.PLPA1+SBOX98
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC63      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC64.COMMON1+SBOX99
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC64      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC64.LOCAL1+SBOX01
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
S=SYSTEMS SYSZILRD PAGE.SC64.LOCAL2+SBOX22
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC64      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC64.LOCAL3+SBOX55
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC64      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC64.LOCAL4+SBOXA0
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC64      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC64.LOCAL5+SBOXA1
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC64      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC64.PLPA1+SBOX99
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC64      *MASTER*           0001       00FC8190 EXCLUSIVE    OWN
SC65      *MASTER*           0001       00FC4F18 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC65.LOCAL1+SBOX08
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC65      *MASTER*           0001       00FC4F18 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC65.LOCAL2+SBOX37
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC65      *MASTER*           0001       00FC4F18 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC65.LOCAL3+SBOX56
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC65      *MASTER*           0001       00FC4F18 EXCLUSIVE    OWN
S=SYSTEMS SYSZILRD PAGE.SC65.PLPA+SBOX08
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
SC65      *MASTER*           0001       00FC4F18 EXCLUSIVE    OWN
```

Figure 3-3 GRS display of page data sets

Data set signature record

A data set signature record containing status information is written to every page data set. The signature record is validated and updated every five minutes with an updated time stamp for every page data set. The signature record contains the following:

- An identifier indicating it is a signature record
- Which system is using the data set, which includes:
  - GMT IPL time stamp
  - GMT last updated time stamp
  - Processor token
  - LPAR name
  - VM userid, if appropriate

Error conditions and messages

The following conditions can occur and the appropriate action can be taken, as follows:

- If an incorrect or in-use signature is detected at IPL, the operator can override it and allow the system to use the data set.

  An in-use signature is one where the GMT last updated time stamp is within +/- 10 minutes of the current GMT time and the other identifying information, processor token, LPAR name, does not match.

  ```
  ILR030A PAGE DATA SET MAY BE IN USE:
  DATA SET NAME - dsname
  SYSTEM NAME - sysname
  CPU IDENTIFIER - mach-serial
  [LPAR NAME - lparname]
  [VM USERID - vmuserid]
  [IPL IS IN PROGRESS (TIME/DATE MAY NOT BE ACCURATE)]
  DATA SET LAST UPDATED AT hh:mm:ss ON mm/dd/yy (GMT)
  [                        hh:mm:ss ON mm/dd/yy (LOCAL)]
  ILR031A REPLY 'DENY' TO PREVENT ACCESS, 'CONTINUE' TO ALLOW USE OF dsname.
  ```

  Operator Response:
  If the data set is in use by the identified system, respond with DENY. If the data set may validly be used by this system, respond with CONTINUE. (Note, however, that a response of CONTINUE may cause a system failure if the data set is actually in use by the identified system.)

- The signature is incorrect if the identifier is not the expected value.

  ```
  ILR029I STATUS RECORD BEING FORMATTED FOR PAGE DATA SET dsname.
  ```

  The status information record in the identified page data set was not recognized. This may mean that the data set is currently being *use*d by a system that does not support page data set protection (this cannot be determined), or was *formatted* by a system that does not support page data set protection, or is being used by a system that does not support page data set protection.

  The status information record is formatted, and processing continues.

- A page data set is being used by another system.

  ```
  ILR032I PAGE DATA SET HAS BEEN USED BY ANOTHER SYSTEM:
  DATA SET NAME - dsname
  SYSTEM NAME - sysname
  CPU IDENTIFIER - mach-serial
  [LPAR NAME - lparname]
  [VM USERID - vmuserid]
  [IPL IS IN PROGRESS (TIME/DATE MAY NOT BE ACCURATE)]
  DATA SET LAST UPDATED AT hh:mm:ss ON mm/dd/yy (GMT)
  [                        hh:mm:ss ON mm/dd/yy (LOCAL)]
  ```

The identified page data set appears to have been used by a different system more than 20 minutes from the current time.

The status information record will be updated to indicate that this system is now using the data set, and processing continues.

- During an IPL, an ENQ cannot be obtained for a page data set.

  ILR033I UNABLE TO SERIALIZE IPL DEFINED PAGE DATA SET dsname.

  The system was unable to obtain an ENQ of SYSZILRD dsname.volser for page data sets defined during the IPL process.The page data set may be in use by another system.

- The system is unable to read the signature record.

  ILR034I PERMANENT I/O ERROR ON STATUS RECORD FOR PAGE DATA SET dsname.

  The system was unable to read or write the status information record.

  Status record checking for the identified data set is terminated. Protection against sharing by other systems for this data set has become limited until the system is reinitialized.

- The catalog cannot be accessed for the requested page data set.

  ILR038I PERMANENT CATALOG ERROR FOR THE PAGE DATA SET dsname

  The system was unable to read catalog information for the data set due to an unrecoverable catalog error.

  Status record checking for the identified data set is terminated. Protection against sharing by other systems for this data set has become limited until the system is initialized.

- The page data set cannot be accessed.

  ILR028I PERMANENT I/O ERROR OPENING PAGE DATA SET dsname.

  An uncorrectable I/O error occurred during the open process when trying to read or write the status information record from the identified page data set. This information is required for the data set to be considered usable.

  Open processing for the data set is terminated. The data set cannot be used.

Error messages with wait states

The following error conditions cause wait states to occur:

- Errors accessing the catalog are ignored. The catalog or status information record for the identified page data has been altered unexpectedly, and the integrity of the data in the data set records has been lost.

  ```
  ILR035W STATUS ERROR ON PAGE DATA SET:
  DELETED WHILE IN USE.
  DATA SET NAME - dsname

  -- WAIT 02E-07 --
  ```

  The catalog entry for the data set has been deleted using IDCAMS on another system.

  ```
  ILR035W STATUS ERROR ON PAGE DATA SET:
  CATALOG INFORMATION ALTERED.
  DATA SET NAME - dsname

  -- WAIT 02E-08 --
  ```

  The catalog data set extents, volume, device type, or page data set attributes have changed, which indicates that the data set has been deleted and defined using IDCAMS on another system.

  ```
  ILR035W STATUS ERROR ON PAGE DATA SET:
  RECORD HEADER DESTROYED.
  ```

```
DATA SET NAME - dsname
```

-- WAIT 02E-09 --

The status information record header has become unrecognizable, which indicates that the data set is probably in use by another system which does not have page data set protection support.

```
ILR035W STATUS ERROR ON PAGE DATA SET:
ALTERED BY ANOTHER SYSTEM.
DATA SET NAME - dsname
SYSTEM NAME - sysname
CPU IDENTIFIER - mach-serial
[LPAR NAME - lparname]
[VM USERID - vmuserid]
[IPL IS IN PROGRESS (TIME/DATE MAY NOT BE ACCURATE)]
DATA SET LAST UPDATED AT hh:mm:ss ON mm/dd/yy (GMT)
                         hh:mm:ss ON mm/dd/yy (LOCAL)
```

-- WAIT 02E-0A --

The status information record has been updated, which indicates that the data set is in use by another system.

3.1.4 Binder changes

The default for the Program Management COMPAT binder option, which lets you specify the compatibility level of the binder, has changed from CURRENT to the new option, MIN. The old default, CURRENT, specifies that the binder output is defined for the current level of the binder.

For example, a system running at the z/OS V1R3 level that specifies COMPAT=CURRENT will get the z/OS V1R3 PM4 program object format. The new default, MIN, specifies that the binder will choose the earliest format supporting all of the binder features in use.

This means that a system at the z/OS V1R3 level specifying or defaulting to COMPAT=MIN will get the program object format appropriate to the binder features in use, rather than the current one.

Archived

Archived

4

msys for Operations enhancements

This chapter describes and discusses the following:

- What Managed Systems Infrastructure for Operations (msys for Operations) is
  - How is msys for Operations delivered, what does it contain, and what is the value for you.
- How to implement msys for Operations step-by-step in your environment.

Note on terminology: msys for Operations, in this redbook (as well as in other IBM documentation) is referred to in many ways including: msys/Ops, msys for Ops, and Automation In the Base (AIB).

Archived

© Copyright IBM Corp. 2003

4.1 msys for Operations overview

z/OS Managed System Infrastructure for Operations (msys for Operations) is a base element of z/OS. Initially introduced in z/OS V1R2, it has been further extended in z/OS V1R3 (SPE UZ99415). The capabilities provide important functions for z/OS in the area of outage avoidance. msys for Operations addresses self-healing and self-managing qualities of the IBM autonomic computing initiative.

msys for Operations is an automation infrastructure within z/OS delivering functionality that targets system resources. It is based upon IBM's well-proven automation technology. Functions control and manage both hardware and software resources, thus making fully automated solutions possible. The focus is to simplify complicated operator interaction, to detect failure situations, and to react to them quickly and effectively. This is achieved through *panel-driven operator dialogs* and *automated recovery routines* that run in the background.

You decide which automated routines are permitted to run. A policy controls not only turning on and off background functionality, but also describes installation-specific details including:

- Naming preferences and volumes to be used for data sets that are dynamically allocated.
- How far a recovery process can go. Which jobs can be canceled?
- Critical resources that require special checking, and the frequency of such checks.
- The CPC/LPAR configuration required for hardware interaction functions.
- How LPARs should be treated during system failure isolation. Should the failed system image immediately be reloaded, or left in a reset status?

The objective of msys for Operations is to simplify day-to-day operational tasks associated with Parallel Sysplex and z/OS. This is achieved by reducing operator complexity, creating greater operational awareness of important indicators, and improving system recoverability. All of these factors are essential to z/OS availability, and in turn directly affect the performance and availability of your business applications.

Important: msys for Operations is not a product, nor will it be displacing any product. In many cases it will, however, remove the necessity for you to write your own automation-exploiting code to achieve control over specific system events.

Today, many installations write and maintain their own versions of this type of code. For the less sophisticated customer, this is virtually impossible. msys for Operations may be used to address these issues and may result in higher availability for your business applications.

By supplying msys for Operations, you can decide to enable and make use of the new capabilities or leave them disabled, continuing to use existing automation routines. Installations unable to develop their own code, or who have lost key automation skills and consequently are running unsupported code, are particularly interested in this concept. msys for Operations has very little coexistence conflict with existing automation products. Installations that enable msys for Operations functions must simply ensure that other, similar functions that run are *disabled*.

Note: msys for Operations *cannot* be extended or modified by you. Usage is locked, and any change will invalidate that part of the code.

It is also important to note that msys for Operations cannot automate the startup and shutdown of workloads and applications, nor correlate the registration of their dependencies on one another.

For more information on msys for Operations, refer to *Managed System Infrastructure for Operations Setting Up and Using,* SC33-7968, and *Managed System Infrastructure for Setup Installation Version 1 Release 4,* SC33-7997.

4.1.1 Foreground msys for Operations command dialogs

Foreground functional capabilities through msys for Operations command dialogs include the following:

- Display systems
- Display consoles
- Control Coupling Facilities
- Control couple data sets (CDSs)
- Display IPL information
- Control dumps

Let us review each of the command dialogs.

Display systems

Panels show sysplex systems and summarize current status, failure detection intervals, and SFM policy indicators for each of your systems. More details can be obtained by using a set of display commands that can be performed against any system by typing a single character against the system name.

Display consoles

Panels show a consolidated view of all consoles in the sysplex. The summarized display makes it possible to quickly identify the master console, assess buffer usage, show outstanding replies, connect console names to devices, and check the status, authority, owning system and mscope associated with the console. Details for each console and outstanding replies can be obtained by typing a single character against the appropriate console name. Replies to messages and other commands can be entered directly from this location.

Control Coupling Facilities

Panels show CFs in the Parallel Sysplex and summarize storage usage, volatility settings, system-managed process level and CFCC level. From this initial panel and subsequent panels, functions are performed by typing a single character against the appropriate CF name. From here CFs can be removed from service and reintroduced into service. Sender paths can be displayed and controlled (online/offline), and CF structures can be completely managed through panel interaction. It is a simple task to display details for a specific structure, rebuild it, force it, and start or stop duplexing.

This area of focus can cause serious impact to availability if operational errors are made. msys for Operations helps you address command complexity, as well as control critical sequences of execution, which helps ensure that errors are avoided. Interaction with the hardware enables CFs to be activated and deactivated without manual action at the Hardware Management Console (HMC). Another benefit is that it is easy to distinguish between an operational and non-operational CF because the color of the CF Icon changes at the HMC.

Control couple data sets (CDSs)

Panels show CDSs in your Parallel Sysplex. The information displayed gives a quick overview of some key sysplex settings. The COUPLExx member in use is identified and for each CDS, details about the type, maximum systems supported, volume and device on which the CDS is allocated, and the data set name is displayed. Any anomalies such as missing channel paths

or alternate CDS, or where the primary and alternate CDS are allocated on the same volume, device or storage controller, are raised as alerts as they represent single points of failure. From this initial panel and subsequent panels, functions are initiated by typing a single character against the appropriate CDS Type.

In this way alternate CDSs can be dynamically allocated and brought into service; details can be displayed for every system about the specific CDS device, channel paths and storage controllers; alternate CDSs can be switched to become the primary CDS; input parameters used in creating the CDS can be displayed; policies can be displayed, the active policy identified, and new policies can be started.

Display IPL information

Panels show IPL summary details for all systems in your Parallel Sysplex. From the information displayed you can quickly ascertain when a particular system was IPLed, the system residence volume/device, and the operating system release. From this initial panel and subsequent panels, functions are initiated by typing a single character against the IPL record of choice.

Details can be viewed, comparisons can be run against different records, and records can be erased. The detailed information includes the load parameters used for the particular IPL, CPC/LPAR name that was IPLed, active IODF, master catalog and every PARMLIB member identified that was used to initialize the system. Each member can be displayed, and comparisons can be run to identify differences.

Control dumps

This panel is used to control SDUMP options, SLIP Trap settings and issue multi-system SVC dumps, including any data spaces and structures across the Parallel Sysplex. From this initial and subsequent panels, functions are initiated by typing single characters and using function keys.

4.1.2 Background msys for Operations automated recovery routines

The background automated recovery routines address the following:

- No alternate couple data set condition
- Console buffer shortage recovery
- Logstream data set directory shortage recovery
- Logstream share options checks
- System log recovery
- Capturing of detailed IPL information
- Elimination of long ENQs
- Auxiliary storage shortage recovery
- Isolation of a failed system
- Scheduling SVC dumps for certain conditions
- Rebuild CF structures affected by a CFRM policy switch
- Direct interaction with the hardware

4.1.3 msys for Operations business value

Whether interest is in the panel-driven operator dialogs that assist in the control of CDSs, CFs and CF structures—or in the background recovery routines that guard against console buffer shortages, long-running enqueues or auxiliary storage shortages—z/OS customers find msys for Operations benefits operations in their organizations and leads to improved availability.

4.2 msys for Operations implementation checklist

The following summarizes the customization steps necessary to implement msys for Operations, including SPE UW99415. These activities need to be performed on each system following the installation. If the same PROCLIB and PARMLIB data sets are shared among the participating systems, then changes need only be made once on any system and will be applicable to every system.

The customization is organized into 12 steps as shown in Appendix A, “msys for Operations implementation checklist” on page 347.

Some of the steps are optional, as described in the following sections:

- “Step 1: Create VSAM and non-VSAM data sets” on page 348.
- “Step 2: Copy additional PROCs into PROCLIB data set” on page 354.
- “Step 3: Data sets for LNKLST and LPALST” on page 356.
- “Step 4: Add a PPT entry” on page 358.
- “Step 5: Update MPFLST” on page 359.
- “Step 6: Define application major nodes for VTAM” on page 361.
- “Step 7: Make determined security definition changes” on page 362.
- “Step 8: Alter msys for Operations NVSS style sheet” on page 371.
- “Step 9: Enable msys for Operations functions” on page 373.
- “Step 10: Build the VTAM logon mode table AMODETAB” on page 383.
- “Step 11: REXX environment table entries” on page 384.
- “Step 12: Perform hardware customization on SEs” on page 387.

Archived

Archived

5

JES3 Version 1 Release 4 enhancements

This chapter describes the enhancements made to JES3 Version 1 Release 4:

- JES3 MAINPROC refresh
- New Inquiry command, `*I MAIN=`
- Main processor status enforcement
- JES3 checkpoint protection

Archived

© Copyright IBM Corp. 2003

5.1 JES3 enhancements

There is a need to be able to add processors to a sysplex without requiring a sysplex-wide outage, which requires a JES3 warm start. This is an inhibitor to sysplex growth. z/OS JES3 V1R4 provides the ability to add, change, and delete a MAINPROC statement without a warm start, which is referred to as JES3 MAINPROC refresh. In most cases, a hot start with refresh without an IPL is sufficient, although an IPL is optional and in a few special cases it is required on individual processors.

In cases where an IPL is required, JES3 does not just ask the operator to reset the processor and reply when done. JES3 uses the JESXCF information service to find out which systems are still active and gives the operator a message about particular processors that need to be IPLed and waits until this happens. JES3 uses this procedure elsewhere when IPLs are required to improve spool integrity.

A command is provided for inquiry on MAINPROCs so that the systems programmer can examine the values in effect and change them on a hot start with refresh if they are not what were expected; see “New Inquiry command” on page 55.

5.1.1 JES3 MAINPROC refresh

This new support is to allow sysplex growth without disruption. Since each member of the sysplex must be defined in the initialization stream as a MAINPROC statement, a warm start is required in order to add MAINPROCs. Therefore, JES3 installations cannot grow their JESplex without causing a JESplex-wide outage. This restriction is being removed with JES3 V1R4.

OS/390 JES3 V2R9 created better flexibility by allowing a processor name of *ALL to be specified on the DEVICE statement instead of requiring processors to be listed out. This made it easier to add not just devices but also processors, because a device defined with *ALL does not need to have its JUNIT or XUNIT changed again no matter how many MAINPROCs are added or deleted. An installation that defines every device with *ALL therefore has a workaround.

MAINPROC is a statement in the JES3 initialization stream that defines a processor. There can be up to 32 MAINPROC statements in the JES3 initialization stream. A processor can be global or local, but there is only one global per JESplex.

There is no parameter on the MAINPROC statement that makes a processor become a global or local; it is all dependent on how the operator initializes the JESplex. A processor becomes global when the operator cold or warm starts JES3 on it. A local processor can also become global through the Dynamic System Interchange (DSI).

New MAINPROC support

The MAINPROC support in JES3 V1R4 is as follows:

- MAINPROC statements can be added or deleted at the end of the list during a hot start with refresh.

  The order of the existing processor names that are carried over cannot change. If the order of the existing MAINPROC statements is SC63, SC64, and SC65, as shown in Figure 5-1 and Figure 5-2 on page 51, this order cannot be changed at any time and a new MAINPROC must be added at the end.

Figure 5-1 Sample JES3 configuration

Note: Although z/OS 1.4.0 can be a local coexisting with a lower JES3 release, the global must be at z/OS 1.4.0 in order to make use of this support.

```
MAINPROC,NAME=SC63,...
MAINPROC,NAME=SC64,...
MAINPROC,NAME=SC65,...
```

Figure 5-2 Existing MAINPROC statements

- Any parameter on an existing MAINPROC statement can be changed during a hot start with refresh.

  Changing the NAME= parameter is called “renaming” a processor and is subject to the restriction that the new name must not already exist, as this would cause an order change.

- Many syntax errors on the MAINPROC statement are ignored and now allow initialization to continue instead of failing.

  In most cases defaults are provided; in other cases, a statement is ignored. In particular, if a NAME= is missing, the entire statement is ignored with a warning message. If a statement is ignored and you fix it later, you must move it to the *end* of the list; otherwise, JES3 thinks you are trying to add it in the middle.

- A deleted processor must be reset. JES3 will not allow you to delete an active processor.

Legal specifications for MAINPROCs

In the following figures, the MAINPROC names use the names of fruits to make it easier to distinguish between MAINPROC system names.

Archived

Figure 5-3 Legal specifications of adding and deleting MAINPROC statements

Figure 5-3 shows examples of legally adding or deleting of MAINPROC statements, as follows:

1 This is an example of an addition of a single MAINPROC at the end. A change must be made with the global being either on APPLES or ORANGES, and then GRAPES can be brought up as a local. To add GRAPES and make it global at the same time requires a warm start. To avoid the warm start, add GRAPES as a local and if you want GRAPES to be the global, then perform a DSI to make GRAPES the global.

2 This is an example of a deletion from the end. GRAPES must be reset. This change is not allowed if GRAPES is the current global.

3 This is an example of a rename of ORANGES to BANANAS, and an addition of PEACHES at the same time. ORANGES must be reset. The change is not allowed if ORANGES is the current global.

4 This is an example of returning to the previous example's From: condition, so that it is a simultaneous rename and deletion. BANANAS and PEACHES must be reset. This change is not allowed if BANANAS or PEACHES is the current global.

Illegal specifications for MAINPROCs

Figure 5-4 on page 53 shows examples of illegally adding or deleting MAINPROC statements.

From:
1 MAINPROC,NAME=APPLES,...
MAINPROC,NAME=ORANGES,...
MAINPROC,NAME=GRAPES,...

To:
MAINPROC,NAME=APPLES,...
MAINPROC,NAME=ORANGES,...
MAINPROC,NAME=PEACHES,...
MAINPROC,NAME=GRAPES,...

From:
2 MAINPROC,NAME=APPLES,...
MAINPROC,NAME=ORANGES,...
MAINPROC,NAME=GRAPES,...

To:
MAINPROC,NAME=APPLES,..
~~***MAINPROC,NAME=ORANGES,..***~~
MAINPROC,NAME=GRAPES,...
MAINPROC,NAME=PEACHES,...

From:
3 MAINPROC,NAME=APPLES,...
MAINPROC,NAME=ORANGES,...
MAINPROC,NAME=GRAPES,...

To:
~~***MAINPROC,NAME=APPLES,...***~~
MAINPROC,NAME=ORANGES,...
MAINPROC,NAME=GRAPES,...

Figure 5-4 Illegal combinations of adding and deleting MAINPROC statements

1 This is illegal because it is an attempt to add PEACHES before GRAPES, rather than at the end of the MAINPROC sequence.

2 This is illegal because although PEACHES is being added at the end, it is an attempt to move GRAPES from the third processor to the second. In other words, it is an attempt to rename ORANGES to GRAPES and move GRAPES elsewhere at the same time.

3 This is illegal because it is an attempt to delete APPLES from the beginning of the MAINPROC sequence, which causes the sequencing of ORANGES and GRAPES to be changed at the same time.

MAINPROC statement parameters

Now that you can change or add MAINPROC statements on a JES3 hot start with refresh, you must consider the following parameters if they are changed:

- Changes to the PRTPAGE or FIXPAGE parameter on any processor requires that processor to be re-IPLed to pick up the changes.
- The global cannot add, delete, or rename itself. If you IPL a processor that is not defined in the initialization stream, either you must warm start it, or you must hot start with refresh another processor, and then bring up the processor in question as a local.

> **Restriction:** If a MAINPROC change of any kind is done, and there are processors lower than z/OS 1.4.0, those processors must be re-IPLed. This is necessary because downlevel processors do not have the new support code and need to be IPLed in order to rebuild the MAINPROC chain. If this were not done, MAINPROC chains become out of synch. Therefore, a DSI to a down level main would lose all MAINPROC changes that were made on z/OS 1.4.0 during a hot start with refresh.

Statement examples

In addition to adding and deleting, existing MAINPROCs can have individual parameters changed, as shown in the examples in Figure 5-5 on page 54.

```
From:
1 MAINPROC,NAME=APPLES,FIXPAGE=5,...
  MAINPROC,NAME=ORANGES,...
To:
  MAINPROC,NAME=APPLES,FIXPAGE=10,...
  MAINPROC,NAME=ORANGES,...

From:
2 MAINPROC,NAME=APPLES,
    JESMSGLMT=(100000,100),...
  MAINPROC,NAME=ORANGES,...
  MAINPROC,NAME=GRAPES,...
To:
  MAINPROC,NAME=APPLES,
    JESMSGLMT=(100000,200),...
  MAINPROC,NAME=ORANGES,...
  MAINPROC,NAME=GRAPES,...
```

Figure 5-5 Examples of individual parameter changes on the MAINPROC statement

1 If APPLES has not been re-IPLed, message IAT3423 is issued. APPLES may be global or local. If APPLES is a local, IAT2061 and IAT2064 are issued. If APPLES is a global and has not been IPLed, a DM026 occurs and the hot start with refresh fails. JES3 can be hot started to revert to the previous configuration, or the operator can reset APPLES and retry the hot start with refresh.

2 This is a MAINPROC change that does not inherently require the changed processor to be IPLed. However, an IPL might be needed in the case of mixed JES3 releases. For example, assume GRAPES is a local at z/OS V1R2. GRAPES must be re-IPLed. Messages IAT3426, IAT2061, and IAT2064 are issued.

Installation considerations

There are several considerations that you need to be aware of when adding, deleting, or renaming processors. There are other statements that refer to the MAINPROC statements. Some of these statements will also need to be changed, as follows:

- You should use the special system name, *ALL, as using this name means you will never have to change that parameter again no matter how many MAINPROC statements you add or delete. Specify this parameter on the following statements and parameters:
  - DEVICE, ...JUNIT/XUNIT parameters
  - GROUP,..... EXRESC parameter,

  Use of *ALL eliminates having to specify the system name, as follows:

  Before JES3 V2R9:

  ```
  DEVICE,DTYPE=TA435901,JNAME=TA0B9A,JUNIT=(0B9A,SC63,S1,OFF,
    0B9A,SC64,S1,OFF,0B9A,SC65,S1,OFF),XTYPE=(D13590,CA),
    XUNIT=(0B9A,SC63,S1,OFF,0B9A,SC64,S1,OFF,0B9A,SC65,S1,OFF)
  ```

  With JES3 V2R9 and later releases:

  ```
  DEVICE,DTYPE=TA435901,JNAME=TA0B9A,JUNIT=(0B9A,*ALL,S1,OFF),
         XTYPE=(D13590,CA),XUNIT=(0B9A,*ALL,S1,OFF)
  ```

  Therefore, when you add, delete, or change system names on the MAINPROC, you do not have to change anything on all the DEVICE statements.

- The DEVICE DTYPE=SYSMAIN statement was made optional in OS/390 JES3 V2R9. You only need to specify it if you want some mains to come up offline. Here too, if you cannot omit the statement, you should use the *ALL parameter on the JUNIT parameter.

Important: Look for other initialization statements that have system names (including the following: FSSDEF, MSGROUTE, GROUP, CLASS). You might have to change these also.

New ABEND code

The new ABEND code is DM026. It indicates that there was an error validating MAINPROC changes during a hot start with refresh. The reason codes are as follows:

1. Sequence changes because MAINPROC was not added or deleted at the end.
2. Attempt to delete the global.
3. Error occurred while validating the state of a deleted processor.
4. Reset of deleted processor requested, but operator replied CANCEL.
5. Error occurred while validating the state of downlevel processors.
6. IPL of downlevel processor requested, but operator replied CANCEL.
7. Error occurred while validating the state of a processor on which PRTPAGE or FIXPAGE was changed.
8. IPL required of a processor on which PRTPAGE or FIXPAGE was changed, but operator replied CANCEL.
9. PRTPAGE or FIXPAGE changed on the global without an IPL.

5.1.2 New Inquiry command

A new command, **`*I,MAIN=`**, is provided to inquire on MAINPROCs. This command has several purposes. In order to make it easy to test whether additions, deletions, and changes to MAINPROC statements have been correctly defined, the **`*I,MAIN`** command has the following forms:

***I MAIN=name** Provides brief information about a specific main, all mains (MAIN=ALL), or the global (MAIN=JGLOBAL).

***I MAIN=name,X** Provides extended information about a specific main, all mains (MAIN=ALL), or the global (MAIN=JGLOBAL).

For the command examples, the configuration shown in Figure 5-1 on page 51 is used.

New command considerations

There is already an **`*I,S`** command that tells you which processors are online and connected (IPLed), but if you are running without JES3 SETUP, the *I,S command is rejected, despite the fact that the processor status information might have been useful. For this reason, the **`*I,MAIN`** command is always available, even with SETUP off.

Furthermore, the **`*I,S c`**ommand has always been limited in the amount of main-related information it displays. The **`*I,MAIN`** command displays every parameter that is defined on the MAINPROC statement. It also identifies the JES3 release running on that processor and tells you if the processor is a global or local.

The **`*I,MAIN`** command accepts an individual system name, as well as various special system names, such as ALL for all mains, or JGLOBAL for the global. The **`*I,S`** command has only one form that lists all mains.

Display processor status

The ***I,MAIN=** command always displays the following brief information, as shown in Figure 5-6:

- The FMID of the processor, provided it is attached to JESXCF
- Whether the processor is online or offline
- Whether the processor is connected, not connected, or flushed
- Attach status to JESXCF
- Whether the processor is global or local

The ***I,MAIN=** command provides every piece of non-SETUP-related information that the ***I,S** command provides, plus a lot of other information. However, the text CONNECTED/NOT-CONNECTED is used instead of IPLD/NOTIPLD as the ***I,S** command uses. This is because “IPLD” is a confusing term—it has sometimes been incorrectly assumed to mean that the processor was previously up and was re-IPLed, when it in fact it means that the processor is connected right now.

```
*I MAIN=SC65
 IAT8643 MAIN INQUIRY RESPONSE
 INFORMATION FOR MAINPROC SC65
   FMID=HJS7707, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL)
 MAINPROC INQUIRY RESPONSE COMPLETE
```

Figure 5-6 New inquiry command on main processors

Display processor status with extended information

The extended command, ***I MAIN=name,X**, displays the brief information plus the following extended information, as shown in Figure 5-7:

- JES3 product level
- JES3 service level
- Message prefix identifier
- Message destination
- Select mode
- Spool partition
- Primary and secondary track group allocation
- Message limit and interval
- Number of pages fixed at initialization time
- Number of pages in CSA and JES3AUX
- Number of pages used for open SYSOUT data sets

```
*I MAIN=SC65,X
 IAT8643 MAIN INQUIRY RESPONSE
 INFORMATION FOR MAINPROC SC65
   FMID=HJS7707, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL), PLEVEL=15,
   SLEVEL=00, ID='R= R=', MDEST=M2, SELECT=SELA, SPART=NONE,
   TRKGRPS=(1,1), JESMSGLMT=(00000000,00010), FIXPAGE=00005,
   PRTPAGE=(00025,00000), USRPAGE=00004
 MAINPROC INQUIRY RESPONSE COMPLETE
```

Figure 5-7 New Inquiry command with extended information

Information description

The FMID specified is:

FMID The FMID specified is the JES3 FMID that the main had at the time of the last JESXCF attach. JESXCF remembers this value even after the main goes down.

PLEVEL This is the JES3 product level used internally by JES3, and it is displayed for diagnostic purposes. (FMID is probably a more useful display value.)

SLEVEL This is the JES3 service level used internally by JES, and it is displayed for diagnostic purposes. (FMID is probably a more useful display value)

The status= information, as shown in Figure 5-7 on page 56, is as follows:

ONLINE This status could be ONLINE or OFFLINE. This is the online or offline status of the main processor. It is the same as in the corresponding text in the `*I,S` command response.

CONNECTED This field can be CONNECTED, NOT-CONNECTED, or FLUSHED. This means the same as IPLD/NOTIPLD/FLUSHED in the `*I,S` command response.

ATTACHED This field can be ATTACHED, NOT-ATTACHED, or DOWN. This field is the JESXCF attach status. For more detail, see “Attached status” on page 57. NOT-ATTACHED means the JESXCF connection has been broken by bringing the JES3 address space down.

DOWN means either the processor has not attached to JESXCF (via a JES3 start), and there is no FMID—or the processor has been reset or gone down hard since the last global JES3 start and the FMID appears in the information.

GLOBAL Specifies whether this main processor is a LOCAL or a GLOBAL.

Attached status

The purpose in providing the JESXCF status, in addition to the online and connected status, is to highlight when someone has started JES3 and received an IAT3100 message, but forgotten to `*S JSS` (in the global case) or vary the local online (in the local case).

> **Important:** If an operator forgets to issue the `*S JSS` command, this status information indicates that JES3 is ATTACHED but NOT CONNECTED.

Other command options

The remaining options are to display all the main processors, as shown in Figure 5-8, or to display just the global processor.

```
*I MAIN=ALL
 IAT8643 MAIN INQUIRY RESPONSE
 INFORMATION FOR MAINPROC SC64
   STATUS=(ONLINE,NOT-CONNECTED,DOWN,LOCAL)
 INFORMATION FOR MAINPROC SC63
   STATUS=(ONLINE,NOT-CONNECTED,DOWN,LOCAL)
 INFORMATION FOR MAINPROC SC65
   FMID=HJS7707, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL)
 MAINPROC INQUIRY RESPONSE COMPLETE
```

Figure 5-8 New command to display all JES3 systems

The new command provides information that the ***I,S** command does not provide. In particular, it shows the JES3 release and which system is the current global. Previously, the only way to display the global processor was to issue a ***I,D,D=main** command for all the mains you have and then look for the one that results in the following message:

```
IAT8562 xxxxxxxx THE GLOBAL.
```

Display which main processor is the global

The new command can specifically only display the current global processor, as shown in Figure 5-9.

```
*I MAIN=JGLOBAL
IAT8643 MAIN INQUIRY RESPONSE
 INFORMATION FOR MAINPROC SC65
   FMID=HJS7707, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL)
 MAINPROC INQUIRY RESPONSE COMPLETE
```

Figure 5-9 New command to determine JES3 global

Displaying multiple mains

The commands ***I,MAIN=(main1,main2...mainn)** and ***I,MAIN=(main1,main2,...,mainn),X** are also allowed and provide information about all the listed mains, but like many other inquiry commands, the list inside the parentheses is broken up and processed internally as individual inquiry commands. Therefore, each individual processor is listed by separate complete MLWO responses, whereas the result of an ***I,MAIN=ALL** is one large MLWO response.

5.1.3 Main processor status enforcement

JES3 V1R4 now requires that a processor must be reset or re-IPLed by making sure that the operator actually brings the processor down. During this new check, JES3 figures out which processors need to be brought down and does not proceed until that happens. This is done at the following times:

- During a warm start, JES3 uses the status enforcement to identify all processors that need to be reset, and stops the warm start until the operator resets them. There is no longer the opportunity for the operator to reply that a processor is “DOWN” and allow an active processor to remain up.

  Note: This eliminates the possibility of having two globals through an operator error.

- JES3 now also makes sure that a local being flushed by the ***S,main,FLUSH** command is brought down by checking that the processor is really reset.
- When a job number is deleted by a reduction in range and an active job is deleted, JES3 makes sure that the processor on which the job is active is brought down.
- Processor status enforcement is done during a dynamic system interchange (DSI). When invoking DSI on a local to be made the global, JES3 makes sure that the old global is brought down. As a result of this checking, message IAT0910 is not issued if the global is already down, and so the operator procedure for IAT0910 is simplified. The ***S DSI** command is no longer needed at this point.

JES3 hot start with refresh

Figure 5-10 on page 59 is an example of a hot start with refresh in which a processor named GRAPES is being deleted. JES3 does not allow the hot start with refresh to proceed until GRAPES has been reset.

```
IAT3040 STATUS OF JES3 PROCESSORS IN JESXCF GROUP NODE1
IAT3040 APPLES    <UP>, ORANGES   (UP), GRAPES    (UP)
IAT3011 SPECIFY JES3 START TYPE
36 IAT3011 (C, L, H, HA, HR, HAR, W, WA, WR, WAR, OR CANCEL)
r 36,hr
37 IAT3012 SELECT JES3 INISH ORIGIN (N OR M=), AND OPTIONAL EXIT
PARM (,P=) OR CANCEL
r 37,m=2a
...
IAT3424 DELETED SYSTEM GRAPES MUST BE RESET
IAT2061 SYSTEM GRAPES  IS ACTIVE IN JESXCF GROUP NODE1
38 IAT2064 RESET ALL SYSTEMS SHOWN OR REPLY CANCEL
...
```

Figure 5-10 Main processor GRAPES is being deleted on a hot start with refresh

The operator can reply CANCEL, which causes the hot start with refresh to abend. The operator can then hot start to return to the previous configuration.

Warm start example

With this new support for status enforcement, and with the warm start requirement that all locals be reset and re-IPLed, JES3 does not allow the warm start to proceed until the following happens, as shown in Figure 5-11:

- Message IAT3046 does not appear any longer.

  Message IAT3046 is issued to remind the operator that no other JES3 processor may be operating while a global cold start or warm start is in progress. Otherwise, any resulting queue destruction may require a cold start.

  This message may be issued, when other processors are not in operation, if the status information for the processor was not updated when it ended processing.

  Note: The operator response to this message was to Reply DONE to confirm that no other JES3 processor is active in the complex, or Reply CANCEL to end JES3. Any other reply will cause message IAT3011 to be reissued. This reply is no longer possible with the new enforcement.

- Message IAT2064 replaces the manual confirmation that IAT3046 used to require, as shown in Figure 5-11.

```
IAT3040 STATUS OF JES3 PROCESSORS IN JESXCF GROUP NODE1
IAT3040 APPLES    <UP>, ORANGES   (UP), GRAPES    (UP)
IAT3011 SPECIFY JES3 START TYPE
36 IAT3011 (C, L, H, HA, HR, HAR, W, WA, WR, WAR, OR CANCEL)
r 36,w
37 IAT3012 SELECT JES3 INISH ORIGIN (N OR M=), AND OPTIONAL EXIT
PARM (,P=) OR CANCEL
r 37,m=00
...
IAT2061 SYSTEM GRAPES     IS ACTIVE IN JESXCF GROUP NODE1
IAT2061 SYSTEM ORANGES  IS ACTIVE IN JESXCF GROUP NODE1
38 IAT2064 RESET ALL SYSTEMS SHOWN OR REPLY CANCEL
...
```

Figure 5-11 JES3 warm start and status enforcement of the main processors

Operator flush of a local

When an operator issues a **`*s,main,flush`** command, the main processor enforcement checks to make sure that the processor is really reset.

In previous releases, message IAT2626 would appear to warn the operator that the local must be reset, and ask for a second FLUSH command to confirm the request. Message IAT2626 no longer appears and there is no need to enter a second flush command. The confirmation is the act of resetting the local in question. By replying CANCEL, the operator would cancel the FLUSH command, whereas in previous releases the operator would cancel the request by issuing some other command instead of the second FLUSH command.

```
*s,sc65,flush
IAT2061 SYSTEM SC65      IS ACTIVE IN JESXCF GROUP NODE1
61 IAT2064 RESET ALL SYSTEMS SHOWN OR REPLY CANCEL

...Operator resets GRAPES and some XCF messages are generated and replied to...

IAT2628 '*S SC65,FLUSH ' ACCEPTED
IAT2006 PREMATURE JOB TERM - JOB  SYSLOG   (JOB00182) - CANCELED     - SC65
IAT2006 PREMATURE JOB TERM - JOB  RACF     (JOB00185) - CANCELED     - SC65
```

Figure 5-12 Operator command to flush a main processor

Using JCT utility IATUTJCT

There is a new message related to processor status when running the JCT IATUTJCT utility. This message is not used to enforce that a system must be down, but provides true status, rather than “last known” status.

The old IAT7787 reply is still needed because IATUTJCT might be a test of the utility which is allowed when JES3 is active. However, the IAT7786 message has been replaced by the “true status” message IAT2062, as shown in Figure 5-13.

```
s iatutjct,sub=mstr
IAT2062 JES3 IS ACTIVE ON SYSTEM SC65 IN JESXCF GROUP NODE1
*17 IAT7787 CONFIRM SYSTEM STATUS AND REPLY CONTINUE OR  CANCEL
```

Figure 5-13 New message IAT2062 when using the IATUTJCT utility

Note the difference between IAT2062 here and IAT2061 shown in Figure 5-12 on page 60. The issue is whether JES3 is active, not whether the *processor* is active.

Message IAT2062

The JES3 address space on a processor (system) in the JES3 complex is still running. For certain JES3 functions, further processing of the requested JES3 function requires that the JES3 address space be brought down, in which case this message is highlighted until you either end JES3 on the system shown or reply CANCEL to message IAT2065.

For other JES3 functions, this message is informational and requires you to confirm the state of the JES3 complex with respect to the specific environment of the JES3 function that caused this condition to be detected. It is possible that more than one processor will be affected in this manner. If this is the case, a separate IAT2062 message is issued for each processor.

Note: The operator should respond as follows: If the message is highlighted, bring the JES3 address space on the processor in question down by issuing the **`*RETURN`** command. If you prefer, you can disable the processor completely as described in the operator response for message IAT2061. If you perform either of these actions, no further reply to message IAT2065 is required.

If you prefer that the requested JES3 function not continue, reply CANCEL to message IAT2065. If the message is not highlighted, refer to other messages issued by the particular JES3 function informing you of your options.

JES3 DSI processing

Another check of the processor status enforcement is during a DSI. When invoking DSI on a local to be made global, JES3 makes sure that the old global is brought down. As a result of this checking, message IAT0910 is not issued if the global is already down, and so the operator procedure for message IAT0910 is simplified. The **`*S DSI`** command is no longer needed at this point.

Figure 5-14 is an example of processor status enforcement with dynamic system interchange, and shows the messages issued before this change in JES3 V1R4. The following changes to DSI processing can be used in this new support:

- The **`*s dsi`** command is no longer necessary after message IAT0910, but if it is issued, it is ignored and messages IAT0927 and IAT0910 are reissued. A **`*c dsi`** command is still allowed here.
- If global is already down, JES3 proceeds to the IAT0900 message without issuing the IAT0927 and IAT0910 messages.

```
*x dsi
SC64 JES3     *IAT0915 DSI - REVIEW LOCAL DSI PROCEDURE FOR SC64
*s dsi
SC64 JES3      IAT0927 OLD GLOBAL SC65  IS STILL ACTIVE
SC64 JES3     *IAT0910 DSI - DISABLE OLD GLOBAL PROCESSOR
*s dsi
SC64 JES3      IAT0927 OLD GLOBAL SC65  IS STILL ACTIVE
SC64 JES3     *IAT0910 DSI - DISABLE OLD GLOBAL PROCESSOR
*x dsi
IAT0920 DSI - CHECK GLOBAL DSI PROCEDURE FOR SC65
*s dsi
IAT0905 DSI - STARTED FOR SC65
SC64 JES3      IAT7124 DLOG IS NOW INACTIVE
*SC64 JES3     *IAT0900 DSI - SWITCH GLOBAL DEVICES
```

Figure 5-14 DSI processing with processor enforcement

5.1.4 JES3 checkpoint protection

JES3 is providing support for the JES3 checkpoint data set in V1R4 to avoid the possibility of forcing the operator to re-IPL or bring down a processor when a mistake is made in the JES3 procedure when defining one system to point to another system's checkpoint without being in the same JESXCF group. This can typically happen when a production system is “cloned” into a test system and someone forgets to change the JES3 procedure.

This has been an almost guaranteed disaster unless the mistake is seen and corrected. With JES3 V1R4, there is at least a chance that this new support can catch the mistake shown in Figure 5-15 on page 62.

Figure 5-15 JES3 procedures for two different JESXCF groups

In Figure 5-15, system SCTEST is being IPLed, with the intention being to warm start it. As soon as JES3 starts, the initialization sees the checkpoint (which is the wrong one) and sees that SCTEST is not in the checkpoint status record (CSR). Message IAT3099, shown in Figure 5-16, is immediately issued.

At this point, it is realized that there is a functioning global called SC65 and processing cannot continue until they bring down SC65. Message IAT2061, which identifies both SC65 and the JESXCF group, is a sure sign that mistake has been made. Message IAT2061 is issued to indicate that you should disable (reset) main processor SC65.

Important: The operator should immediately see that SCTEST is not in JES3GRP, and that SCTEST is supposed to be in JESXCF group TEST. Therefore, the operator should reply CANCEL.

The proper reply of CANCEL kills SCTEST but protects SC65 (r 10,cancel).

This, of course, depends on SC65 being up. If by chance SC65 is down, processing continues without the first two messages shown in Figure 5-16. Processing continues until message IAT3040 appears, listing the processor status.

However, as of JES3 V1R4, message IAT3040 also displays the JESXCF group name, so there is still a chance the operator will notice the mistake before the reply to message IAT3011 for the JES3 start type.

```
IAT3099  SCTEST IS NOT DEFINED, COLD OR WARM START HERE OR HOT START WITH REFRESH ON
SC65 REQUIRED
IAT2061 SYSTEM SC65     IS ACTIVE IN JESXCF GROUP JES3GRP
*10   IAT2064 RESET ALL SYSTEMS SHOWN OR REPLY CANCEL
.........................
IAT3040 STATUS OF JES3 PROCESSORS IN JESXCF GROUP JES3GRP
IAT3040 SCTEST    <UP>, SC65      (UP)
```

Figure 5-16 JES3 initialization messages during an IPL

5.2 JES3 BCP compatibility

While the four-consecutive-release policy also applies to JES3, the way in which four consecutive releases is determined is different from the rest of the operating system. If a JES3 release is functionally equivalent to its predecessor (its FMID is the same, as shown in Figure), then from a coexistence-migration-fallback standpoint, the release is considered to be the same JES3 release.

Release	FMID	BCP Req	Notes
OS/390 V2 R8	HJS6608	OS/390 V2 R8	1
OS/390 V2 R9	HJS6609	OS/390 V2 R9	2
OS/390 V2 R10	HJS7703	OS/390 V2 R10	3
z/OS V1 R1	HJS7703	z/OS V1 R1	4
z/OS V1 R2	HJS7705	z/OS V1 R2	5
z/OS V1 R3	HJS7705	z/OS V1 R3	6
z/OS.e V1R3	HJS7705	z/OS.e V1R3	7
z/OS V1 R4	HJS7707	z/OS V1 R4	8

Figure 5-17 JES3 active releases

Notes:

Availability	End of service
1. September 24, 1999	September 30, 2002
2. March 31, 2000	March 31, 2003
3. September 29, 2000	At least September 2004
4. March 30, 2001	March 2004
5. October 26, 2001	October 2004
6. March 29, 2002	March 2005
7. March 29, 2002	March 2005
8. September 27, 2002	September 2005

5.2.1 JES3 coexistence maintenance

During JES3 processing on the global, downlevel processors identify (in a data area shared between JES3 processors) what their level of JES3 is so that a global at the z/OS 1R4 level can make decisions about that processor based on its level.

The following PTFs for APAR OW52172 must be installed for coexistence, migration, and fallback irrespective of whether MAINPROCs are being added, deleted or changed:

- UW86764 for OS/390 V2R8
- UW86765 for OS/390 V2R9
- UW86766 for OS/390 V2R10 and z/OS V1R1
- UW86767 for z/OS RV12 at PUT0210

Therefore, this support allows future JES3 releases to coexist with HJS7705, HJS7703, or HJS6609; and provides the capability to fall back to HJS7705, HJS7703, HJS6609, or HJS6608. This service can be installed on all processors in any order.

Archived

Figure 5-18 shows the supported JES3 releases and which level of the BCP that release runs on and coexists with.

	JES3 OS/390 V2 R8	JES3 OS/390 V2 R9	JES3 OS/390 V2 R10	JES3 z/OS V1 R1	JES3 z/OS V1 R2	JES3 z/OS V1 R3	JES3 z/OS V1 R4
OS/390 V2 R8							
OS/390 V2 R9							
OS/390 V2 R10							
z/OS V1 R1							
z/OS V1 R2							
z/OS V1 R3							
z/OS V1 R4							

Figure 5-18 JES3 and BCP release compatibility

Archived

6

JES2 Version 1 Release 4 enhancements

This chapter describes the change made to JES2 V1R4. The major themes for V1R4 are availability and performance. This chapter covers the support for all functional changes introduced by z/OS V1R4, as follows:

- A JES2 health monitor
- End of Memory (EOM) processing changes
- Enhanced recovery from bad JES2 checkpoint
- HASP access method (HAM) I/O improvements
- INCLUDE initialization statement enhancements
  - Default parmlib processing externals
- //XMIT JCL support
- SFSF enhancements

© Copyright IBM Corp. 2003

6.1 JES2 health monitor

The JES2 health monitor was developed as the result of a number of multi-system outages and is intended to address situations where JES2 is not responding to commands and it is not possible to determine what the problem is. There are many possibilities about what the problem could be and some examples of problems are:

- A JES2 command that is taking a long time to complete
- An error in a JES2 module or exit
- Checkpoint hangs

The JES2 health monitor is intended to help identify the problem so that corrective action can be taken. The monitor is designed to collect data that could be useful in determining the cause of the problem.

This monitor is not intended to be a performance monitor. Though the monitor does collect data that could be useful in tuning JES2, that is not the intended purpose of the monitor. Monitoring performance parameters is a possible future consideration for the monitor.

6.1.1 JES2MON address space

The monitor runs in a separate address space from JES2, as shown in Figure 6-1 on page 67. The name is JES2MON where JES2 is the name of the subsystem being monitored. There is one monitor address space per JES2 address space.

The monitor itself is a set of subtasks in the monitor address space. Each subtask does a particular task. The subtasks that execute in the JES2MON address space are:

- Main task
- Sampler task
- Probe task
- Command task

The monitor starts as a part of JES2 initialization processing. Once the monitor address space is started, the first code that runs in the address space copies the monitor load module from the JES2 address space into the monitor address space at the same address the code was loaded in the JES2 address space.

Categories of problem areas

Potential errors for monitoring in the JES2 address space are divided into three categories, based on the severity and nature of the problem, as follows:

Notices Notices are conditions that arise in JES2 but are not time-related in nature (that is, it does not matter how long the condition existed). Notices describe conditions that could explain why JES2 is not processing normally. Notices are not time-related. Notices are displayed on the **$JDJES** and **$JDSTATUS** commands. Many have related JES2 messages or commands that further explain the situation. They are gathered to provide a single place to determine what is happening in the JES2 address space.

Events An event occurs when a normal JES2 process lasts too long. Depending on the duration of the event, it is first tracked as an incident; then, if it lasts long enough, an alert. Just how long an event must last before being tracked or alerted depends on the specific event.

Alerts Alerts generate periodic highlighted messages to the operator. The **$JDSTATUS** command displays all alerts that are outstanding for the JES2 address space and all notices.

Figure 6-1 JES2 and JES2MON address spaces

JES2 termination

If JES2 comes down cleanly, the monitor is stopped. If JES2 ABENDs, the monitor remains active while the JES2 address space is down. On a hot start of JES2, if the monitor code is updated, the monitor is automatically restarted. If the monitor fails for any reason, there is code in the JES2 address space to restart the monitor.

When JES2 terminates, it posts the monitor main task. If this is a normal JES2 termination, using the **$PJES2** command, then the monitor address space is shut down and the ECSA storage areas are freed and the subtasks will shut down.

If this is an abnormal JES2 termination, the monitor main task is posted that JES2 is down. The posts of the monitor main task is propagated to all subtasks.

6.1.2 Sampler processing

The sampler subtask gathers data about what the JES2 main task TCB, main PRB, and current RB are doing. This is done by using an access list entry token (ALET) service to directly access storage in the JES2 address space. Based on the data obtained, the sampler determines if the JES2 main task is:

- In its normal MVS wait in the HASPNUC module
- In some other MVS wait
- Waiting for the local lock
- Non-dispatchable
- Waiting for a page fault resolution
- Running code normally

The sampler does some initial loop analysis and keeps a list of all MVS waits JES2 encounters. The sampler also tracks data on resource usage. The resources monitored are currently the same as those reported on the HASP050 message. A list can be seen later in some of the command responses.

Once this is complete, a new STIMER is established and the code returns to MVS.

Sampler time

Sampler time starts at zero when the monitor starts. Every time a sample is taken, a counter is updated. When the counter reaches 20, one sampler second is added to the “sampler time”.

Sampler time makes it possible to limit the number of bogus alerts generated when the system is unable to process any work. No alerts are issued unless the appropriate number of samples can be taken. Since probes are based on sampler data, if tracking starts at 4 seconds and alerts at 8 seconds, then 4 sampler seconds (or 80 samples) must pass after the tracking starts before an alert is issued.

6.1.3 Probe processing

Probe processing consists of looking at the sampler data and determining if there is a problem. It also accesses the JES2 address space to do some of its analysis. Probe processing issues tracks or alerts as appropriate. Only one alert for probe one will be active at a time. Multiple tracks can be active at any one time. Currently, the time limits for tracks and alerts and the message IDs that are issued are shown in Example 6-1:

Example 6-1 Time limits for tracks and alerts

```
Main task wait (HASP9201) track 2 seconds, alert 8 seconds
Loop detected (HASP9202) track 2 seconds, alert 8 seconds
Long PCE dispatch (HASP9203) track 4 seconds, alert 10 seconds
Main task busy (HASP9204) track 4 seconds, alert 20 seconds
Local lock wait (HASP9208) track 2 seconds, alert 8 seconds
Non-dispatchable (HASP9209) track 2 seconds, alert 8 seconds
Paging wait (HASP9210) track 2 seconds, alert 8 seconds
Not running (HASP9211) track 4 seconds, alert 20 seconds
```

After all tracks and alerts are deleted (and there was at least one alert), an all-clear message is issued (HASP9301) as follows:

```
$HASP9301 JES2 MAIN TASK ALERTS CLEARED
```

JES2 main task probe reporting

The probe reports on the following events:

- JES2 main task normal wait (no message)
- JES2 non-dispatchable, local lock wait, paging
- Unexpected MVS wait
- Looping (within ± 500 decimal bytes)
- PCE long dispatch (no $WAIT)
- JES2 main task busy (not MVS waiting)
- JES2 not running (long wait at normal MVS wait)
- All-clear message when no problems exist

JES2 main task busy is detected when the JES2 address space has not entered its normal MVS wait for a long time. Main task not running is detected when the main task is in its normal wait even though there are indications that it should be running.

Checkpoint status probe reporting

The probe reports on the following events:

- JES2 checkpoint lock:
  - Tracks and alerts when this system holds the checkpoint longer than the MASDEF HOLD= specification
  - An all-clear is issued when the CKPT no longer held

Probe processing examines the current checkpoint lock obtain time and the hold value to determine if a track or an alert is needed. Checkpoint tracks and alerts can be issued at the same time as other tracks and alerts. The checkpoint lock held probe starts tracking at 2 times hold, alert at 2 times hold plus 10 seconds.

If hold is less than 100, then tracking starts at 2 seconds. If hold is greater than 5 minutes (30000, then checkpoint holds are not tracked. When the checkpoint lock is released (and there had been an alert), then an all-clear message is issued (HASP9302) as follows:

```
$HASP9302 JES2 CHECKPOINT LOCK RELEASED
```

Note: The monitoring of the checkpoint held condition is based on hold times. If the hold time is set high (intentionally or by mistake), this condition is not monitored.

BERT lock and PCE wait probe reporting

The probe reports on the following events:

- BERT and job locks
  - Examines BERTL and LOCK PCE wait queues
  - Uses PSV wait time to determine how long PCE has been waiting
  - Only detects actual contention
  - Does not detect locks delaying processing
  - Up to 100 PCEs examined

Probe processing examines the dispatcher wait queues and issues tracks or alerts as appropriate. The time the PCE has been on the queue dictates whether the condition is tracked or an alert is issued. Currently, the time limits are:

- BERT lock held (HASP9205) track 4 seconds, alert 20 seconds
- Job lock held (HASP9206) track 4 seconds, alert 20 seconds
- $DILBERT PCE is ignored when encountered on the wait queue.

Only actual lock contention is detected. If a lock is held for a job that is queued for processing to a particular phase, and the lock is preventing selection, that will only be identified if a PCE is actually waiting for the lock. If the lock causes the job to be bypassed, then no PCEs are waiting and no message is issued. In this case, the symptom is a job is not running and a **$DJ** command should identify that the job is locked.

Probe analysis results

After the probe looks for conditions that may indicate a potential problem, it then examines how long the condition has existed. Based on the duration, it will do one of the following:

- Ignore the condition if it has only been happening for a very short time.
- Start tracking it as a potential problem. This creates a record in the monitor address space which can be displayed via the $JDJES command.
- Start alerting the condition. This causes a message to be issued to the operator of the problem. The message may be repeated on a timer with updated status information.

Events are grouped based on type. Two major groupings are main task events and checkpoint lock held events. When these transition from having had an alert to no tracks, an all-clear message is issued.

Probe messages

Example 6-2 lists messages generated by the probe processing. All messages have a duration associated with them.

- Exit and PCE information is displayed if appropriate.
- Exit, PCE, and job information is not displayed for messages $HASP9207 and $HASP9211.
- Command information is displayed if the PCE is the command PCE and there is a current command.
- If the PCE is the warm start PCE, the JQE index is displayed instead of the job number. This is because warm start processes the job in job index order.

Example 6-2 Typical probe messages

```
$HASP9201 JES2 MAIN TASK WAIT DETECTED AT module+offset
$HASP9202 POTENTIAL JES2 MAIN TASK LOOP DETECTED NEAR module+offset
$HASP9203 LONG PCE DISPATCH
$HASP9204 JES2 MAIN TASK BUSY
$HASP9205 PCE WAITING FOR BERT LOCK
$HASP9206 PCE WAITING FOR JOB LOCK
$HASP9207 JES2 CHECKPOINT LOCK HELD
$HASP9208 JES2 MAIN TASK LOCAL LOCK WAIT AT module+offset
$HASP9209 JES2 MAIN TASK NON-DISPATCHABLE AT module+offset
$HASP9210 JES2 MAIN TASK PAGING WAIT AT module+offset
$HASP9211 JES2 MAIN TASK NOT RUNNING
```

In Example 6-3, two alerts are being issued: one for the potential loop, and one to indicate that we are holding the checkpoint lock. The checkpoint lock alert is independent of all other alerts and is intended to inform the installation of the multisystem nature of the current problem.

Example 6-3 Probe example

```
23.10.29 *$HASP9202 POTENTIAL JES2 MAIN TASK LOOP DETECTED NEAR HASTDIAG+01F2FE
DURATION-000:00:14.60 PCE-COMM     EXIT-NONE JOB ID-NONE
23.10.29 *$HASP9207 JES2 CHECKPOINT LOCK HELD
DURATION-000:00:14.66
23.11.01 *$HASP9202 POTENTIAL JES2 MAIN TASK LOOP DETECTED NEAR HASTDIAG+01F2FE
DURATION-000:00:46.61 PCE-COMM     EXIT-NONE JOB ID-NONE
23.11.01 *$HASP9207 JES2 CHECKPOINT LOCK HELD
DURATION-000:00:46.66
23.11.33 *$HASP9202 POTENTIAL JES2 MAIN TASK LOOP DETECTED NEAR HASTDIAG+01F2FE
DURATION-000:01:18.62 PCE-COMM     EXIT-NONE JOB ID-NONE
23.11.33 *$HASP9207 JES2 CHECKPOINT LOCK HELD
DURATION-000:01:18.67
23.11.41  $HASP468 MONITOR  OK
23.11.41  $HASP9301 JES2 MAIN TASK ALERTS CLEARED
23.11.41  $HASP9302 JES2 CHECKPOINT LOCK RELEASED
```

Note: Messages with an asterisk (*) before the message id are issued as highlighted messages.

6.1.4 Command processing

All health monitor commands have the JES2 command prefix, followed by a letter “J”. All commands that have the JES2 command prefix followed by a J are sent to the monitor command subtask. If the monitor does not recognize the command, it is routed to the JES2 address space for normal command processing.

Because of this, commands starting with a J may execute out of order from other command processing issued from the same source. As with other JES2 commands, spaces and comments (/* */) are ignored.

RACF profiles

RACF calls are made for all commands. The format of the RACF entity name protecting commands is similar to what is used for other JES2 commands. The RACF profiles are of the form:

```
jes2MON.action.object
```

jes2 is the monitored subsystem name. For example, to display JES2 information, READ access is required to the RACF profile names, as follows:

```
JES2MON.DISPLAY.JES
JES2MON.DISPLAY.STATUS
JES2MON.DISPLAY.DETAILS
JES2MON.DISPLAY.HISTORY
JES2MON.DISPLAY.MONITOR
```

Monitor commands

The HASJCMDS module processes all monitor commands. The commands are intercepted in the command SSI. All commands that start with a letter “J” are queued to the monitor. An ECB is posted to indicate a command was received. Commands can also be received from HASPCOMM automatic command processing (queued in the same way as commands from the SSI).

> **Note:** Commands from NJE, RJE, the initialization deck, and JCL are not recognized. No exits are called for monitor commands (other than pre/post SAF exits 36 and 37).

There are some JES2 settings that do not apply to monitor commands, such as command limits, command redirection, and display max. The command limits do not apply because we may be at the limit due to a JES2 problem, for the following reasons:

- DISPMAX on the CONDEF statement does not apply because some commands have large output.
- Monitor commands are not limited by CONDEF CMDNUM=.
- RDIRAREA was not honored because L= support was not in the original support and it was not deemed important enough to do.

Archived

Table 6-1 displays the monitor commands.

Table 6-1 Monitor commands

Command	Description
$JDSTATUS	Display current status of JES2
$JDJES	Display information about JES2
$JDMONITOR	Display monitor task and module status information
$JDDETAILS	Display detailed information about JES2 resources, sampling and MVS waits
$JDHISTORY	Display history information
$JSTOP	Stops the monitor (JES2 will restart it automatically within a few minutes)

$JDSTATUS command

This command displays the current status of JES2. It is used to identify potential JES2 problems. This is the primary command to determine what problems may exist in JES2. Only conditions that the monitor considers potential problems are displayed. Two types of information are displayed:

Alerts These are the alerts for which the monitor has already issued a message

Notices These are other conditions which can exist in JES2 which could be contributing to a problem, but are not things that the monitor displays via alerts.

Example 6-4 $JDSTATUS command examples

```
$JDSTATUS
$HASP9120 D STATUS
$HASP9121 NO OUTSTANDING ALERTS
$HASP9150 NO JES2 NOTICES

$jdstatus
$HASP9120 D STATUS
$HASP9121 OUTSTANDING ALERTS
$HASP9211 JES2 MAIN TASK NOT RUNNING
DURATION-000:00:28.14
$HASP9150 JES2 NOTICES
$HASP9159 JES2 EXECUTION PROCESSING STOPPED ($PXEQ)
```

The second command in Example 6-4 shows one outstanding alert and one notice.

Notice messages

Example 6-5 on page 73 displays the possible notice messages for the health monitor.

Example 6-5 Health monitor notice messages

```
$HASP9150 {NO} JES2 NOTICES
$HASP9151 JES2 ADDRESS SPACE NOT ACTIVE
$HASP9152 JES2 INITIALIZING
$HASP9153 JES2 TERMINATING
$HASP9154 CKPT RECONFIGURATION IN PROGRESS
$HASP9155 ADDRESS SPACES WAITING FOR INTERNAL READERS
$HASP9156 ADDRESS SPACES WAITING FOR SPOOL SPACE
$HASP9157 CANNOT RESTART JES2, IPL REQUIRED
$HASP9158 JES2 PROCESSING STOPPED, $S NEEDED
$HASP9159 JES2 EXECUTION PROCESSING STOPPED ($PXEQ)
$HASP9160 AT LEAST ONE PCE HAS ENDED
$HASP9161 NOT ALL SPOOL VOLUMES ARE AVAILABLE
$HASP9162 PCES WAITING FOR SPOOL SPACE
$HASP9163 FAST SPOOL GARBAGE COLLECTION (SPOOLDEF GCRATE=FAST)
```

Creating a RACF profile, JES2MON.DISPLAY.STATUS, access READ protects this command.

$JDJES command

This command displays information about JES2. It displays events that are being tracked but are not necessarily problems; see Example 6-6. This command is very similar to the **$JDSTATUS** command; the major difference is that it displays information that may not be a problem in addition to what the monitor considers a problem.

Example 6-6 $JDJES command example

```
$jdjes
$HASP9120 D JES
$HASP9121 NO OUTSTANDING ALERTS
$HASP9122 NO INCIDENTS BEING TRACKED
$HASP9150 NO JES2 NOTICES
```

Example 6-7 shows there are four items being tracked at the time of this command. Any one of these could, if it lasts long enough, become an alert.

Example 6-7 $JDJES command example

```
$jdjes
$HASP9120 D JES
$HASP9121 NO OUTSTANDING ALERTS
$HASP9122 INCIDENTS BEING TRACKED
$HASP9204 JES2 MAIN TASK BUSY
DURATION-000:00:09.55 PCE-COMM     EXIT-NONE JOB ID-NONE
$HASP9203 LONG PCE DISPATCH
DURATION-000:00:09.55 PCE-COMM     EXIT-NONE JOB ID-NONE
$HASP9201 JES2 MAIN TASK WAIT DETECTED AT 7F6FC5B6
DURATION-000:00:09.51 PCE-COMM     EXIT-NONE JOB ID-NONE
$HASP9207 JES2 CHECKPOINT LOCK HELD
DURATION-000:00:09.75
$HASP9150 JES2 NOTICES
$HASP9158 JES2 PROCESSING STOPPED, $S NEEDED
```

Example 6-8 on page 74 shows more examples of the various notice messages, indicating some type of problem, that can be displayed by using the **$JDJES** command.

Example 6-8 $JDJES command notice messages

```
$HASP9150 {NO} JES2 NOTICES
$HASP9151 JES2 ADDRESS SPACE NOT ACTIVE
$HASP9152 JES2 INITIALIZING
$HASP9153 JES2 TERMINATING
$HASP9154 CKPT RECONFIGURATION IN PROGRESS
$HASP9155 ADDRESS SPACES WAITING FOR INTERNAL READERS
$HASP9156 ADDRESS SPACES WAITING FOR SPOOL SPACE
$HASP9157 CANNOT RESTART JES2, IPL REQUIRED
$HASP9158 JES2 PROCESSING STOPPED, $S NEEDED
$HASP9159 JES2 EXECUTION PROCESSING STOPPED ($PXEQ)
$HASP9160 AT LEAST ONE PCE HAS ENDED
$HASP9161 NOT ALL SPOOL VOLUMES ARE AVAILABLE
$HASP9162 PCES WAITING FOR SPOOL SPACE
$HASP9163 FAST SPOOL GARBAGE COLLECTION (SPOOLDEF GCRATE=FAST)
```

Creating a RACF profile, JES2MON.DISPLAY.JES, access READ protects this command.

$JDMONITOR command

This command displays the status of the monitor itself; see Example 6-9. It includes two messages:

- $HASP9100 displays the status of each of the monitor subtasks.
- $HASP9102 displays module information for each monitor module, similar to what the $D MODULE command displays for other JES2 modules.

Example 6-9 $JDMONITOR command example

```
$JDMONITOR
$HASP9100 D MONITOR
NAME     STATUS       ALERTS
-------- ------------ ------------------------
MAINTASK ACTIVE
SAMPLER  ACTIVE
COMMANDS ACTIVE
PROBE    ACTIVE
$HASP9102 MONITOR MODULE INFORMATION
NAME     ADDRESS  LENGTH   ASSEMBLY DATE  LASTAPAR LASTPTF
-------- -------- -------- -------------- -------- --------
HASJMON  06A2A000 00001088 07/10/02 04.34 NONE     NONE
HASJSPLR 06A2C000 00002838 07/10/02 04.35 NONE     NONE
HASJCMDS 06A2F000 00003050 07/10/02 04.34 NONE     NONE
```

Creating a RACF profile, JES2MON.DISPLAY.MONITOR, access CONTROL protects this command.

$JDDETAILS command

This command displays detailed information about JES2 resources, sampling, and MVS waits. It is intended as a diagnostic aid. This command displays various information that the monitor is collecting. The resource usage is reset at the top of every hour (low, high, and average). The output from this command is shown in Example 6-10 on page 75 and Example 6-11 on page 76.

The $HASP9104 message displays the resource usage and does this currently for all of the resources that are tracked by the $HASP050 message in the JES2 main task. The intent is to get all of the information in a single display and have it available when JES2 commands cannot be processed (for example, if a serious CMB shortage exists).

Example 6-10 $JDDETAILS command output

```
$JDDETAILS
$HASP9103 D DETAIL
$HASP9104 JES2 RESOURCE USAGE SINCE 2002.122 14:00:07
RESOURCE    LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- --------
BERT         2000     2000     1999     2000     2000
BSCB           10        0        0        0        0
BUFX          200        0        0        6        0
CKVR           17        0        0        2        0
CMBS          208        0        0        0        0
CMDS          200        0        0        0        0
ICES           33        0        0        0        0
JNUM        29001     2859     2855     2859     2856
JOES        10000     2770     2762     2770     2768
JQES         5000     2861     2857     2861     2858
LBUF          120        0        0        0        0
NHBS          100        0        0        0        0
SMFB          102        0        0        0        0
TGS         35360    18786    18786    21513    18838
TTAB            3        0        0        0        0
VTMB           50        0        0        0        0
```

Sampling statistics, display counts, and percentages of what the sampler detected the JES2 main task was doing is also reset at the top of every hour. The main task wait table from the **$jddetails** command is maintained until the monitor is recycled. It has information on explicit waits of the JES2 main task. A wait of the main task could indicate a potential problem.

The WT-COUNT, shown in Example 6-11 on page 76, is the number of unique times the sampler encountered a wait. The SM-COUNT is the number of time the sampler saw the wait. So in Example 6-11, the first wait was encountered once and sampled 1048 times. Since sampling is done 20 times a second, this wait lasted about 52 seconds. If the WT-COUNT was 2, then that would imply that we waited at the wait two times, for an average of 26 seconds each time.

Archived

Example 6-11 Sampling statistics from the $jddetails command

```
$HASP9105 JES2 SAMPLING STATISTICS SINCE 2002.186 19:00:00
TYPE               COUNT  PERCENT
----------------  ------  -------
ACTIVE              3372    11.92
IDLE               23850    84.36
LOCAL LOCK             0     0.00
NON-DISPATCHABLE       0     0.00
PAGING                 0     0.00
OTHER WAITS         1048     3.70
TOTAL SAMPLES      28270
$HASP9106 JES2 MAIN TASK MVS WAIT TABLE
DATE     TIME     ADDRESS  MODULE   OFFSET WT-COUNT SM-COUNT PCE XIT
-------- -------- -------- -------- ------ -------- -------- --- ---
2002.186 18:47:00 7F6FC5B6 UNKNOWN +000000        1     1048  10 JCO
2002.186 18:46:28 010E6D80 IGC018  +000B40        4       10  23 JCO
2002.186 18:46:25 0003D5B2 HASPCKPT+0065B2        1        2  23 JCO
2002.186 18:46:25 0003E25A HASPCKPT+00725A        1        1  23 JCO
2002.186 18:46:26 06B057D2 HASPIRDA+0027D2        2       36  23 JCO
2002.186 18:46:28 0125767E IEWFETCH+00152E        1        1  23 JCO
2002.186 18:46:28 00066FFA HASPDYN +000FFA        1        1  23 JCO
```

The processor control element (PCE) and the exit information table (XIT), shown in Example 6-12, give information on which PCE and exit were in control at the time of the wait. The fields can be the following:

- PCE can be:
  - Number - PCE id
  - MLT - multiple PCEs
- XIT can be:
  - number - exit number
  - MLT - Multiple exits
  - JCO - JES2 code only (no exits)
  - JNX - JES2 code and exits

RACF profile, JES2MON.DISPLAY.DETAILS, with access READ protects this command.

Resource type action recommendations

In general, take one or more of the following actions as appropriate for the RESOURCE types and the corresponding LIMIT and USAGE, as shown in Example 6-11 on page 76:

- Increase the quantity of the resource on its corresponding JES2 initialization statement.
- Increase the quantity of the resource with a $T command.
- Decrease demand for the resource (such as purging old held output to relieve a shortage of JOEs).
- Monitor temporary or non-impact shortages for possible future action.

The resource types and the corresponding initialization statements where you can modify the parameter are shown in Table 6-2 on page 77.

Archived

Table 6-2 Resource types and the corresponding JES2 initialization statement

Resource type	JES2 initialization statement
BERT (block extension reuse table)	BERTNUM on CKPTSPACE statement
BSCB (bisynchronous buffers)	BSCBUF on TPDEF statement
BUFX (extended logical buffers)	EXTBUF on BUFDEF statement
CKVR (checkpoint versions)	NUMBER on the CKPTDEF statement
CMBs (console message buffers)	BUFNUM on the CONDEF statement
CMDs	CMDNUM on the CONDEF statement
ICES (SNA interface control elements)	SESSIONS on the TPDEF statement
JNUM (job numbers)	RANGE on the JOBDEF statement
JQEs (job queue elements)	JOBNUM on the JOBDEF statement
JOEs (job output elements)	JOENUM on the OUTDEF statement
LBUF (logical buffers)	BELOWBUF on the BUFDEF statement
NHBs (NJE header/trailer buffers)	HDRBUF on the NJEDEF statement
SMFB (system management facilities buffers)	BUFNUM on the SMFDEF statement
TGs (spool space/track groups)	TGSPACE=(MAX=) on the POOLDEF statement
TTAB (trace tables)	TABLES on the TRACEDEF statement
VTMB (VTAM ® buffers)	SNABUF on TPDEF statement

For further information, see the $HASP050 Message For Resource Shortages section in *z/OS JES2 Diagnosis*, GA22-7531.

$JDHISTORY command

Display history information is shown in Example 6-12. The resource usage and sampling statistics are reset every hour. The old values are retained and displayed on this command. This data is maintained until the monitor is restarted; up to 72 hours of history is displayed.

Example 6-12 Output from the $jdhistory command

```
$jdhistory
$HASP9130 D HISTORY
$HASP9131 JES2 BERT USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00    12000      301      297     7052     2697
2002.122 15:00:00    12000     4874      296     7086     3468
2002.122 14:00:07     6000     3218     1999     3224     2880
2002.122 12:00:00     2000      284      284      514      297
$HASP9131 JES2 BSCB USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00       10        0        0        0        0
2002.122 15:00:00       10        0        0        0        0
2002.122 14:00:07       10        0        0        0        0
2002.122 12:00:00       10        0        0        0        0
$HASP9131 JES2 BUFX USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
```

```
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00      200        0        0        8        0
2002.122 15:00:00      200        4        0        8        1
2002.122 14:00:07      200        0        0        6        0
2002.122 12:00:00      200        0        0       10        0
$HASP9131 JES2 CKVR USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00       17        0        0        0        0
2002.122 15:00:00       17        0        0        0        0
2002.122 14:00:07       17        0        0        2        0
2002.122 12:00:00       17        0        0        0        0
$HASP9131 JES2 CMBS USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00      208        0        0      135        1
2002.122 15:00:00      208        0        0       92        0
2002.122 14:00:07      208        0        0        2        0
2002.122 12:00:00      208        0        0      208        3
$HASP9131 JES2 CMDS USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00      200        0        0        5        0
2002.122 15:00:00      200        0        0        1        0
2002.122 14:00:07      200        0        0        0        0
2002.122 12:00:00      200        0        0        1        0
$HASP9131 JES2 ICES USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00       33        0        0        0        0
2002.122 15:00:00       33        0        0        0        0
2002.122 14:00:07       33        0        0        0        0
2002.122 12:00:00       33        0        0        0        0
$HASP9131 JES2 JNUM USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00    29001     1625     1605     4998     3038
2002.122 15:00:00    29001     3892     1596     4998     3201
2002.122 14:00:07    29001     3099     2855     3467     3040
2002.122 12:00:00    29001     2481     1951     3466     2354
$HASP9131 JES2 JOES USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00    10000     1830     1796     6434     2226
2002.122 15:00:00    10000     1796     1785     1964     1820
2002.122 14:00:07    10000     1866     1842     2780     2135
2002.122 12:00:00    10000     3722     2675     5696     3448
$HASP9131 JES2 JQES USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00     5000     1627     1607     5000     3040
2002.122 15:00:00     5000     3894     1598     5000     3203
2002.122 14:00:07     5000     3101     2857     3469     3042
2002.122 12:00:00     5000     2483     1953     3468     2356
$HASP9131 JES2 LBUF USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00      120        0        0        0        0
2002.122 15:00:00      120        0        0        0        0
2002.122 14:00:07      120        0        0        0        0
```

```
2002.122 12:00:00      120        0        0        0        0
$HASP9131 JES2 NHBS USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00      100        0        0        0        0
2002.122 15:00:00      100        0        0        0        0
2002.122 14:00:07      100        0        0        0        0
2002.122 12:00:00      100        0        0        0        0
$HASP9131 JES2 SMFB USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00      102        0        0        4        0
2002.122 15:00:00      102        0        0        4        0
2002.122 14:00:07      102        0        0        0        0
2002.122 12:00:00      102        0        0        4        0
$HASP9131 JES2 TGS  USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00    35360    14645    14513    20092    16145
2002.122 15:00:00    35360    16772    14209    18205    17011
2002.122 14:00:07    35360    18205    18143    21513    18384
2002.122 12:00:00    35360    12891    12305    25661    14465
$HASP9131 JES2 TTAB USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00        3        0        0        0        0
2002.122 15:00:00        3        0        0        0        0
2002.122 14:00:07        3        0        0        0        0
2002.122 12:00:00        3        0        0        0        0
$HASP9131 JES2 VTMB USAGE HISTORY
DATE     TIME        LIMIT    USAGE      LOW     HIGH  AVERAGE
-------- -------- -------- -------- -------- -------- --------
2002.122 16:00:00       50        0        0        0        0
2002.122 15:00:00       50        0        0        0        0
2002.122 14:00:07       50        0        0        0        0
2002.122 12:00:00       50        0        0        0        0
$HASP9132 MAIN TASK SAMPLING PERCENT HISTORY
DATE     TIME       COUNT ACTIVE   IDLE   WAIT L-LOCK N-DISP PAGING
-------- --------  ------ ------ ------ ------ ------ ------ ------
2002.122 16:00:00   71369   2.06  97.46   0.00   0.47   0.00   0.00
2002.122 15:00:00   71388   1.48  98.20   0.00   0.30   0.00   0.00
2002.122 14:00:07  140099   1.28  87.43  10.39   0.86   0.01   0.00
2002.122 12:00:00   71395   1.12  98.43   0.00   0.43   0.00   0.00
$HASP9132 MAIN TASK SAMPLING PERCENT HISTORY
DATE     TIME       COUNT ACTIVE   IDLE   WAIT L-LOCK N-DISP PAGING
-------- --------  ------ ------ ------ ------ ------ ------ ------
2002.122 16:00:00   71369   2.06  97.46   0.00   0.47   0.00   0.00
2002.122 15:00:00   71388   1.48  98.20   0.00   0.30   0.00   0.00
2002.122 14:00:07  140099   1.28  87.43  10.39   0.86   0.01   0.00
2002.122 12:00:00   71395   1.12  98.43   0.00   0.43   0.00   0.00
```

$JSTOP command

This command shuts down the monitor address space. JES2 restarts the address space in a few minutes. The command is intended to recycle the monitor to correct any errors it may be having or to clear any history it may be keeping. Recycling the monitor does not include any new fixes to the monitor code. If a fix needs to be applied to the monitor, the JES2 address space must be recycled by doing a JES2 hot start.

Note: JES2 will also restart the monitor address space if the MONITOR is cancelled, forced, CALLRTM'ed, or ABENDs.

Example 6-13 shows an example of the **$JSTOP** command.

Example 6-13 $JSTOP command example

```
$jstop
 $HASP9101 MONITOR STOPPING
 $HASP9085 JES2 MONITOR ADDRESS SPACE STOPPED FOR JES2
 IEF404I IEESYSAS - ENDED - TIME=00.31.08
 IEF403I IEESYSAS - STARTED - TIME=00.31.34
 $HASP9084 JES2 MONITOR ADDRESS SPACE STARTED FOR JES2
```

6.2 End of memory

End of memory occurs when an address space is being deleted (the memory is going away). JES2 gets control in an SSI to clean up any JES2 resources the address space may have owned when it terminated. The problem is that in JES2 V1R2, MVS added code to clean up services that were stuck in end of memory processing. JES2 could appear stuck because it waits for the JES2 address space before completing the clean up. Since JES2 may be down or unable to access the JES2 checkpoint, this can be a long wait. To prevent being ABENDed by the support added in JES2 V1R2, JES2 added a timer to make it appear JES2 is actively processing the request.

To solve the problem, JES2 has now eliminated all explicit waits in JES2 end of memory processing. If there are resources to be cleaned up, the request is queued to JES2 for processing, but the address space is allowed to continue to terminate. To accomplish this, code was changed to not use the $SJB (an internal job-related control block) to queue requests to JES2, but to use instead individual work elements representing the resource to be cleaned up.

Some consequences of this change may impact installations, as follows:

- The PSO control block has been moved from ECSA to a new data space. The PSO is an internal control block used in the external writer interface. Though never an intended interface, there were some vendors looking at this control block. Vendors are aware of this change; however, new levels of some products that access SYSOUT may be needed.
- The STATUS/CANCEL TSO interface was also updated to pass data in a new control block that resides in a data space. This interface is similar to the PSO interface, however, we know of no vendors accessing any internal data areas used in this interface. It is mentioned for completeness.
- Because we now queue work to JES2 for processing, it is possible for an address space to be gone from an MVS point of view but still executing from a JES2 point of view. This can cause some confusion when first encountered.

6.2.1 TSO/E multiple logons

JES2 is no longer enforcing one user ID, one logon throughout the JESplex. This check was added to JES2 Release 3 (1976) when JES2 first became a multisystem component. JES2 Release 2 did not support a MAS.

You can protect against multiple logons by issuing a SYSTEMS ENQ with a major name of SYSIKJUA. If an installation fails to specify that this major ENQ is a SYSTEMS, then it is possible for a given user ID to logon to multiple members of the JES2 MAS.

Note: Multiple instances of a single user ID logged onto a JESplex is not officially supported. IBM will take no APARs if there is a problem because of the failure of an installation to update the RNL list.

6.3 Checkpoint data corruption

Problems have occurred over the years where an installation started with the wrong spool or checkpoint volumes online (production on a test system, or test on a production system). Also, problems have occurred where only one checkpoint data set was bad. Typically, an installation notices this when they start to see thousands of error messages flood the screen. Often the system is stopped at that point to try to prevent problems, but it is too late; JES2 has already written some or all of the bad data to the checkpoint.

New logic in JES2 V1R4 ensures that nothing is written to the checkpoint until the warm start processing completes. If more than ten errors are encountered, the operator is given the option of not starting JES2 (before anything has been written). Now, the checkpoint is never written to until warm start processing has completed.

6.4 HAM I/O improvements

HAM is the access method used to read and write data sets to SPOOL. It has been improved in this release to have better performance and greater RAS. HAM now uses multirecord I/O to read and write SYSOUT and SYSIN from and to the spool. EXCPVR also replaces EXCP to reduce overhead.

6.5 Enhanced INCLUDE statement externals

With JES2 V1R2, the syntax of the INCLUDE initialization statement specifies a required data set name, and a volser and unit (required only if needed for allocation). The data set name can have a member name. The statements in the included data set are processed immediately. When the end of the included data set is reached, processing continues with the statement after the include of the original data set. Includes can be nested. There is loop detection to prevent a nesting loop.

```
INCLUDE DSNAME=dsn,VOLSER=vol,UNIT=unit
```

The INCLUDE statement has the following considerations:

- DSNAME can include a member name.
- VOLSER and UNIT are optional (if data set is cataloged).
- Statements in the data sets are processed immediately.
- An INCLUDE data set can have INCLUDE statements.

Example 6-14 on page 82 shows four members that make up the JES2 initialization definitions in the JES2 procedure.

Example 6-14 Sample JES2 procedure with four initialization members

```
//JES2     PROC
//         EXEC PGM=HASJES20,...
//HASPPARM DD DSN=SYS1.PARMLIB(MEMBER1)
//         DD DSN=SYS1.PARMLIB(COMMON)
//         DD DSN=SYS1.PARMLIB(NJEDEFS)
//         DD DSN=SYS1.PARMLIB(PRINTERS)
```

Using the INCLUDE statement, the JES2 procedure is simplified, as shown in Figure 6-2.

Figure 6-2 New HASPPARM specification using the INCLUDE statement

6.5.1 Enhancement to INCLUDE statement

The improvements to the INCLUDE initialization statement allow the capability:

1. To read an additional member from the current data sets for the initialization statements.

   The new syntax added to the INCLUDE statement in JES2 V1R4 is as follows:

   – INCLUDE MEMBER=member_a

     member_a should be in the current parmlib data set being processed.

   – INCLUDE PARMLIB_MEMBER=member_b

     member_b should be in the default logical parmlib.

> **Note:** The new keywords MEMBER and PARMLIB_MEMBER are mutually exclusive with the existing keywords DSNAME or VOLSER or UNIT.

2. To read a member from the logical data set for initialization decks. The default logical parmlib will be SYS1.PARMLIB.

If we try to include the PDS data set without the member name, it will issue the HASP003 error message. Therefore, DSNAME should contain the member name if it is PDS. Also, if we try to include the member from the current PS data set, it will issue the error message. INCLUDE PARMLIB_MEMBER will include only from the logical parmlib, whatever may be the current data set.

Default parmlib member

This JES2 V1R4 support allows for a default PARMLIB member. This member is used only if there is no HASPPARM DD and the operator does not use a HASPPARM= start option. The default is HASJES2, where JES2 is the actual subsystem name. The default member comes from the logical PARMLIB concatenation.

There is also the ability to specify the “default” member as a start option, (MEMBER=). If MEMBER= is specified, it will be used. HASPPARM = and MEMBER= are mutually exclusive and can be specified as follows:

MEMBER= If specified by the operator as a start option, then use that member as the default PARMLIB.

HASPPARM= If specified by the operator as a start option, then use that DD name for the PARMLIB.

HASPPARM DD If this DD exists in the JES2 procedure, then use that DD name as the PARMLIB.

Note: If none of the above are specified, then use HASJES2 as the default PARMLIB concatenation.

If there is a problem with any of these, message HASP450 is issued. The operator can then respond if JES2 should continue (as is done previously if the OPEN of the DD for PARMLIB fails). If the operator replies Y, then JES2 initialization continues and console mode is entered.

PARMLIB search order

Following is the search order to determine which PARMLIB to use:

1. If the HASPPARM=ddname parameter is specified, use that DD.
2. If the MEMBER=PARMLIB_MEMBER= parameter is specified, use that member from the logical PARMLIB.
3. If neither is specified, try to open DD with a ddname of HASPPARM.
4. If HASPPARM DD is not found, use the HASjesx member from the logical parmlib which is the default PARMLIB concatenation (this is HASJES2, in the examples).

6.5.2 Using default PARMLIB

With the new enhancements to the INCLUDE processing, it is now possible to remove the HASPPARM DD from the JES2 procedure, as shown in Figure 6-3 on page 84. In this example, all members must be in the logical PARMLIB concatenation and they are in SYS1.PARMLIB.

The HASJES2 member contains an INCLUDE statement that points to PARMLIB member JES2M1 as the value of &SYSCLONE is M1.

Archived

Figure 6-3 Simplified JES2 procedure and using the default PARMLIB member

New start option with default PARMLIB

You can now start JES2 with the new start option of reading the initialization stream from the default PARMLIB specification, as follows:

- S JES2,PARM=('MEMBER=member')
- S JES2,PARM=('PARMLIB_MEMBER=member')

The operator should reply to the $HASP467 message as follows:

```
r xx,MEMBER=member
```

Note that member should be the member of the logical parmlib in each of the above specifications.

The following conditions should be considered:

- HASPPARM=ddname and MEMBER= are mutually exclusive.
- If neither HASPPARM= nor MEMBER= is specified, then it will process from the default HASjesx member of the logical parmlib (where jesx is the JES2 subsystem name).
- IBM does not ship a default parmlib member.

Starting JES2 without a JES2 PROC

In an emergency, you can start JES2 without a JES2 procedure because of the elimination of the need to specify the HASPPARM data set and the PROCLIB data sets, a change that was introduced in JES2 V1R2. Start JES2 as follows:

```
S IEESYSAS,PROG=HASJES20,JOBNAME=JES2
```

This start command assumes that HASJES20 is in the LINKLIST (no STEPLIB). During JES2 initialization, when the OPEN of HASPPARM fails, the logical parmlib member HASJES2 will be used. If HASJES2 is not found, message $HASP469 is issued.

Archived

```
S IEESYSAS,PROG=HASJES20,JOBNAME=JES2,PARM='MEMBER=MEMBER2'
```

This start command uses the logical parmlib member MEMBER2 and no OPEN of the HASPPARM DD is attempted. This option is new with JES2 V1R4 because of the new MEMBER= capabilities. If MEMBER2 is not found, message $HASP469 is issued.

6.6 XMIT JCL card externals

JES2 now supports the XMIT JCL card to route job execution to another node. This JCL statement is being made available in the JCL format mainly because JES2 installations are attempting to reduce their use of /* JECL cards.

There is currently no way to route job execution to another node without using JECL. Additionally, installations with both JES2 and JES3 are confronted with another JCL incompatibility and must maintain two sets of JCL, or manually change JCL, depending on where the input processing for the job is done.

Note: The JES2 JECL control statements /*XMIT, /*XEQ, and /*ROUTE XEQ are still supported.

The SUBCHARS= keyword is not supported and a JCL error occurs and a new message is issued:

```
HASP108 jobname NON-VALID XMIT STMT - reason
```

All the JCL statements between the XMIT JCT statement and a delimiter are transmitted to the specified node, as shown in Example 6-15. The delimiter is AA as specified on the XMIT statement. If DLM= is omitted, the delimiter is then the first /* statement.

Example 6-15 Job using the XMIT JCL statement

```
//JOBC JOB ALEX,'DEPT XYZ'
//X2 XMIT DEST=NODEA,DLM=AA
.......
        (job statements)
.......
AA
```

6.7 SDSF enhancements

There are no changes in SDSF for z/OS V1R3; however, there is a Small Programming Enhancement (SPE) that can also be applied to z/OS V1R2 SDSF. The SPE supports the z/OS Workload Manager (WLM) enclave service class reset function in z/OS V1R3.

The SDSF function consists of:

- Making the service class overtypeable on the ENC panel.
- The name of the column that is used in field lists in ISFPARMS also changed.
- Adding a column for subsystem to the ENC panel.

There are new action characters, **R** and **RQ**, to resume and quiesce enclaves on the ENC panel. For consistency, **R** and **RQ** action characters are also added to the DA panel to set the quiesce indicator for an address space.

This is an alternative to overtyping the Quiesce column on the DA panel.

The SPE consists of these PTFs:

- UQ58774, base code and English help panels
- UQ58773, Japanese panels

Archived

7

z/OS Workload Manager (WLM)

In this chapter, the following topics are described:

- For WLM enhancements in z/OS V1R3, refer to section:
  - 7.1, “Removal of WLM compatibility mode” on page 88
  - 7.2, “PAV dynamic alias management for paging devices” on page 99
  - 7.3, “WLM independent enclave service class reset” on page 101
  - 7.4, “WLM support for sub-capacity pricing” on page 104
  - 7.5, “WLM enqueue management enhancements” on page 105
  - 7.6, “WLM WebSphere performance enhancement” on page 107
- For WLM enhancements in z/OS V1R4, refer to section:
  - 7.7, “WLM batch initiator balancing enhancements” on page 107
  - 7.8, “Performance block application state reporting for enclaves” on page 113
  - 7.9, “WLM msys for Setup enhancement” on page 116
- For other notable WLM enhancements, refer to section:
  - 7.10, “WLM support for ESS FICON and I/O priority management” on page 116
  - 7.11, “WLM temporal affinity for WebSphere Application Server” on page 117
  - 7.12, “WLM velocity goals” on page 118
  - 7.13, “Enterprise workload management: eWLM” on page 119

© Copyright IBM Corp. 2003

7.1 Removal of WLM compatibility mode

WLM *compatibility mode* support is removed starting with z/OS V1R3. From this release and onward, only WLM *goal mode* operation is valid. As a consequence, everybody must now use a *service definition* to manage their systems. If a non-blank IPS is specified in the IEASYSxx parmlib member, message IRA903I is shown and the system IPLs in goal mode.

> **Note:** All IEAIPSxx and IEAICSxx, and most options in the IEAOPTxx members (those options that previously worked in WLM compatibility mode only) are ignored in the IEASYS parmlib member starting with z/OS V1R3. Only WLM goal mode operation will be tolerated by the system.
>
> We strongly recommend that you to migrate to goal mode *before* moving on z/OS V1R3.
>
> In regard to IEASYSxx, also note that as of z/OS V1R3 in IEASYSxx, there is a LICENSE= parameter which specifies which operating system is running (z/OS or z/OS.e). The default is for z/OS. There is no change necessary for your z/OS system if you plan to use the default.

7.1.1 Sample service definition with z/OS 1.3

WLM provides a sample service definition IWMSSDEF in SYS1.SAMPLIB. The service definition is a sample that can be installed using JCL that is also supplied in SYS1.SAMPLIB in member IWMINSTL.

WLM service definition description

For those who are new to goal mode, a service definition consists of the following:

Service policy
A named set of performance goals that workload management uses as a guideline to match resources to work. Service policies can be activated by an operator command, or through the ISPF administrative application utility function. A policy applies to all of the work running in a sysplex. Because processing requirements change at different times, service level requirements may change at different times. If you have performance goals that apply to different times, or a business need to limit access to processor capacity at certain times, you can define multiple policies. In order to start workload management processing, you must define at least one service policy. You can activate only one policy at a time.

Workloads
A workload is just a name and a description. Workloads aggregate a set of service classes together for reporting purposes.

Service classes
Service classes are subdivided into periods and they allow you to group work with similar performance goals, business importance, and resource requirements for management and reporting purposes. You assign performance goals to the periods within a service class.

Report classes
Report classes group work for reporting purposes. They are commonly used to provide more granular reporting for subsets of work within a single service class.

Resource groups
Resource groups define processor capacity boundaries across the sysplex. You can assign a minimum and maximum amount

of CPU service units per second to work by assigning a service class to a resource group.

Classification rules Classification rules determine how to assign incoming work to a service class and report class.

Application environments Application environments are groups of application functions that execute in server address spaces and can be requested by a client. Workload management manages the work according to the defined goal, and automatically starts and stops server address spaces as needed.

Scheduling environments Scheduling environments are lists of resource names along with their required states. If an MVS image satisfies all of the requirements in a scheduling environment, then units of work associated with that scheduling environment can be assigned to that MVS image.

Coefficients The amount of system resources an address space or enclave consumes is measured in service units. Service units are calculated based on the CPU, SRB, I/O, and storage (MSO) service an address space consumes. Service units are the basis for period switching within a service class that has multiple periods. The duration of a service class period is specified in terms of service units. When an address space or enclave running in the service class period has consumed the amount of service specified by the duration, workload management moves it to the next period. The work is managed to the goal and importance of the new period. Because not all kinds of service are equal in every installation, you can assign additional weight to one kind of service over another. This weight is called a *service coefficient*.

I/O priority management I/O priority queueing is used to control non-paging DASD I/O requests that are queued because the device is busy. You can optionally have the system manage I/O priorities in the sysplex based on service class goals. The default for I/O priority management is `no`, which sets I/O priorities equal to dispatching priorities. This is identical to how I/O priorities were handled prior to OS/390 R3. If you specify `yes`, then workload management sets I/O priorities in the sysplex based on goals. WLM dynamically adjusts the I/O priority based on how well each service class is meeting its goals and whether the device can contribute to meeting the goal. The system does not micro-manage the I/O priorities, and changes a service class period's I/O priority infrequently.

Dynamic alias management As part of the Enterprise Storage Subsystem's implementation of parallel access volumes, the concept of base addresses versus alias addresses is introduced. While the base address is the actual unit address of a given volume, there can be many alias addresses assigned to a base address, and any or all of those alias addresses can be reassigned to a different base address. With dynamic alias management, WLM can automatically perform those alias address reassignments to help work meet its goals and to minimize IOS queueing. When you specify `yes` for this value on the Service Coefficient/Service Definition Options panel, you enable dynamic alias management globally throughout the sysplex.

WLM will keep track of the devices used by different workloads and broadcast this information to other systems in the sysplex. If WLM determines that a workload is not meeting its goal due to IOS queue time, then WLM attempts to find alias devices that can be moved to help that workload achieve its goal. Even if all work is meeting its goals, WLM will attempt to move aliases to the busiest devices to minimize overall queueing.

7.1.2 Service policy description

Figure 7-1 shows a WLM service policy environment with a minimum number of definitions defined. They are the definitions defined in the sample service policy provided in SYS1.SAMPLIB(IWMSSDEF).

Service class goals

You can assign the following kinds of performance goals to service classes:

- Average response time
- Response time with percentile
- Velocity
- Discretionary

A service class can have multiple periods. Workload management manages a service class period as a single entity when allocating resources to meet performance goals. A service class can be associated with only one workload. You can define up to 100 service classes.

Service class importance

You assign an importance level to the performance goal. *Importance* indicates how vital it is to the installation that the performance goal be met relative to other goals. Importance is a number from 1 to 5.

Figure 7-1 WLM service policy with workloads and service classes

7.1.3 IPLing z/OS V1R3

A z/OS V1R3 and higher system can use any one of the following options to use a service definition the first time you IPL the new system:

1. The existing service definition you created on the system you migrate from, which means you are already in goal mode (the “recommended” option).
2. A sample service definition provided in SYS1.SAMPLIB (the “quick” option).
   a. To reduce the overall complexity of goal mode migration some powerful new WLM tasks are introduced: a sample service definition called IWMSSDEF residing in SAMPLIB, as well as sample JCL in SYS1.SAMPLIB(IWMINSTL).
3. The WLM default internal service definition (the “last resort” option).
   a. If no WLM service definition is defined at all, WLM uses a “default” service definition. (This was also the case in previous releases of z/OS when the system was operated in goal mode and no user-defined service definition was available.)

When the system is IPLed and a WLM Couple Data Set (CDS) is defined, then an installation-defined service definition can be installed, and a policy can be activated.

Sample service definition

The sample service definition provided in SYS1.SAMPLIB(IWMSSDEF) includes the following service classes, as shown in Figure 7-2. This panel is displayed by using the WLM ISPF application and selecting Option 2 from the primary menu once the service definition is installed in the WLM couple data set.

```
  Service-Class  View  Notes  Options  Help
-------------------------------------------------------------------------
                         Service Class Selection List          Row 1 to 9 of 9
Command ===> ____________________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar


Action  Class     Description                      Workload
  __    ASCH      APPC Transaction Programs        STCTASKS
  __    BATCH     Batch Workload                   BATCH
  __    CICS      CICS Transactions                ONLINES
  __    DB2QUERY  DB2 Sysplex Queries              DATABASE
  __    DDF       DDF Requests                     DATABASE
  __    IMS       IMS Transactions                 ONLINES
  __    OMVS      UNIX System Services             STCTASKS
  __    STC       Started Tasks                    STCTASKS
  __    TSO       TSO User Community               TSO
******************************* Bottom of data ********************************
```

Figure 7-2 Service classes in sample service definition

The service classes listed in Figure 7-2 are organized into the workloads shown in Figure 7-3 on page 92.

```
  Workload  View  Notes  Options  Help
-------------------------------------------------------------------------
                          Workload Selection List                Row 1 to 5 of 5
Command ===> ____________________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar
                                                   ----Last Change-----
Action  Name      Description                      User      Date
  __    BATCH     Batch Workloads                  IBMUSER   2001/08/13
  __    DATABASE  Database Workloads               IBMUSER   2001/08/13
  __    ONLINES   Online Workloads                 IBMUSER   2001/08/13
  __    STCTASKS  System Workloads                 IBMUSER   2001/08/13
  __    TSO       TSO Workloads                    IBMUSER   2001/08/13
****************************** Bottom of data ********************************
```

Figure 7-3 Workloads defined in the sample service definition

A service definition contains all the information about the installation needed for workload management processing. There is one service definition for the entire sysplex. The service level administrator normally specifies the service definition through the WLM administrative application. For installations that have no service definitions when converting to z/OS V1R3 or higher, this sample service definition allows you to IPL your new system in goal mode using the sample service definition.

The service level administrator normally sets up “policies” within the service definition to specify the goals for work. Using this sample service definition, you can use the IWMINSTL member of SYS1.SAMPLIB to install the service definition, as shown in “WLM install definition utility” on page 94. Using this JCL, you can specify the name of a service policy, as shown in Example 7-2 on page 95 and displayed in Figure 7-4.

```
  Service-Policy  View  Notes  Options  Help
-------------------------------------------------------------------------
                     Service Policy Selection List            Row 1 to 1 of 1
Command ===> ____________________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              7=Override Service Classes, 8=Override Resource Groups,
              /=Menu Bar
                                                   ----Last Change-----
Action  Name      Description                      User      Date
  __    WLMPOL    Sample WLM Service Policy        IBMUSER   2001/08/13
****************************** Bottom of data ********************************
```

Figure 7-4 WLM service policy in sample service definition

7.1.4 Sample service definition IWMSSDEF

The WLM sample service definition is in SYS1.SAMPLIB(IWMSSDEF). Member IWMSSDEF must be unpacked into a fixed block 80 partitioned data set format before being installed and a service policy activated.

To unpack the member, use the **TSO RECEIVE** command:

```
RECEIVE INDSNAME('SYS1.SAMPLIB(IWMSSDEF)')
```

When prompted, specify a PDS name of your choice:

```
DSNAME('your.def.pds')
```

The **RECEIVE** command places the sample definition in the desired data set, and even allocates it, if needed.

Messages from TSO RECEIVE command

Example 7-1 shows the output messages that result from issuing the following commands and replies to the system messages:

```
RECEIVE INDSNAME('SYS1.SAMPLIB(IWMSSDEF)')
Dataset WLM.SAMPLE.DEF from IBMUSER on IBM
Enter restore parameters or 'DELETE' or 'END' +
DSNAME('your.def.pds')
```

Example 7-1 Creation of WLM PDS (IEBCOPY messages and control statements)

```
IEBCOPY MESSAGES AND CONTROL STATEMENTS                                PAGE     1
IEB1135I IEBCOPY  FMID HDZ11G0  SERVICE LEVEL NONE     DATED 20020130 DFSMS 01.03.00 z/OS
01.03.00 HBB7706  CPU 2064
IEB1035I ERIK   IKJACCT  IKJACCNT 14:13:15 WED 29 MAY 2002 PARM=''
 COPY INDD=((SYS00080,R)),OUTDD=SYS00079
IEB1013I COPYING FROM PDSU  INDD=SYS00080 VOL=
DSN=SYS02149.T141314.RA000.ERIK.R0101098
IEB1014I       TO PDS  OUTDD=SYS00079 VOL=SBOX79 DSN=YOUR.DEF.PDS
IEB167I FOLLOWING MEMBER(S) LOADED FROM INPUT DATA SET REFERENCED BY SYS00080
IEB154I ADATAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I AETAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I ATTRTAB  HAS BEEN SUCCESSFULLY LOADED
IEB154I AXTTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I CRTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I EDATAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I EXTTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I GMTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I GRTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I OPTTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I PADTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I RCTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I RDATAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I RELTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I RGTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I RTAB     HAS BEEN SUCCESSFULLY LOADED
IEB154I RXTTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I SCTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I SDCTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I SDXTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I SETAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I SEXTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I SGTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I SPTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I SRTAB    HAS BEEN SUCCESSFULLY LOADED
IEB154I SSTTAB   HAS BEEN SUCCESSFULLY LOADED
IEB154I WLTAB    HAS BEEN SUCCESSFULLY LOADED
IEB1098I 27 OF 27 MEMBERS LOADED FROM INPUT DATA SET REFERENCED BY SYS00080
IEB144I THERE ARE 1 UNUSED TRACKS IN OUTPUT DATA SET REFERENCED BY SYS00079
IEB149I THERE ARE 5 UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
Restore successful to dataset 'YOUR.DEF.PDS'
***
```

Archived

7.1.5 WLM install definition utility

The WLM install definition utility allows you to install a service definition and activate a policy without having to use the ISPF WLM application. The utility may be submitted as a batch job, or executed as a started task, as shown in Figure 7-5. The purpose of this job or task is to load a service definition and activate a sample WLM policy. The WLM shipped sample installation utility job is in SYS1.SAMPLIB(IWMINSTL).

Next, the service definition may be installed in the WLM CDS, and it may then be activated. The whole process is illustrated in Figure 7-5 on page 94. Any valid service definition can be installed and any policy within that service definition can be activated by the utility.

Note: The install definition utility may also be used to alter a WLM policy by simply using commands or via batch jobs, without the need to use the WLM ISPF-supplied panels. This may simplify your WLM operation on a daily basis and may even be exploited by automation. Refer to “WLM goal mode migration reference information” on page 99 for examples.

Figure 7-5 WLM install definition utility process overview

The sample JCL parameters (SETPARMx) in the installation utility job (shown in Example 7-2 on page 95) that perform the installation and activation of the WLM policy are:

SVDEFPDS The PDS (FB 80) that contains the service definition. This is a required parameter which must specify a valid service definition data set.

FORCE Force the install to a WLM Couple Data set. The install will be forced even if the SVDEFID of the installed service definition is different than the definition you are attempting to install. This is not a required parameter. Anything other than FORCE=Y is interpreted as FORCE=N. The default is N.

POLNAME This parameter refers to the WLM policy to be activated. This is not a required parameter. If POLNAME is not given, no policy is activated. Supplying a POLNAME indicates an activate. The default is blank.

IWMINSTL job without activate

An installation job with **SVDEFPDS='YOUR.DEF.PDS'**, **FORCE='N'**, and **POLNAME='WLMPOL'** parameters coded is shown in Example 7-2.

Example 7-2 Sample install definition utility to install WLM service definition and activate policy

```
//IWMINSTL   JOB MSGLEVEL=(1,1),REGION=0M
//*
//SETPARM1  SET   SVDEFPDS='YOUR.DEF.PDS'
//SETPARM2  SET   FORCE='N'
//SETPARM3  SET   POLNAME='WLMPOL'
//*
//STEP1    EXEC  PGM=IKJEFT1B,
//         PARM='IWMARIDU F=&FORCE P=&POLNAME'
//SYSPROC  DD DSN=SYS1.SBLSCLIO,
//         DISP=SHR
//*
//ISPLLIB  DD DSN=ISP.SISPLPA,
//         DISP=SHR
//         DD DSN=ISP.SISPLOAD,
//         DISP=SHR
//ISPMLIB  DD DSN=ISP.SISPMENU,
//         DISP=SHR
//ISPPLIB  DD DSN=ISP.SISPPENU,
//         DISP=SHR
//ISPTLIB  DD DSN=ISP.SISPTENU,
//         DISP=SHR
//ISPSLIB  DD DSN=ISP.SISPSENU,
//         DISP=SHR
//ISPTABL  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1,5)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPPROF  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1,5)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPLOG   DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPCTL1  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPLST1  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//         DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1210)
//SVDEF    DD DSN=&SVDEFPDS,
//         DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
```

Job output

The service definition when installed contains information about the installed definition. Part of this information is the service definition ID which contains the name of the definition, the user ID of the last person who installed, a time stamp, and the name of the system the definition was installed from. This information from the service definition ID is displayed in the job output shown in Example 7-3.

Example 7-3 WLM install definition utility job output(1)

```
<IWMARIDU>
<IWMARIDU> Start WLM Install Definition Utility
<IWMARIDU>
<IWMARIDM> IWMARIDU options
<IWMARIDM> Force:               N
<IWMARIDM> Service policy name: WLMPOL
<IWMARIDM> Definition data set: 'YOUR.DEF.PDS'
```

```
<IWMARIDM>
<IWMARIDM> Global Variables Set
<IWMARIDM>
<IWMARIDM> IWMARIDU level:    LEVEL013
<IWMARIDM> Current WLM level: LEVEL013
<IWMARIDM>
<IWMARIDM> Calling IWMARZFL for OPEN request
<IWMARIDM>
<IWMARIDM> GetServiceDefinition
<IWMARIDM> Successful service definition open.
*IWMARIDM*
*IWMARIDM* InstallServiceDefinition
*IWMARIDM* Base definition was modified
*IWMARIDM* Attempting to install:     Sampdef
*IWMARIDM* Attempt by:                IBMUSER
*IWMARIDM* Attempt from system:       SYS1
*IWMARIDM*
*IWMARIDM* Last definition installed: itsor4
*IWMARIDM* Last installed on:         2002/05/29  13:20:50
*IWMARIDM* Last installed by:         HENRIK
*IWMARIDM* Installed from system:     SC65
*IWMARIDM*
*IWMARIDM* Force option is N
*IWMARIDM*
*IWMARIDM* Install failed, service definition was modified.
<IWMARIDM>
<IWMARIDM> Calling IWMARZFL for END request
<IWMARIDM>
*IWMARIDM* RC=324
<IWMARIDM>
<IWMARIDM> End WLM Install Definition Mainline
ISPD117
The initially invoked CLIST ended with a return code = 324
 SYS02149.T142423.RA000.IWMINSTL.R0101105 was preallocated (no free was done).
<IWMARIDU>
<IWMARIDU> End WLM Install Definition Utility
```

IWMINSTL job with activate

An installation job with **SVDEFPDS='YOUR.DEF.PDS'**, **FORCE='Y'**, and **POLNAME='WLMPOL'** parameters coded is shown in Example 7-4.

Example 7-4 Sample install definition utility to install WLM service definition and activate policy

```
//IWMINSTL   JOB MSGLEVEL=(1,1),REGION=0M
//*
//SETPARM1  SET   SVDEFPDS='YOUR.DEF.PDS'
//SETPARM2  SET   FORCE='Y'
//SETPARM3  SET   POLNAME='WLMPOL'
//*
//STEP1    EXEC  PGM=IKJEFT1B,
//         PARM='IWMARIDU F=&FORCE P=&POLNAME'
//SYSPROC  DD DSN=SYS1.SBLSCLIO,
//         DISP=SHR
//*
//ISPLLIB  DD DSN=ISP.SISPLPA,
//         DISP=SHR
//         DD DSN=ISP.SISPLOAD,
//         DISP=SHR
//ISPMLIB  DD DSN=ISP.SISPMENU,
```

```
//         DISP=SHR
//ISPPLIB  DD DSN=ISP.SISPPENU,
//         DISP=SHR
//ISPTLIB  DD DSN=ISP.SISPTENU,
//         DISP=SHR
//ISPSLIB  DD DSN=ISP.SISPSENU,
//         DISP=SHR
//ISPTABL  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1,5)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPPROF  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1,5)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPLOG   DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPCTL1  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//ISPLST1  DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//         DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1210)
//SVDEF    DD DSN=&SVDEFPDS,
//         DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
```

Job output

The job output produced by the WLM install definition is shown in a “progress log” displaying items such as input parameters, WLM and install utility levels, success and failure messages, return codes, and the final result. The result from our experiment is shown in Example 7-5.

Example 7-5 WLM install definition utility job output(2)

```
<IWMARIDU>
<IWMARIDU> Start WLM Install Definition Utility
<IWMARIDU>
<IWMARIDM> IWMARIDU options
<IWMARIDM> Force:              Y
<IWMARIDM> Service policy name: WLMPOL
<IWMARIDM> Definition data set: 'YOUR.DEF.PDS'
<IWMARIDM>
<IWMARIDM> Global Variables Set
<IWMARIDM>
<IWMARIDM> IWMARIDU level:    LEVEL013
<IWMARIDM> Current WLM level: LEVEL013
<IWMARIDM>
<IWMARIDM> Calling IWMARZFL for OPEN request
<IWMARIDM>
<IWMARIDM> GetServiceDefinition
<IWMARIDM> Successful service definition open.
*IWMARIDM*
*IWMARIDM* InstallServiceDefinition
*IWMARIDM* Base definition was modified
*IWMARIDM* Attempting to install:     Sampdef
*IWMARIDM* Attempt by:                IBMUSER
*IWMARIDM* Attempt from system:       SYS1
*IWMARIDM*
*IWMARIDM* Last definition installed: itsor4
*IWMARIDM* Last installed on:         2002/05/29  13:20:50
*IWMARIDM* Last installed by:         HENRIK
*IWMARIDM* Installed from system:     SC65
*IWMARIDM*
*IWMARIDM* Force option is Y
<IWMARIDM>
```

Archived

```
<IWMARIDM> InstallServiceDefinition
<IWMARIDM>
<IWMARIDM> Calling IWMARZFL for SAVE request
<IWMARIDM>
<IWMARIDM> SaveServiceDefinition
<IWMARIDM> Successful service definition save.
<IWMARIDM>
<IWMARIDM> ActivateServicePolicy
<IWMARIDM> Policy activation successful.
<IWMARIDM>
<IWMARIDM> Calling IWMARZFL for END request
<IWMARIDM>
<IWMARIDM> End WLM Install Definition Mainline
 SYS02149.T143159.RA000.IWMINSTL.R0101110 was preallocated (no free was done).
<IWMARIDU>
<IWMARIDU> End WLM Install Definition Utility
```

Syslog messages

The corresponding syslog messages are shown in Figure 7-6. Notice that jobs currently in the system go through a reclassification due to a new policy being activated.

```
IEF403I IWMINSTL - STARTED
*IAT2011 WLM RECLASSIFICATION IS IN PROGRESS
 IAT2016 WLM RECLASSIFICATION HAS COMPLETED
IWM001I WORKLOAD MANAGEMENT POLICY WLMPOL NOW IN EFFECT
IEF404I IWMINSTL - ENDED
```

Figure 7-6 IWMINSTL syslog entries

Activating a new service policy

To use the Install Definition Utility, configure the sample JCL (member IWMINSTL, shipped in SYS1.SAMPLIB) as directed in the prolog. Once the JCL has been prepared, it can be started from the command console or submitted as a batch job.

To activate a service policy with the Install Definition Utility, start or submit the sample JCL (member IWMINSTL in SYS1.SAMPLIB).

Once you have installed a service definition, you can activate it as a service policy. You can activate a policy either from the administrative application with the VARY operator command **VARY WLM,POLICY=xxxx**, or you can use the WLM Install Definition Utility, as follows:

- Install new service definition without activating a policy:

  ```
  s iwminstl,force=n,svdefps=your.definition.dataset
  ```

- Activate a new policy:

  ```
  s iwmninstl,polname=weekends,force=y
  ```

- Another example to install a service definition and activate a new policy:

  ```
  s iwminstl,force=y,svdefpds=your.definition.dataset,polname=yearend
  ```

Once you issue the command, there is an active policy for the sysplex. Systems will start managing system resources to meet the goals defined in the service policy.

Messages and commands not allowed in goal mode

If an IPS is specified in the IEASYSxx parmlib member, or WLM is attempting to switch to WLM compatibility mode, message IRA903I is shown, as illustrated in Figure 7-7 on page 99.

```
F WLM,MODE=COMPAT
IRA903I WLM COMPATIBILITY MODE IS NOT SUPPORTED
```

Figure 7-7 WLM compatibility mode not supported message

Similar results occur if you issue the following modify WLM commands, as shown in Figure 7-8.

```
F WLM,MODE=GOAL
IWM008I MODIFY WLM REJECTED, SYSTEM SC65 ALREADY IN WORKLOAD MANAGEMENT GOAL MODE
```

Figure 7-8 Modify WLM rejected message

In general, any compatibility mode function is rejected. Operator command examples include:

```
D DMN
E jobname,PERFORM=xxx
```

If you try to issue these or similar commands, or try to IPL with a IPS member, you will receive the message listed in Figure 7-7.

IEAOPTxx member

Beginning with z/OS V1R3, since WLM compatibility mode is no longer available, you can no longer use any of the IEAOPTxx options that were valid in compatibility mode only. The information left in the documentation is for reference purposes, and for use on backlevel systems.

SMF considerations

In particular, you should turn off SMF type 99 records. They trace the actions SRM takes while in goal mode, and are written frequently. SMF type 99 records are for detailed audit information only. Before you switch your systems into goal mode, you should make sure you do not write SMF type 99 records unless you want them.

If you do charge back based on SMF record type 30 or record type 72 records, you may need to update your accounting package since the records contain different fields for compatibility mode and goal mode.

WLM goal mode migration reference information

For a list of reference material including service offerings, as well as descriptions and downloads of the Goal Mode Migration Aid (GMMA) spreadsheet-based tool, refer to:

http://www.ibm.com/servers/eserver/zseries/zos/wlm/migration/migration.html

7.2 PAV dynamic alias management for paging devices

Prior to z/OS V1R3, WLM I/O priority applies only to *non-paging* I/O requests. I/O priority is an option which allows I/O priorities to be set for queued I/O requests when devices are busy.

Your system availability may be impacted when certain events, such as an SVC dump, cause a burst of paging activity that locks out or slows down page fault resolution for your critical workloads. New, combined ASM/WLM exploitation of parallel access volumes (PAVs) on the IBM Enterprise Storage Server® (ESS) allows the system to prevent page-outs from blocking page-in operations. Before we discuss this enhancement, let us review WLM I/O priority management and dynamic alias management.

7.2.1 Globally enabling WLM I/O priority management

When I/O priority management is enabled, then I/O scheduling priority is separate from CPU dispatching priority. This option is enabled globally (the effect is sysplex-wide) and is specified in the WLM ISPF panels. If I/O priority management is disabled, then I/O priorities are set similar to the CPU dispatching priorities. This option is disabled by default in your WLM service definition, but must be enabled in order to exploit dynamic alias management of parallel access volumes (PAVs). Static PAVs do not require I/O priority management to be turned on.

Velocity goal calculation revisited: We strongly recommend that you apply PTFs for APAR OW47667. This APAR enhances the way WLM calculates velocity calculations by eliminating disconnect times from I/O using samples during the execution velocity calculations. Disconnect time is no longer counted as productive I/O time. It is also not counted as I/O delay because there is nothing WLM can do to reduce disconnect time.

This change will affect achieved velocities for systems which have the I/O priority queueing option set to ON in the WLM policy. Achieved velocities will be the same or lower, depending on how much disconnect time the service class experiences. Customers should review velocity goals in their WLM policy and adjust downward if needed.

WLM counts disconnect time along with connect time as productive I/O time (I/O using samples) which increases the velocity and, therefore, decreases the performance index (PI) for a service class. A lower PI makes it appear the service class is performing better than if disconnect time were not counted. This can allow a service class to become a donor of I/O resource when the devices it is using are performing poorly.

Prior to OW47667, the formula was:

I/O Using = Connect Time + Disconnect Time

After you implement OW47667, the formula is:

I/O Using = Connect Time

Prior to this APAR, many installations choose not to enable I/O priority management. Since applying this APAR changes the way execution velocities are calculated, you need to revisit your service class definitions for proper execution velocity values in some of your goal settings. The consequence of applying the APAR is that your execution velocities may be somewhat *lower*.

RMF Postprocessor report

RMF workload activity Postprocessor reports may serve as guidance for you to establish an understanding of your actual execution velocities calculated by WLM before and after this APAR is implemented. RMF gives you migration help with I/O priority management in general, as it shows you the actual velocity if I/O using and I/O delay samples would be considered, regardless of how WLM calculates them. This migration help is in the WLMGL report, which is the goal mode version of the workload activity Postprocessor report.

7.2.2 Individually enabling dynamic alias management on a per volume basis

While you *globally* enable or disable parallel access volumes by enabling or disabling dynamic alias management on the WLM ISPF panel as described, you must also *individually* enable or disable dynamic alias management for your devices via HCD. Do this by specifying **WLMPAV=YES** or **NO** in the HCD definition corresponding to the particular device.

Observe that there is no consistency checking for dynamic alias management between different systems in a sysplex. If at least one system in the sysplex specifies **WLMPAV=YES** for a device, then dynamic alias tuning will be enabled on that device for *all* systems in the sysplex, even if other systems have specified **WLMPAV=NO**.

We recommend that you do not use dynamic alias management for a device unless all systems sharing that device have dynamic alias management enabled. Otherwise, WLM attempts to manage alias assignments without taking into account the activity from "non-participating" systems. For more information about HCD considerations, refer to *z/OS Planning: Workload Management*, SA22-7602.

7.2.3 Enabling dynamic alias management for paging devices

As of z/OS V1R3, performance and availability of your paging subsystem may be enhanced by exploiting parallel access volumes on the IBM Enterprise Storage Server (or other similarly defined 2105 devices). PAV devices lead to higher page I/O throughput, reduced contention, and less lockout times. This is, for example, the case when you write a dump. The purpose of this enhancement is to reduce the impact of heavy I/O on page I/Os (such as when a dump is written).

Enhancements in z/OS V1R3 allow Auxiliary Storage Manager (ASM) to state the minimum number of PAV dynamic aliases needed for a paging device. The more page data sets you have on the volume, the more alias devices will be set aside. A minimum number of aliases will be preserved by the system even in cases where aliases are being moved to reflect changes in workload. You do not need to define any of this; simply enable the volume for dynamic alias management, as previously described for your paging devices.

Note: All systems in your sysplex must be on a z/OS V1R3 or above level in order for this support to work.

7.3 WLM independent enclave service class reset

Prior to z/OS V1R3, there is no mechanism for you to change the service class of an executing WLM enclave, other than changing the classification rules in the WLM policy. This presents operational problems in managing subsystems that exploit WLM enclaves, such as DDF or WebSphere. For example, if you have a "runaway" DDF query, an operator cannot change the service class of the enclave that represents the query to reduce its resource usage without changing the policy. Operators often do not have the authority to change the WLM policy.

The WLM enclave service class reset function, discussed in "Resetting independent enclaves" on page 103, provides a mechanism to change the service class of original independent enclaves or to quiesce enclaves. There is no operator command interface provided for this reset function, because there is no single name (other than a hex-digit token) that can be used to uniquely identify an enclave for an address space.

Before we discuss this enhancement, let us review the concepts of an *enclave,* a *multisystem enclave*, and *dependent enclaves* versus *independent enclaves*.

7.3.1 Enclaves

An enclave represents a transaction that can span multiple dispatchable units of work (enclave SRBs and enclave TCBs) in one or more address spaces. All units of work are reported and managed as a single entity. The address space creating an enclave is referred to as the *enclave owner*. An address space can create independent enclaves, which is a completely new SRM transaction of their own, or dependent enclaves, which are a continuation of an existing address space transaction. An address space can own multiple enclaves at the same time (for example, Distributed Data Facility, or DDF).

When tasks or SRBs within an address space have joined an enclave, the address space is managed towards the goal and the importance of the enclave's service class. Such an address space is referred to as an *enclave server address space*. An enclave server address space can serve multiple enclaves at the same time. It will be managed towards the service class of the enclave with the most stringent resource requirements.

Examples of products that use enclaves include:

- DB2 V4 and upwards for Distributed Data Facility (DDF)
- DB2 V5 and upwards for Stored Procedures
- DB2 V5 and upwards for sysplex query parallelism
- IBM HTTP Server for OS/390 (IWEB) scalable Web server
- WebSphere
- MQ/Series workflow request
- SOM® client object class binding requests
- LAN Server for OS/390 (LSFM)

Dependent enclave

A dependent enclave represents a transaction that is a continuation of an existing transaction (for example, a TSO user or a batch job). A dependent enclave starts in an address space and “spawns” into an dependent enclave. A dependent enclave is used when you have an existing address space defined with its own performance goal that is extended to programs running under dispatchable units in other address spaces.

The owner of the dependent enclave is the home address space (TSO user or batch job) that created the dependent enclave. A dependent enclave derives its performance goal from the owning address space. The enclave's service class is inherited from the owning address space service class.

Independent enclave

An independent enclave represents a complete, independent transaction. It is usually a transaction that is arriving from your network. Once a transaction is submitted, an enclave is created, after which it is classified into a service class and then executes. The home address space that creates the independent enclave, for example DDF, becomes the owner. CPU service consumed by the enclave is accumulated in the SMF Type 30 record of the home address space and the SMF Type 72 (RMF) record of the enclave's service class.

Note: All CPU service consumed by the dependent and independent enclaves is accumulated in the SMF Type 30 record of the owning address space and the SMF Type 72 record of the owning address space's service class.

For more information about dependent versus independent enclaves, refer to *z/OS Programming: Workload Management Services*, SA22-7619.

7.3.2 Resetting independent enclaves

z/OS V1R3 allows you to reset the service characteristics of independent enclaves. Through SDSF or via commands, individual independent enclaves can be **RESET** to another service class, **QUIESCE**d, or **RESUME**d. This enhancement allows you to have more control of independent enclave transaction behavior in your environment. This support very much resembles the existing support for address spaces.

> **Note:** Dependent enclaves are reset through their owning address spaces. You are not allowed to reset into the system service classes SYSSTC and SYSTEM.

For an example on how you may reset, resume or quiesce an enclave using overtypes in the SDSF ENC panel, refer to Figure 7-9. SDSF has been enhanced with new SrvClass (service class) and Quiesce columns. New action characters **R** (Resume) and **RQ** (Quiesce) are now allowed in the NP column.

Figure 7-9 SDSF ENC Panel: Enclave manipulation using overtyping

> **Attention:** When resetting dependent enclaves, the owning address space is reset by SDSF. WLM then resets all dependent enclaves that this one address space owns, along with the address space.

For more information about SDSF enhancements refer, to “SDSF enhancements” on page 85.

In summary, prior to the WLM independent enclave service class reset enhancement, the service class of an executing enclave could not be changed except by changing the classification rules in your WLM service definition. The existing **RESET** command cannot be used because enclaves do not have the equivalent of an address space name. This could make it difficult to react to runaway transactions in an enclave-exploiting subsystem, such as runaway DDF transactions.

WLM independent enclave service class reset enhancement allows operators to reset the service characteristics of an enclave transaction in case it “runs away”. You can either select a service class with a less restrictive goal, or the enclave can be quiesced. By demoting enclaves, you can prevent impacts on your overall system performance. Similarly, the same means can be applied to promote an enclave to “get it out of the way” fast by resetting it to a service class with less restrictive goals.

7.3.3 Multisystem enclave support

With multisystem enclave support, enclaves run in multiple address spaces spanning multiple systems within your Parallel Sysplex. As in a single system enclave, the work is reported on and managed as a single unit. USS Parallel Environment to run parallel jobs is an example of a product that uses multisystem enclaves.

With all tasks of the job running in the same enclave, WLM can manage all of the work to a single performance goal. As a prerequisite, you need to define a CF structure for multisystem enclave support called **SYSZWLM_WORKUNIT**. This function also requires CFLEVEL 9 or above CFCC support.

Refer to *z/OS Programming: Workload Management Services*, SA22-7619, for more information on multisystem enclaves.

Enclave registration/deregistration services

New WLM registration and deregistration services (IWMEREG, IWMEDREG) are introduced to avoid premature deletion of enclaves. Use IWMEREG to register interest in an enclave, and IWMEDREG to deregister interest. Enclaves with interested subsystems will not be deleted, even if the enclave creator invokes the IWMEDELE service. Enclaves are deleted when there are no interested subsystems and the IWMEDELE service has been invoked, regardless of the order in which they occur.

New function APAR OW46363 is included in z/OS V1R2 and is applicable to OS/390 V2R7 and higher releases. It addresses problems when one authorized subsystem may delete an enclave prematurely, while another subsystem still tries to use this enclave (for example, it schedules additional enclave SRBs, not knowing that the enclave does not exist anymore). This problem can become more prevalent when multiple different subsystems are involved in serving an enclave transaction. With the new registration and deregistration services, using enclaves in such subsystems becomes easier and less error-prone, since no protocol is required that determines who is allowed to delete the enclave and when.

7.4 WLM support for sub-capacity pricing

The RMF Monitor III is enhanced to provide online monitoring of service unit definitions versus actuals, as well as LPAR CPU utilization information. This enables you to effectively deal with the flexible load balancing and peak load handling in environments using sub-capacity pricing, as well as in IBM License Manager environments.

7.5 WLM enqueue management enhancements

SRM provides an interface (Sysevent EnqHold/EnqRlse) that can be used by resource managers. The current WLM/SRM enqueue promotion support is strictly efficiency-based; no consideration is given to your WLM policy or the business importance (BI) of waiters for resources under contention. The current support also considers GRS-managed resources such as ENQs, DEQs, and RESERVEs.

The WLM enqueue management enhancement in z/OS V1R3 assists in the process to ensure critical work on your system such as DB2 is not blocked by resources such as RACF databases, IRLM locks, or latches just because the blocker has low priority, is possibly swapped out, or not receiving services for any number of reasons.

Awareness of contention situations in authorized resource managers other than GRS (for example) is implicitly part of this solution. WLM enqueue management in z/OS V1R3 establishes the necessary WLM infrastructure and it is expected that authorized resource managers over time will support this effort.

First, SRM will give better treatment to the holder of resources under contention in the hope that the holder will release that resource *each time* SRM is notified of contention caused by that holder, rather than only the first time. With this support, SRM may take action whenever contention is reported even if the blocker has been helped before.

Second, SRM will gather additional information about the users of the contention notification interfaces in order to make it easier to identify situations where a resource manager does not pair its notifications of contention beginning and ending.

Third, SRM will change its implementation to make it less vulnerable to third-party software modifying its control information directly.

This is primarily an internal enhancement and there are no new actions required from a customer implementation standpoint.

SMF type 30 is expanded by a field in the processor accounting section that contains the CPU time consumed while enqueue promoted. For more information, refer to *z/OS System Management Facilities (SMF)*, SA22-7630.

7.5.1 ERV parameter in IEAOPTxx

The ERV parameter specifies the number of CPU service units that an address space or enclave is allowed to absorb when it is possibly causing enqueue contention. During this "enqueue residency" time, the address space or enclave runs with the privileged dispatching priority (coded on the PVLDP keyword of the IEAIPSxx parmlib member). Also during this interval, the address space (including the address space associated with an enclave) is not considered for swap-out based on recommendation value analysis. If you are running in Workload Manager goal mode, the address space or enclave runs with a high enough priority to guarantee the needed CPU time.

ERV is in effect for an address space or enclave that meets one of the following criteria:

- The address space or enclave is enqueued on a system resource needed by another address space.
- An authorized program in the address space or enclave obtains control of the resource (even if another address space does not need that resource) as a result of issuing a reserve for a DASD device that is SHARED.

General recommendation for ERV setting

Until WLM enqueue management is fully operational in your environment, the following recommendations that may help you to avoid certain enqueue-related contention situations in your environment:

- The default value for ERV is 500, which indicates that if an address space is swapped out holding an enqueue, it should be swapped back in until it has either released the enqueue, or has used 500 CPU service units. However, empirical analysis has shown that this may not be enough in today's environments.

  Therefore, you should ensure that work, on average, has enough CPU service time to release the enqueue before being swapped back out again. Trial and error may be one approach to finding a more optimal ERV value; however, it is probably a safe bet to bump this value up to 10000 or higher.

  The net effect will be that low importance (or discretionary work, to a lesser extent) is stopped for the more important workload.

- To avoid enqueue lockout situations where low priority or discretionary workload may be swapped out while holding an enqueue, consider the following. In both WLM goal as well as compatibility mode, IEAOPTxx parmlib member parameters such as the enqueue residency value (ERV) value specification are still valid. The ERV keyword defines the amount of CPU service (in terms of raw, unweighted CPU service units) that an address space is allowed to receive before it is considered for a workload recommendation swap out. The parameter applies to all swapped-in address spaces that are enqueued on a resource needed by another user.

  This process is sometimes referred to as "enqueue promotion". Enqueue promotion is limited by the promotion interval, governed by your setting of the ERV value. This limitation is enforced in order to prevent abuse of this facility by end users who could intentionally create contention in order to receive better service.

Enhanced enqueue management

With z/OS V1R3, enqueue management has been improved by a more sophisticated enqueue promotion algorithm.

An address space or enclave is promoted in terms of dispatch priority when it holds a resource that another address space or enclave wants to have. The resource manager indicates this situation to SRM through an ENQHOLD-sysevent. By promoting the address space or enclave for a limited amount of time, the system hopes that the holder gives up the resource faster than it usually would with a lower-dispatching priority. Also, while being promoted, the system ensures that the address space or address space associated with the enclave is not swapped-out.

When a contention disappears, the resource manager notifies SRM through an ENQRLSE-sysevent.

The enqueue promotion interval, shown in Figure 7-10 on page 107, can be set by the installation through the ERV-option in IEAOPTxx-parmlib member. The ERV-option specifies the CPU service units that an address space or enclave can absorb while it is promoted before the promotion is taken back.

In goal mode, the enqueue promotion dispatch priority is determined dynamically at every policy adjustment interval (10 sec). It can change based on available processing capacity and amount of high dispatch work in the system. Address spaces are promoted to that priority if their current dispatch priority is not already higher than that.

Before z/OS V1R3, an address space or enclave was only promoted once for a given resource. If the resource was not returned before the expiration of the promotion interval, it was just a “nice try”.

Figure 7-10 Enhanced enqueue management

With z/OS V1R4, an address space or enclave is promoted anew by SRM for every contention indication as told by the resource manager. Also, while there is an outstanding contention release notification, quiesced work is kept swapped-in as long as there is ready work to run. This includes address spaces associated with enclaves.

7.6 WLM WebSphere performance enhancement

This enhancement, introduced in WLM with z/OS V1R3, benefits short-running WebSphere transactions. This small enhancement in WLM continues to improve functionality and performance. By using cell pools rather than explicit storage obtains for single work requests added to a WLM work queue, the creation and the process of inserting a work request can be accomplished faster. Another net effect is that a couple of modules implementing WLM services use cell pools rather than storage obtains and so achieve the same benefits.

Also refer to 7.11, “WLM temporal affinity for WebSphere Application Server” on page 117 for a discussion on another WLM enhancement for WebSphere.

7.7 WLM batch initiator balancing enhancements

WLM has offered the possibility to manage batch initiators since OS/390 V2R4. With this support, WLM can control the number of batch initiators on each system in a sysplex dynamically. New initiators are preferably started on low-utilized systems to obtain some degree of batch workload balancing.

Before we look at the enhancements introduced in z/OS V1R4, let us discuss the concept of WLM batch initiator management.

7.7.1 WLM batch initiator management

WLM has the capability to dynamically manage the number of batch initiator address spaces serving specific classes of batch jobs that have been identified and submitted through JES. These jobs become assigned to WLM service classes, allowing WLM to start new initiators on demand, to meet the performance goals of that category of work.

The system resources manager (SRM) relies on additional work queuing delay information which is provided by JES2 using programming interfaces supplied by WLM. JES2 uses the WLM-provided services during initialization to connect to WLM via the IWMCONN service. This initial contact with WLM during initialization occurs by specifying the following options:

WORK_MANAGER(YES) This parameter identifies the role that the connecting address space processes.

SUBSYS(JES) This parameter identifies the set of WLM classification rules to be used.

SUBSYSNM(name) This parameter, when combined with the SUBSYS-TYPE value, uniquely identifies the subsystem instance.

This information physically binds the JES2 address space to WLM, enabling use of WLM work classification services by JES2 and establishing WLM address space termination recovery. The initiator code allows WLM to intercept job scheduling requests. This makes it possible to dynamically reassign initiator address spaces to work queues that require additional throughput.

You can have as many job classes under WLM-managed mode as you choose. Once there, you have the flexibility, through operator command or SDSF, to dynamically switch any class back to JES2-managed mode. WLM-managed initiators are created as they are needed, depending on:

- The presence of backlog work in WLM-managed classes
- Adequate capacity to do more work
- The service class goals associated with jobs in the backlog

7.7.2 Queueing jobs for execution

Batch jobs are selected for execution by initiators, each one running in a separate address space. Initiators are controlled by JES or by WLM. Which one controls the initiators is determined by the job class and by the MODE= parameter (JES or WLM) on the JES2 JOBCLASS statement. JES2 now maintains two different queue organizations for all jobs awaiting execution, as shown in Figure 7-11 on page 109:

1. All jobs are queued by job class, priority, and the order in which they finished conversion. This is the queue from which JES2-managed initiators select jobs for execution.
2. Jobs awaiting execution in WLM managed job classes are also queued by their WLM assigned service class in the order they were made available for execution.

If jobs are scheduled by JES2-managed initiators, the Class Queue Heads queue is used. For jobs scheduled to WLM-managed initiators, jobs are queued off of the Service Class Queue Heads, as shown in Figure 7-11 on page 109. HOTBATCH and NORMAL are service classes.

Figure 7-11 JES2 and service class queues

Note: The dotted lines in Figure 7-11, starting from the service class queue heads, indicate the order in which the jobs are selected from the queues for WLM-managed initiators. The selection of the jobs from the queues are in “first-in, first-out” order and not in priority order, as for JES2-managed initiators. The solid lines indicate the order of job selection if all the initiators are JES2-managed.

7.7.3 Classifying jobs

Following JCL conversion, JES2 calls the WLM IWMCLSFY service to classify the job. Passed to WLM work classification are many of the job attributes extracted by the conversion phase, as shown in Figure 7-12 on page 110. A service class is passed back to JES2 for the job.

```
IWMCLSFY USERID=$DWORK2,  Owning userid
      TRXCLASS=JQXJCLAS,  Job class
      TRXNAME=JQEJNAME,   Job name
      ACCTINFO=(R6),      Accounting string
      ACCTINFL=$DWORK+4,  Address of acctg length
      PERFORM=(R8),       Address of Perf group
      PRIORITY=$DWORK,    Priority
      CONNTKN=(R4),       WLM connect token
      SCHEDENV=JQASCHE,   Scheduling environment
      SERVCLS=JQASTOK,    Service class token (rtn)
      SRVCLSNM=JQAWSCN,   Service class name  (rtn)
      SRMTOKEN=JQAXSRMT,  SRM token
      SUBCOLN=$XCFGPNM,   XCF group name (node name)
      RETCODE=WLMRETCD,   Save return code
      RSNCODE=WLMRESCD,     and reason code
      MF=(E,$CLSFY)       Address of list form
```

Figure 7-12 Job classification service attributes

Note: The parameters passed by the IWMCLSFY service are the same as the JES work qualifiers shown in Figure 7-13 on page 111. You create classification rules by using any—or as many—of the work qualifiers as you want.

7.7.4 Service classification rules

There is one set of classification rules in the service definition for a sysplex. They are the same regardless of what service policy is in effect; a policy cannot override classification rules. You should define classification rules after you have defined service classes, and ensure that every service class has a corresponding rule.

The service class to be passed back to JES2 for the job is determined based on the classification rules used in the WLM policy. The JES subsystem classification rules are determined by the JES work qualifiers (shown in Figure 7-13 on page 111) and the rules created using these qualifiers to determine a service class.

Archived

```
           Qualifier Selection Row 1 to 12 of 15
Command ===> ________________________________

Select a type with "/"

Sel  Name  Description
 _   AI    Accounting Information
 _   PF    Perform
 _   PFG   Perform Group
 _   PRI   Priority
 _   PX    Sysplex Name
 _   SE    Scheduling Environment
 _   SI    Subsystem Instance
 _   SIG   Subsystem Instance Group
 _   SSC   Subsystem Collection
 _   TC    Transaction Class
 _   TCG   Transaction Class Group
 _   TN    Transaction Name
 _   TNG   Transaction Name Group
 _   UI    Userid
 _   UIG   Userid Group
```

Figure 7-13 JES work qualifiers

Figure 7-14 shows a simple single rule as an example. For job class A (Type=TC, Name= A in Figure 7-14.), a service class of BATCHLO is assigned. If the job being classified in class A has a priority greater than 9, the BATCHHI service class is assigned. In this example with only one rule, all other jobs are assigned the default service class BATCHME.

Note: PRI is a sub-qualifier of TC.

```
               Modify Rules for the Subsystem Type       Row 1 to 2 of 2
Command ===> ________________________________________ SCROLL ===> PAGE

Subsystem Type . : JES         Fold qualifier names?   Y  (Y or N)
Description  . . . Batch work

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                          More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service     Report
                                          DEFAULTS: BATCHME     ________
 ____  1  TC         A        ___                  BATCHLO     ________
 ____  2    PRI        >9     ___                  BATCHHI     ________
```

Figure 7-14 Example batch classification rule

7.7.5 Steps required for batch initiator management

Batch initiator management allows for job classes to be initiated by JES2-managed initiators and service classes to be initiated by WLM-managed initiators. In preparation for this support, the following steps must be considered:

- Make sure the WLM couple data set is at a correct level.

- For job classes to be managed by WLM, the job classes must be defined to JES2 as MODE=WLM on the JES2 JOBCLASS initialization statement.
- Service classes for the WLM-managed batch must be defined in a WLM policy.

The service policy must be activated.

7.7.6 Batch initiator management limitations

WLM-managed initiators are started and stopped under the control of WLM and SRM. Each initiator belongs to a service class. Batch initiator management was introduced in JES2 with OS/390 V2R4. The limitations before z/OS V1R4 were the following problems:

- Balancing of initiators between systems was done only when new initiators were started.
- When initiators were available to select jobs, those jobs were started on any system independent of the system load (assuming that they have no system affinity). As a result, batch jobs could start on fully loaded systems, even if other systems had more free capacity.

7.7.7 z/OS V1R4 enhancements

To correct the problems of previous releases, new algorithms for rebalancing initiators in z/OS V1R4 are as follows:

- More aggressively reducing the number of initiators on constrained systems
- Starting new initiators on low usage systems
- Checking for potential rebalancing every 10 seconds

Important: WLM initiator balancing enhancements are only available on z/OS V1R4 systems. These new enhancements become automatically active when there are job classes in MODE=WLM. See “Queueing jobs for execution” on page 108.

Stopping initiators on constrained systems

A check is made every ten seconds to check for conditions of when to stop initiators for balancing jobs in the sysplex, as follows:

- A constrained system is now considered to be one with a one-minute average of used CPU greater than 95%.
- Jobs are available in a batch queue that is eligible for balancing. A batch queue is eligible if there are no pending stops of initiators. A pending stop could occur, if a stop was requested—but the stop does not occur immediately if the job running under the initiator has not finished.
- Select an eligible service class queue with the lowest importance. If several queues have lowest importance, take the one that has not been balanced for the longest time.
- The system has the most accumulated max demand of all constrained systems.
- There is another unconstrained system, which has enough storage and CPU available to start a new initiator of the same queue for the same service class.
- The other system would have not more than 93% used CPU when starting such an initiator.
- The other system is z/OS V1R4 or higher.

If the reduction of initiators on constrained systems is very aggressive, jobs with affinity to that system must not be treated worse than before. If there are such jobs waiting, then WLM keeps enough initiators on constrained systems.

Starting new initiators

The conditions to start new initiators are as follows:

- If there is a request waiting on the queue, then start as many initiators as will fit into the available CPU and memory up to a maximum of 5. Previously this value was 1.
- New initiators are only started if there was no initiator balancing for the same queue in the last 30 seconds.

Note: In case of mixed JES2 releases, the reduction of initiators is only done on systems with z/OS V1R4 or higher. On systems with older JES2 releases, this function is not active.

Jobs with system affinity

Jobs that have a system affinity to a constrained system will be treated the same as in previous releases. So if the reduction in initiators on that system is very aggressive, these jobs will execute as before. To make sure that this works as before, WLM will ask JES2 about jobs waiting for execution that have system affinities. If these jobs exist, WLM will keep enough initiators on constrained systems to execute the jobs.

Note: Jobs with affinity to non-constrained systems can be treated even better than before, as initiators on non-constrained systems are started more rapidly than before if there is enough idle capacity.

7.8 Performance block application state reporting for enclaves

Performance block (PB) state reporting for enclaves when using WebSphere is an enhancement of WLM reporting that allows report-only performance blocks (PBs) to be associated with enclaves, and that supports performance block reporting for multi-period classes for all goal types.

Prior to this support, there was a restriction that a PB could only be associated with an address space in a service class with a single period. Now a report-only PB can be associated with an enclave or an address space in a service class with multiple periods. There was also a restriction that a PB could only be associated with an address space in a service class with a response time goal.

Now a report-only PB can be associated with an enclave or an address space in a service class with any goal type.

7.8.1 RMF support

RMF reports about additional WLM performance block (PB) states in the Work Manager Delays report are introduced in z/OS V1R4. RMF provides the monitoring support by providing:

- Subsystem work manager delays for service classes with goal types other than response time goals
- Subsystem work manager delays for multi-period service and report classes
- A new active state for Monitoring Environments to be able to distinguish between active subsystem and active application
- New waiting states for Monitoring Environments such as:
  - Waiting for SSL thread

– Waiting for regular thread
– Waiting for registration to work table

Note: SMF record 72 type 3 and Monitor III table ERBRCDG3 is extended to hold the new ACTIVE and WAITING states. The Postprocessor WLMGL and Monitor III SYSWKM report formats the additional values. In addition, the problem with values >100% in the SUBSYSTEM DELAY section of the Postprocessor WLMGL report is resolved.

Subsystems using enclaves such as CICS and IMS have been able to present subsystem states for reporting and analysis purposes for many releases. Other applications, such as WebSphere, lack detailed performance reporting information about subsystem Work Manager delays.

The performance block (PB) state reporting for enclaves allows WebSphere Application Server to provide more granular performance reporting.

Note: This support is introduced in z/OS V1R4 and can be retrofitted to z/OS V1R2 and z/OS V1R3 by applying APAR OW51848 for z/OS and APAR OW52227 for RMF.

The external interfaces for monitoring environments are changed to allow WLM to establish report-only performance blocks (PBs). The external interfaces are changed to enable association of report-only PBs with enclaves and address spaces. The sampling for report-only performance blocks is extended. There are now sampling for velocity goals, discretionary, multi-period classes, and response time goals.

The new functions allow specification of several new states, active and waiting states. The new active states distinguish whether the subsystem, or an application called by the subsystem is executing the work request. The new wait states are: waiting for security thread (SSL); waiting for executing threat; waiting for registration.

Sample RMF report

The new states are displayed by RMF as shown in Figure 7-15 on page 115. Here you see detailed breakdown information for subsystems based on state samples. The report shows the two types of active states, active subsystem and active application. The wait breakdown shows the resources with the highest values.

```
                                        W O R K L O A D   A C T I V I T Y


                                     POLICY ACTIVATION DATE/TIME 03/21/2002 12.51.18
-------------------------------------------------------------------------------------------------------------------- SRV. CLASS PERIOD(S)

REPORT BY: POLICY=VICOM1     WORKLOAD=WRKLOAD1   SERVICE CLASS=MEDIUM     RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=3
                                                 CRITICAL     =NONE

TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT   --DASD I/O--   ---SERVICE----   --SERVICE RATES--   PAGE-IN RATES    ----STORAGE----
AVG      0.20   ACTUAL             11.716   SSCHRT   0.2   IOC         0    ABSRPTN        61   SINGLE     0.0   AVG        0.00
MPL     0.20   EXECUTION          11.716   RESP     7.6   CPU      3687    TRX SERV       61   BLOCK      0.0   TOTAL      0.00
ENDED       5   QUEUED                  0   CONN     3.9   MSO         0    TCB           0.1   SHARED     0.0   CENTRAL    0.00
END/S    0.02   R/S AFFINITY            0   DISC     2.4   SRB         0    SRB           0.0   HSP        0.0   EXPAND     0.00
#SWAPS      0   INELIGIBLE              0   Q+PEND   1.3   TOT      3687    RCT           0.0   HSP MISS   0.0
EXCTD       0   CONVERSION              0   IOSQ     0.0   /SEC       12    IIT           0.0   EXP SNGL   0.0   SHARED     0.00
AVG ENC  0.20   STD DEV                 0                                    HST           0.0   EXP BLK    0.0
REM ENC  0.00                                                               APPL %        0.0   EXP SHR    0.0
MS ENC   0.00

           RESP  ------------------------------- STATE SAMPLES BREAKDOWN (%) ------------------------------  ------STATE------
SUB    P   TIME  --ACTIVE-- READY IDLE  ----------------------------WAITING FOR----------------------------  SWITCHED SAMPL(%)
TYPE        (%)   SUB  APPL            SSLT WORK REGT DIST                                                  LOCAL SYSPL REMOT
CB    BTE   0.0   0.0   0.0   0.0  0.0   0.0  0.0  0.0  0.0                                                  0.0   0.0   0.0
CB    EXE  99.7   1.7  18.6   0.0 20.3  20.3 18.6 16.9  3.4                                                  0.0   0.0   0.0
```

Figure 7-15 RMF Workload Activity Report showing application state reporting

Response time breakdown RMF reporting enhancements

Work managers post states in WLM performance blocks (PBs). WLM samples such PBs, and records states for all transactions in the BTE or EXE phases. A state is active, ready, or idle. Samples of transactions are taken dynamically, and the count is recorded at the end of the measuring interval.

Prior to z/OS V1R4, RMF calculates response time breakdown values as a percentage of average transaction response time in the response time breakdown section in, for example, the RMF Workload Activity - Goal Mode reports. They are only created for transaction service classes (CICS or IMS).

Similar information is shown for any service class, report class, and workload group in RMF Monitor III reports on response time breakdown in the GROUP report.

The response time is calculated when a transaction completes. This can result in percentages greater than 100 when samples are included for long-running transactions which have not completed in the gathering interval. Similar concerns exist if, in subsequent intervals, few samples will be included, but all of the elapsed time. Never-ending transactions (such as CICS CMT transactions) will have their sampling included, but not their elapsed time (since they never complete).

Percentages greater than100 in the breakdown section are avoided starting with z/OS V1R4 by showing the state values as percentages of the total transaction samples, instead of percentages of response time.

For more information about how to interpret response time breakdown reporting in RMF reports, refer to *z/OS RMF Performance Management Guide*, SC33-7992.

This functionality is available as an SPE delivered with APAR OW52227.

Archived

7.8.2 New classification qualifiers for WebSphere (CB) subsystem

Application environments may be defined and controlled by WLM for DB2, IWEB, and CB (WebSphere Application Server) workloads. WLM can dynamically manage the number of server address spaces for WebSphere. The following classification qualifiers are added for subsystem type CB (WebSphere App Server), starting with z/OS V1 R4:

TN Transaction name

TC Transaction class

These added classification qualifiers further compound the extensive list of classification qualifiers for WLM.

The classification qualifiers TN and TC were previously available for other subsystems. For an overview of available classification qualifiers, see *z/OS MVS Planning: Workload Management*, SA22-7602-03.

7.9 WLM msys for Setup enhancement

Prior to this enhancement, msys for Setup was unaware of your existing WLM CDS attributes. Msys for Setup only used a predefined size to allocate the WLM CDS. This did not respect the possibly larger size of an existing WLM CDS you might already have allocated.

WLM msys for Setup exploitation provides a function to query the size of the WLM CDS and to store these values into the msys for Setup LDAP management directory. The allocation of the WLM CDS is supported by msys for Setup via the Parallel Sysplex wizard. With this new function, the size of an existing WLM CDS can be queried from msys for Setup and the allocation values can be adjusted if necessary.

These WLM CDS allocation values are stored into the msys LDAP management directory:

- Format level of WLM CDS
- Number of policies
- Number of workloads
- Number of SCs
- Number of application environments
- Number of scheduling environments
- Number of service definition extensions
- Number of classification rule extensions
- Number of application environment extensions
- Number of scheduling environment extensions

For more information on msys for Setup, go to:

http://www.ibm.com/eserver/zseries/msys

7.10 WLM support for ESS FICON and I/O priority management

APAR OW51126, included in z/OS V1R3 and applicable to OS/390 V2R8 and above, allows I/O priority management to work correctly with an ESS connected via FICON channels. When I/O priority management is being used with a FICON-channel-attached ESS, I/O priorities may not be adjusted quickly enough. With such DASD devices, connect time can elongate for

I/O requests that are multiplexed across the FICON channel. With FICON ESS devices, part of the connect time can actually be a delay due to multiplexing. The use of connect time for these devices results in an artificially low performance index PI, thus not giving priority to jobs that need help, such as adjusting I/O priorities quickly enough after I/O contention occurs.

Prior to this enhancement, WLM interprets all connect time as productive I/O time and is not aware of what portion of this time is delay due to multiplexing. With APAR OW51126, WLM uses an artificial constant connect time per FICON I/O request of 1ms, rather than using the measured connect time when calculating velocity and performance index (PI). This prevents WLM from overestimating execution velocity for service classes using FICON channels with the ESS, and allows I/O priority management to be effective in this environment.

Note that this time is only seen by WLM for calculating the PI. Other measurements, such as SMF and RMF data, will still see the actual connect time.

Also, refer to 7.2.1, “Globally enabling WLM I/O priority management” on page 100 for a discussion on how APAR OW47667 corrects the way execution velocity is calculated by removing disconnect time from WLM I/O using samples.

7.11 WLM temporal affinity for WebSphere Application Server

New function APAR OW45238, included in z/OS V1R2 and applicable to OS/390 V2R8 and higher releases, provides temporal affinity support for WebSphere Application Server (WAS).

Temporal Affinity provides the capability for exploiters of WLM queueing services to route work requests to specific server regions which maintain data required by the application to execute the work requests. In addition, the application has the ability to mark a server space so that it is not automatically terminated by WLM. The support is introduced especially for WebSphere/390 EE 4.0. There are two cases in a WebSphere environment where it is necessary to route request a specific server region:

- The first is for fully distributed objects whose state lives only in the virtual storage of the server region where they are activated.
- The second is for a transaction that is started by an EJB but is not committed before the request is completed, and the control is returned to the client. The transaction that is left behind in the server region, and the object that started it, are tied to that server region.

In both cases, it is necessary for the application to route subsequent requests to the server region where the object resides. In addition, the application must have the ability to tell WLM *not to remove* the server region as long as this temporal affinity exists.

7.11.1 SDSF support

With SDSF APAR PQ43534, the operator has the ability to identify the server region with temporal affinities through the DA screen, as shown in Figure 7-16 on page 118.

```
SDSF DA WLM4  WLM4     PAG    0 SIO    12 CPU   3      LINE 1-5 (5)
COMMAND INPUT ===>                                          SCROLL ===> CSR
PREFIX=ASAH*  DEST=(ALL)  OWNER=**
NP   JOBNAME  U% Workload SrvClass SP ResGroup Server Quiesce  ECPU-Time  ECPU%
     ASAHIP1A  3 STC      STCDEF    1          YES                 2.90   0.00
     ASAHIP1A  3 STC      STCDEF    1          YES                 2.77   0.00
     ASAHIP1A  3 STC      STCDEF    1          TEMP-AFF            3.16   0.00
     ASAHIQ11  3 BATCH    BTCHDEF   1          NO                  1.38   0.00
     ASAHIP1A  3 STC      STCDEF    1          YES                 3.03   0.00
```

Figure 7-16 SDSF DA panel display showing temporal affinities

7.11.2 RMF support

There are no RMF reports affected. RMF obtains the data and places an indicator whether a temporal affinity exists in its SMF type 79, subtype 1 and 2 records. SDSF retrieves the information from there by using an interface. With RMF APAR OW46622, address space state data is enhanced to update SMF records for temporal affinities.

7.11.3 Vary command

The output for the **vary** command has been changed to indicate whether temporal affinities exist. WLM issues message IWM031I every three minutes until the vary command has been completed. If temporal affinities exist in the sysplex, the message lists the systems where these server address spaces are. Further investigation can be done by using the SDSF DA display, as shown in Figure 7-16.

7.12 WLM velocity goals

A change was made in WLM to make the CPU using samples more accurate, especially in an LPAR environment, by deriving CPU using samples from CPU service time rather than from direct sampling. Part of this change was to increase CPU delay by the difference between sampled CPU using and CPU time-based using. Because the sampler seems to relatively favor lightweight work, this type of work is being attributed too much CPU delay in z/OS because the CPU using samples changed from OS/390 R10 to z/OS V1R1, causing execution velocity in z/OS to be smaller than in OS/390. Because of the increase in CPU delay, the achieved velocity is lower.

Therefore, it is possible to see smaller achieved execution velocities after migrating from OS/390 to z/OS, even though the system load and the behavior of the workload remain the same. The change was necessary in order to manage multiple partitions based on their workloads correctly in a LPAR cluster for IRD. To resolve this, install APAR OW55665.

APAR OW55665

APAR OW55665 does the following:

- If sampling provides more CPU using samples than the service time conversion, the additional using samples are no longer converted to CPU delays.
- In addition, a small fraction of the service time can be lost due to rounding problems. This fraction is accumulated on the service class period and periodically converted to CPU using samples.

WLMZOS tool

A tool that allows you to evaluate the change in execution velocities for your environment when you migrate from OS/390 to z/OS is available at the following WLM site:

`tap://www.ibm.com/servers/server/zseries/zos/wlm/tools/velocity.html`

The tool consists of two parts:

- An SMF reduction program, which processes RMF SMF type 72 subtype 3 records and creates an output table containing information about execution velocities and CPU using for each service class period and the RMF data collection interval.
- An Excel spreadsheet, which uses the table created by the SMF reduction program and creates charts to make the execution velocity changes visible. In addition, it performs a rudimentary analysis of the service class periods running on your system.

7.13 Enterprise workload management: eWLM

Over time, some of the well-known WLM constructs will be used to aid in enterprise-wide workload management. This project is an ongoing effort to provide self-management capabilities throughout IBM eServer families.

At the time of writing, the IBM thinkresearch Web site features a research paper entitled *The Great Balancing Act*, which discusses how core technology may be used to bring autonomic computing to servers, and how eWLM weighs the many demands of today's servers.

For an early view and discussion of the eWLM prototype, refer to:

`http://www.research.ibm.com/thinkresearch/pages/2002/20020529_ewlm.shtml`

The paper contains a discussion on eWLM as follows:

- Managing complexity
- Allocating for efficiency
- Distributed computing
- Autonomic computing

Archived

Archived

8

UNIX System Services enhancements in z/OS V1R3

In this chapter we discuss the enhancements introduced in z/OS V1R3 UNIX System Services:

- ACL support
- ISHELL enhancements
- Shutting down z/OS UNIX without re-IPLing
- Automount enhancements
- Copytree utility
- Shared HFS enhancements:
  - Mount table monitoring
  - Shared support for confighfs

Archived

© Copyright IBM Corp. 2003

8.1 Access control list (ACL) support for V1R3

Traditionally, the authorization checking for accessing z/OS UNIX files and directories in a file system has been done using the file security packet (FSP), shown in Figure 8-1.

The FSP is stored in the file system as part of the attributes of a file or directory and is created when the file directory is created. If a security authorization is needed for a file or directory, the security packet is passed to the security product for authorization checking. The level of authorization for the file or directory through the FSP allows specification of file permission bits for file owner (user), group owner, and others, but cannot permit or restrict the access to other specific users and groups.

HFS and zFS files are currently protected with POSIX permission bits contained within the FSP in the file system (not in RACF).

Access	Permission for File	Permission for Directory
Read (r)	Permission to read or print the contents.	Permission to read, but not search, the contents.
Write (w)	Permission to change, add to, or delete from the contents.	Permission to change, add, or delete directory entries.
Execute (x)	Permission to run the file. This permission is used for executable files.	Permission to search the directory.

Figure 8-1 File security packet (FSP)

8.1.1 File access authorization checking

Both the ACL and FSP are maintained by the Physical File System (PFS). When a security decision is needed, the file system calls the security product, supplying the ACL, if present, and the FSP. If the security product supports ACLs, it applies its own rules to the file access request.

RACF uses the permission bits, the access ACL, and the following UNIXPRIV class profiles to determine whether the user is authorized to access the file with the requested access level:

- SUPERUSER.FILESYS
- RESTRICTED.FILESYS.ACCESS
- SUPERUSER.FILESYS.ACLOVERRIDE

Once RACF has checked the authorization, it returns control to the file system.

SUPERUSER.FILESYS profile

This profile, in the UNIXPRIV class, was introduced in OS/390 V1R8. The UNIXPRIV class provides the capability to assign specific superuser functions to a user or group when you give a user or group either of the following:

- UID of 0
- BPX.SUPERUSER profile

Either of these gives a user or group access to all UNIX functions and resources. A BPX.SUPERUSER profile allows you to request that you be given such access, but you do not have the access unless you make the request. So, instead of giving a user or group access to all functions a superuser has, the UNIXPRIV class provides profiles that allow access to a specific superuser function.

The SUPERUSER.FILESYS profile in the UNIXPRIV class has three access levels that allow access to z/OS UNIX files, as follows:

READ Allows a user to read any local file, and to read or search any local directory.

UPDATE Allows a user to write to any local file, and includes privileges of READ access.

CONTROL/ALTER Allows a user to write to any local directory, and includes privileges of UPDATE access.

RESTRICTED attribute

You can define a restricted user ID by assigning the RESTRICTED attribute through the ADDUSER or ALTUSER command, as follows:

```
ALTUSER RSTDUSER RESTRICTED
```

User IDs with the RESTRICTED attribute cannot access protected resources they are not specifically authorized to access. Access authorization for restricted user IDs bypasses global access checking. In addition, the UACC of a resource and an ID(*) entry on the access list are not used to enable a restricted user ID to gain access.

> **Note:** The RESTRICTED attribute does not prevent users from gaining access to z/OS UNIX file system resources unless you take certain steps, as shown in “New UNIXPRIV profiles with z/OS V1R3” on page 123.

However, the RESTRICTED attribute has no effect when a user accesses a z/OS UNIX file system resource; the file's “other” permission bits can allow access to users who are not explicitly authorized. To ensure that restricted users do not gain access to z/OS UNIX file system resources through "other" bits, you must use the new UNIXPRIV profile, RESTRICTED FILESYS.ACCESS.

8.1.2 New UNIXPRIV profiles with z/OS V1R3

The algorithm in Figure 8-5 on page 130 includes access checking for two of the three new UNIXPRIV profiles:

RESTRICTED.FILESYS.ACCESS This profile in the UNIXPRIV class controls the access to file system resources for restricted users based on the “other” permission bits.

SUPERUSER.FILESYS.ACLOVERRIDE This profile allows RACF to force the use of the ACL authorizations to override a user's SUPERUSER.FILESYS profile authority.

SUPERUSER.FILESYS.CHANGEPERMS This profile allows users to use the **chmod** command to change the permission bits of any file and to use the **setfacl** command to manage access control lists for any file.

RESTRICTED.FILESYS.ACCESS profile

This profile specifies that RESTRICTED users (see "RESTRICTED attribute" on page 123) cannot gain file access by virtue of the "other" permission bits (as shown at (3) in Figure 8-5 on page 130).

Checking for this new profile RESTRICTED.FILESYS.ACCESS (shown at (BB) in Figure 8-5 on page 130) is done for RESTRICTED users regardless of whether an ACL exists, so this function can be exploited regardless of whether you plan to use ACLs or not. You can define the profile as follows:

```
RDEFINE UNIXPRIV RESTRICTED.FILESYS.ACCESS UACC(NONE)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Note: Using UACC(READ) on the RESTRICTED.FILESYS.ACCESS profile does not work, since by definition a RESTRICTED user cannot be granted access via a UACC. This would be a meaningless thing to do given that you simply would not define RESTRICTED.FILESYS.ACCESS if you want "other" bits to be checked for RESTRICTED users.

For exception cases, permit the RESTRICTED user (or one of its groups) to the RESTRICTED.FILESYS.ACCESS profile (shown at (B) and (1) in Figure 8-5 on page 130) as follows:

```
PERMIT RESTRICTED.FILESYS.ACCESS CLASS(UNIXPRIV) ID(RSTDUSER) ACCESS(READ)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

This does not grant the user access to any files. It just allows the "other" bits (shown at (C) in Figure 8-5 on page 130) to be used in access decisions for this user.

Note: SUPERUSER.FILESYS still applies to RESTRICTED users regardless of the existence of the RESTRICTED.FILESYS.ACCESS profile (shown at (D) and path (2) in Figure 8-5 on page 130).

SUPERUSER.FILESYS.ACLOVERRIDE profile

Any user who is not a superuser with UID(0), or the file owner—and who is denied access through the ACL—can still access a file system resource if the user has sufficient authority to the SUPERUSER.FILESYS resource in the UNIXPRIV class. To prevent this, you can force RACF to use your ACL authorizations to override a user's SUPERUSER.FILESYS authority by defining the following profiles:

```
RDEFINE UNIXPRIV SUPERUSER.FILESYS.ACLOVERRIDE UACC(NONE)
SETROPTS RACLIST(UNIXPRIV) RACLIST
```

Figure 8-5 on page 130 shows the algorithm used by RACF to do authorization checking that now includes checking for profiles in the UNIXPRIV class for SUPER.FILESYS.ACLOVERRIDE (shown at (AA) in the figure).

Note: This describes the relationship between the existing SUPERUSER.FILESYS profile and the new SUPERUSER.FILESYS.ACLOVERRIDE profile. Either profile could get checked for a file; it depends upon the presence of an ACL for the file, and the contents of the ACL for granting access.

For exception cases, permit (shown at (E) in Figure 8-5 on page 130) the user or group to SUPERUSER.FILESYS.ACLOVERRIDE with whatever access level would have been required for SUPERUSER.FILESYS as follows:

```
PERMIT SUPERUSER.FILESYS.ACLOVERRIDE CLASS(UNIXPRIV) ID(ADMIN) ACCESS(READ)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

SUPERUSER.FILESYS authority is still checked (shown at (D) in Figure 8-5 on page 130) when an ACL does not exist for the file (if you follow the path (4) shown at (A) in the figure). This should be done for administrators for whom you want total file access authority. That is, you do not want anyone to deny them access to a given file or directory by defining an ACL entry for them with limited, or no, permission bit access.

Important: The intent of these new profiles is to allow ACLs to behave as much as possible like RACF profile access lists. The new profiles are provided to avoid changing the default behavior, as that could introduce compatibility issues with previous releases.

For more information about the new RESTRICTED.FILESYS.ACCESS, SUPERUSER.FILESYS.ACLOVERRIDE, and SUPERUSER.FILESYS.CHANGEPERMS UNIXPRIV profiles, refer to *z/OS V1R3 Security Server RACF Security Administrator's Guide*, SA22-7683.

SUPERUSER.FILESYS.CHANGEPERMS profile

As an enhancement to superuser granularity, when using the **chmod** command, a RACF service (IRRSCF00) has been updated to check the caller's authorization to the resource SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class if the caller's user ID is not one of the following:

- UID(0)
- The owner of the file
- BPX.SUPERUSER

If the user executing the **chmod** command has at least READ authority to the resource, the user is authorized to change the file mode in the same manner as a user having UID(0).

This profile allows users to use the **chmod** command to change the permission bits of any file and to use the **setfacl** command to manage access control lists for any file.

8.1.3 ACL overview

ACLs have existed on various Unix platforms for many years, but with variations in the interfaces. ACL support in z/OSV1R3 is based on a POSIX standard that was never approved and other Unix implementations.

In the POSIX standard, two different ACLs are referenced as follows:

- Base ACL entries are permission bits (owner, group, other). It refers to the FSP.
- Extended ACL entries are ACL entries for individual users or groups, such as the permission bits that are stored with the file, not in RACF profiles.

See “ACL entries” on page 126 for a description of these entries.

Access control lists (ACLs) are introduced in z/OS V1R3 as a way to provide a greater granularity for access to z/OS UNIX files and directories. z/OS V1R3 provides support for access control lists (ACLs) to control access to files and directories by individual user (UID) and group (GID). ACLs are now created and checked by RACF. ACLs are created, modified, and deleted by using either the **setfacl** shell command or the ISHELL interface

To display ACLs, use the **getfacl** shell command. The HFS and zFS file systems support ACLs.

Attention: Before you can begin using ACLs, your security product must support ACLs.

Migration considerations

For migration, you need to upgrade any nodes that share HFS or zFS to z/OS V1R3 or, at a minimum, apply the compatibility APAR OW49334 to the downlevel systems.

8.1.4 Security product and ACLs

ACLs are used together with the permission bits in the FSP to control the access to z/OS UNIX files and directories by individual users (UIDs) and groups (GIDs). An ACL is mapped by the SAF IRRPFACL macro, as shown in Figure 8-2 on page 127, where the set of user entries is followed by the set of group entries.

The entries are sorted in ascending order by UID and GID to optimize the access checking algorithm. The algorithm consists of a list of entries (with a maximum of 1024) where every entry has information about the type (user or group), identifier (UID or GID), and permissions (read, write, and execute) to apply to a file or directory.

ACL entries

There are two types of ACL entries:

Base ACL entries These entries are the same as permission bits (owner, group, other) that have always existed with z/OS UNIX files and directories. You can change the permissions using the **chmod** or the new **setfacl** commands. They are not physically part of the ACL although you can use the **setfacl** command to change them and the **getfacl** command to display them.

Extended ACL entries These entries are for individual users or groups and, like the permission bits, they are stored with the file, not in RACF profiles. Each ACL type (access, file default, directory default) can contain up to 1024 extended ACL entries. Each extended ACL entry specifies a qualifier to indicate whether the entry pertains to a user or a group; the actual UID or GID itself; and the permissions being granted or denied by this entry. The allowable permissions are read, write, and execute. As with other UNIX commands, the **setfacl** command allows the use of either names or numbers when referring to users and groups.

Header	
- type - length	- number of entries - number of user entries

Entries (1 - 1024)		
Entry Type	Identifier (UID or GID)	Permissions
User (X'01')	46	r - x
....		
....		
....		

Figure 8-2 Access control list table

There is no such thing as an empty ACL. If there is only one entry and it is deleted, the ACL table is automatically deleted.

8.1.5 Creating ACLs

This support for ACLs allows you to control access to files and directories by individual user (UID) and group (GID). z/OS UNIX file security on z/OS uses permission bits to control access to files, in accordance with the POSIX standard. However, the permission bit model does not allow for granting and denying access to specific users and groups, such as is possible using RACF profiles. This function will be provided by the introduction of ACLs in the z/OS UNIX file system. An ACL is a SAF-owned construct which resides within the file system. The RESTRICTED attribute of a user is now applicable to file and directory access, as described in “RESTRICTED.FILESYS.ACCESS profile” on page 124.

To create an ACL for a file, you must have one of the following security access controls:

- Be the file owner
- BPX.SUPERUSER
- Have superuser authority (UID=0)
- Have READ access to the SUPERUSER.FILESYS.CHANGEPERMS profile in the UNIXPRIV class, as described in “File and directory access with ACLs” on page 130.

Note: The RACF UNIXPRIV class was introduced in OS/390 V1R8. It allows you to define profiles in the UNIXPRIV class to grant RACF authorization for certain z/OS UNIX privileges. By defining profiles in the UNIXPRIV class, you can grant specific superuser privileges to users who do not have superuser authority (UID=0). This allows you to minimize the number of assignments of superuser authority at your installation and reduces your security risk.

To activate the use of ACLs in z/OS UNIX file authority checks, the following RACF command needs to be run to activate the new RACF class FSSEC:

```
SETROPTS CLASSACT(FSSEC)
```

You can define ACLs prior to activating the FSSEC class and display ACL information—but if the FSSEC class is not active, only the standard POSIX permission bit checks are done, even if an access ACL exists.

Activating the RACF FSSEC class causes the ACLs to be used during access checking. In order to set or modify ACLs, the same requirements are needed as for changing the permission bits.

ACL types

To reduce administrative overhead, three types of ACLs (extended ACLs) are defined to have the capability to inherit ACLs to newly created files and directories:

Access ACLs This type of ACL is used to provide protection for a file system object (specific for a file or directory).

File default ACLs This type is a model ACL that is inherited by files created within the parent directory. The file default ACL is copied to a newly created file as its access ACL. It is also copied to a newly created subdirectory as its file default ACL.

Directory default ACLs This type is a model ACL that is inherited by subdirectories created within the parent directory. The directory default ACL is copied to a newly created subdirectory as both its access ACL and directory default ACL.

> **Attention:** The phrases “default ACL” and “model ACL” are used interchangeably throughout z/OS UNIX documentation. Other systems that support ACLs have default ACLs that are essentially the same as the directory default ACLs in z/OS UNIX.
>
> According to the X/Open UNIX 95 specification, additional access control mechanisms may only restrict the access permissions that are defined by the file permission bits. They cannot grant additional access permissions. Because z/OS ACLs can grant and restrict access, the use of ACLs is not UNIX 95-compliant.

ACL mapping structure

The ACL entries reside in a physically different data structure than the FSP, as shown in Figure 8-2 on page 127. The FSP data structure has been modified to include flags indicating if ACLs exist, as shown in Figure 8-3.

Figure 8-3 FSP updated to contain ACL flags

8.1.6 Access checking with ACLs

The basic algorithm for access checking for files and directories has changed with the introduction of ACLs.

The new order, with the z/OS V1R3 changes showing in bold, is as follows:

1. Check “owner” permission bits.
2. **Check user ACL entries.**
3. Check **union of** “group” permission bits and **group ACL entries.**
   a. All entries are checked until a single entry grants the requested access.
4. Check “other” permission bits.

Note: ACL entries are used only if the RACF FSSEC class is active.

z/OS UNIX file access checking

Authorization checking for z/OS UNIX files and directories is shown in Figure 8-5 on page 130, and RACF makes the following checks:

- The accessor environment element (ACEE) is a control block that contains a description of the current user's security environment, including user ID, current connect group, user attributes, and group authorities. An ACEE is constructed during user identification and verification.
- The effective UID and effective GID of the process is used in determining access decisions. The only exception is that if file access is being tested, rather than requested, the real UID and GID are used instead of the effective UID and GID. The real and effective IDs are generally the same for a process, but if a set-uid or set-gid program is executed, they can be different.

Figure 8-4 Current user’s security environment for access checking

- If the GID matches the file owner GID, the file's “group” permission bits are checked. If the “group” bits allow the requested access, then access is granted.

  If any of the user's supplemental GIDs match the file owner GID, the file's group permission bits are checked. If the group bits allow the requested access, then access is granted.

  If no group bits access is allowed and the FSSEC class is active, and an ACL exists, and there is an ACL entry for any of the user's supplemental GIDs, then the permission bits of that ACL entry are checked. If at least one matching ACL entry was found for the GID, or any of the supplemental GIDs, then processing continues with the ACLOVERRIDE checking.

  If no group ACL matches, then if the UNIXPRIV class is active, the SUPERUSER.FILESYS access is checked.
- SUPERUSER.FILESYS.ACLOVERRIDE is checked only when a user's access was denied by a matching ACL entry based on the user's UID or one of the user's GIDs. If the user's access was denied by the file's permission bits, SUPERUSER.FILESYS is checked.

Figure 8-5 z/OS UNIX file access checking algorithm

8.1.7 File and directory access with ACLs

With the introduction of ACL support, when new files and directories are created in a file system, ACL inheritance is the process of automatically associating an ACL with a newly created object without requiring administrative action.

Figure 8-6 on page 131 shows the access checking flow from a UNIX program to the security product.The z/OS UNIX kernel calls the file system, and the file system calls the security product. The kernel calls the file system iteratively for each directory component of the path name (one is required in order to locate the next). Next, the base file name is retrieved.

For each directory lookup, the file system calls the security product to make sure the user has search authority. Then the security product is called to insure the user has the requested access to the base file.

This is the basic architecture of every UNIX file system. The file system never has the complete path name; this is the major roadblock to implementing file protection with RACF profiles, and is the reason why ACLs are now introduced.

Figure 8-6 Access checking flow

8.1.8 ACL inheritance

ACL inheritance, as shown in Figure 8-7 on page 132, associates an ACL with the newly created file, myfile, without requiring administrative action. However, it is not always (and in fact, may seldom be) necessary to apply ACLs on every file or directory within a subtree. If you have a requirement to grant access to an entire subtree (for example, a subtree specific to a given application), then access can be established at the top directory.

If a given user or group does not have search access to the top directory, then no files within the subtree will be accessible, regardless of the permission bit settings or ACL contents associated with these files. The user or group will still need permission to the files within the directory subtree where appropriate. If this is already granted by the “group” or “other” bits, then no ACLs are necessary below the top directory.

Note: When defining ACLs, we recommend you place ACLs on directories, rather than on each file in a directory.

Figure 8-7 ACL inheritance

8.1.9 Defining ACLs from OMVS

The new shell commands, **setfacl** and **getfacl**, have been introduced in order to create, modify, and display ACL entries specified by the path, as follows:

setfacl The **setfacl** command sets, modifies, and deletes an ACL definition for a file or directory. setfacl has the following syntax:

```
setfacl [–ahqv] -s entries [path ... ]
setfacl [–ahqv] -S file [path ...]
setfacl [–ahqv] -D type [...][path ... ]
setfacl [–ahqv] -m|M|x|X EntryOrFile [...][path ... ]
```

getfacl The **getfacl** command obtains and displays an ACL entry for a requested file or directory. It has the following syntax:

```
getfacl [–acdfhmos][-e user] file
```

Important: See the *z/OS UNIX System Services Command Reference*, SA22-7802, for a complete explanation of the commands and all the parameters.

In Figure 8-7 on page 132, directory harry has the three types of ACLs defined, as described in “ACL types” on page 128.

Creating an access ACL

These ACLs are used to provide protection for a file system directory or file. When you are setting the access ACL, the ACL entries must consist of three required base ACL entries that correspond to the file permission bits. The ACL entries must also consist of zero or more extended ACL entries, which will allow a greater level of granularity when controlling access. The permissions for base entries must be in absolute form.

setfacl command

With the **setfacl** command, to create an ACL, you use the **-s** option. You must create the entire ACL, which includes the base ACL and extended ACL, as described in “ACL entries” on page 126. The base ACL (permission bits) are indicated by omitting user or group qualifiers.

For directory harry in Figure 8-7 on page 132, the following command creates an access ACL that gives user ID JANE rwx access to directory harry:

```
ROGERS @ SC65:/u>setfacl -s u::rwx,g::---,o::---,u:jane:rwx harry
```

Note: When you are *setting* the access ACL, the ACL entries must consist of the three required base ACL entries that correspond to the file permission bits (**u::rwx**). The ACL entries must also consist of zero or more extended ACL entries (**g::---,o::---,u:jane:rwx**), which will allow a greater level of granularity when controlling access. The permissions for base entries must be in absolute form. See “ACL entries” on page 126 for more information.

Issuing the **ls -al** command, shown in Figure 8-8, shows directory harry having a plus (+) sign following the permission bits, which indicates that an ACL exists.

```
ROGERS @ SC65:/u>ls -al
total 152
dr-xr-xr-x  11 HAIMO    NOGROUP       0 Aug  2 10:45 .
drwxr-xr-x  48 HAIMO    SYS1      24576 Jul 25 14:44 ..
drwx------+  2 HARRY    SYS1       8192 Aug  2 10:44 harry
drwx------   2 JANE     SYS1       8192 Aug  2 10:44 jane
drwxr-xr-x   2 HAIMO    SYS1       8192 Jun 28 12:23 ldapsrv
drwx------   2 HAIMO    SYS1       8192 Aug  1 11:02 rogers
drwxr-xr-x   2 HAIMO    SYS1       8192 Nov 15  2001 syslogd
drwx------   3 HAIMO    SYS1       8192 May 26 11:03 user1
```

Figure 8-8 Command to show an ACL exists

getfacl command

The **getfacl** command displays the comment header, base ACL (access control list) entries, and extended ACL entries, if there are any, for each file that is specified. It also resolves symbolic links. You can specify whether to display access, file default, or directory default. You can also change the default display format. The output can be used as input to setfacl.

-a Displays the access ACL entries. This is the default if -a, -d, or -f is not specified. Figure 8-9 displays the ACL entry just created for directory harry.

```
ROGERS @ SC65:/u>getfacl -a harry
#file:  harry/
#owner: HARRY
#group: SYS1
user::rwx               <=== The owner's permission bit setting
group::---              <=== The group's permission bit setting
other::---              <=== Permission bit setting if neither user nor group
user:JANE:rwx
```

Figure 8-9 Display of an ACL entry for a directory

Defining a directory default ACL

Directory default ACLs are model ACLs that are inherited by subdirectories created within the parent directory. The directory inherits the model ACL as its directory default ACL and as its access ACL, as shown in Figure 8-7 on page 132 when directory programx inherits the directory default ACL.

For directory harry in Figure 8-7 on page 132, the following command creates a directory default ACL that gives user jane rwx access in a directory default ACL for directory harry:

```
ROGERS @ SC65:/u>setfacl -s "u::rwx,g::---,o::---,d:u:jane:rwx" harry
```

To display the newly created ACL, see Figure 8-10.

```
ROGERS @ SC65:/u>getfacl -d harry
#file:  harry/
#owner: HARRY
#group: SYS1
default:user:JANE:rwx
```

Figure 8-10 Display of a directory default ACL

Defining a file default ACL

File default ACLs are model ACLs that are inherited by files created within the parent directory. The file inherits the model ACL as its access ACL. Directories also inherit the file default ACL as their file default ACL, as shown in Figure 8-7 on page 132.

For directory harry in Figure 8-7 on page 132, the following command creates a file default ACL that gives user jane r-- access in a file default ACL for directory harry:

```
ROGERS @ SC65:/u>setfacl -s "u::rwx,g::---,o::---,f:u:jane:r--" harry
```

To display the newly created ACL, see Figure 8-11.

```
ROGERS @ SC65:/u>getfacl -f harry
#file:  harry/
#owner: HARRY
#group: SYS1
fdefault:user:JANE:r--
```

Figure 8-11 Display of a file default ACL

Attention: In the previous examples to define an access ACL, a directory default ACL, and a file default ACL, each example deleted the previously defined ACL. Therefore, if you want to create all three ACLs for the directory, you should issue just *one* command for ACLs.

Define all three ACL types

The following command will define the three ACLs for directory harry:

```
ROGERS @ SC65:/u>setfacl -s "u::rwx,g::---,o::---,u:jane:rwx,d:u:jane:rwx,f:
u:jane:r--" harry
```

To display the ACLs just created (and shown in Figure 8-14 on page 135), issue the following command:

```
ROGERS @ SC65:/u>getfacl -adf harry
#file:  harry/
#owner: HARRY
#group: SYS1
user::rwx
group::---
other::---
user:JANE:rwx
fdefault:user:JANE:r--
default:user:JANE:rwx
```

Figure 8-12 Display all the ACL types

Note: Any combination of ACL types can be requested, but if you just issue the command as **`getfacl harry`**, only the access ACL is displayed by default, as shown in Figure 8-13.

```
ROGERS @ SC65:/u>getfacl harry
#file:  harry/
#owner: HARRY
#group: SYS1
user::rwx
group::---
other::---
user:JANE:rwx
```

Figure 8-13 Displays access ACL requested with the default

Delete all ACLs for a directory

If you want to delete the three ACLs for directory harry, issue the following command:

```
setfacl -x user:jane,d:user:jane,f:user:jane harry
```

Figure 8-14 ACLs created for directory harry

Current ACL inheritance summary

At this point, as shown in Figure 8-14, user ID JANE now has access to directory harry with three ACLs giving the access. Figure 8-12 on page 135 displays the three ACLs that give user ID JANE access. These three ACLs can also be displayed by using the ISHELL, as shown in “Using ACLs from the ISHELL” on page 136.

8.1.10 Using ACLs from the ISHELL

If you prefer to use the ISHELL rather than the OMVS command line, you can use the ISHELL to display, add, delete, and modify ACLs.

For the example shown in Figure 8-14 on page 135, once you have entered the ISHELL, you should enter `/u` on the command line, as shown in Figure 8-15.

Display the defined ACLs

The next four figures allow you to see the access ACL, directory default ACL, and file default ACL that were just previously defined.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
                      UNIX System Services ISPF Shell
Command ===>

Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                                More:      +
   /u

EUID=0
```

Figure 8-15 ISHELL panel

The next panel displayed (Figure 8-16) shows the current directory list. For directory harry, the plus sign (+) indicates that ACLs exist.

```
  File  Directory  Special_file  Commands  Help
                              Directory List
Command ===>

Select one or more files with / or action codes.  If / is used also select an
action from the action bar otherwise your default action will be used.  Select
with S to use your default action.  Cursor select can also be used for quick
navigation.  See help for details.
EUID=0   /u/
  Type  Perm  Changed-EST5EDT   Owner      ------Size  Filename    Row 1 of 8
_ Dir    555  2002-08-02 10:45  HAIMO              0  .
_ Dir    755  2002-07-25 14:44  HAIMO          24576  ..
a Dir   +700  2002-08-02 10:44  HARRY           8192  harry
_ Dir    700  2002-08-02 10:44  JANE            8192  jane
_ Dir    755  2002-06-28 12:23  HAIMO           8192  ldapsrv
_ Dir    700  2002-08-01 11:02  HAIMO           8192  rogers
_ Dir    755  2001-11-15 14:35  HAIMO           8192  syslogd
_ Dir    700  2002-05-26 11:03  HAIMO           8192  user1
```

Figure 8-16 ISHELL directory list panel

By placing an `a` action code for directory harry, the **File Attribute** panel is displayed, as shown in Figure 8-17 on page 137. In this example there is one access ACL, indicated by the Access Control List field.

```
File  Directory  Special_file  Commands  Help
  Edit  Help
              Display File Attributes
Pathname : /u/harry
                                          More:      +
File type . . . . . . : Directory
Permissions . . . . . : 700
Access control list . : 1
File size . . . . . . : 8192
File owner  . . . . . : HARRY(10103)
Group owner . . . . . : SYS1(2)
Last modified . . . . : 2002-08-02 10:44:16
Last changed  . . . . : 2002-08-02 10:47:59
Last accessed . . . . : 2002-08-02 10:49:23
Created . . . . . . . : 2002-08-02 10:44:16
Link count  . . . . . : 2
 F1=Help          F3=Exit          F4=Name
 F7=Backward      F8=Forward      F12=Cancel
```

Figure 8-17 Display File Attributes panel

Note: The field Access Control List shows the information related with access ACL; if you scroll forward, you will see if a Directory Default ACL or File Default ACL is defined, as shown in Figure 8-18. These two fields (Directory Default and File Default ACL) only apply to directory files.

```
File  Directory  Special_file  Commands  Help
  Edit  Help
              Display File Attributes
Pathname : /u/harry
                                          More:      -
Major device  . . . . : 0
Minor device  . . . . : 0
File format . . . . . : NA
Shared AS . . . . . . : -
APF authorized  . . . : -
Program controlled  . : -
Shared library  . . . : -
Char Set ID/Text flag : 0000 OFF
Directory default ACL : 1
File default ACL  . . : 1
Seclabel  . . . . . . : _______________
 F1=Help          F3=Exit          F4=Name
 F7=Backward      F8=Forward      F12=Cancel
```

Figure 8-18 Display of directory default ACL and file default ACL for /u/harry

Display the ACLs

By placing the cursor under the **Edit** pull-down in Figure 8-17 and pressing Enter, you can chose the option for displaying ACL information, as shown in Figure 8-19 on page 138.

Options 8, 9, and 10 can be selected to display the access ACL, the directory default ACL, and the file default ACL. If any of the ACLs do not exist, an asterisk will appear in place of the 8, 9, or 10.

```
File  Directory  Special_file  Commands  Help
  Edit  Help
  8_ 1. Mode fields...
     2. Owning user...
     3. Owning group...
     4. User auditing...                More:     +
     5. Auditor auditing...
     6. File format...
     7. Extended attributes...
     8. Access contol list...
     9. Directory default ACL...   )
    10. File default ACL...
                                        10:44:16
Last changed  . . . . : 2002-08-02 10:47:59
Last accessed . . . . : 2002-08-02 10:49:23
Created . . . . . . . : 2002-08-02 10:44:16
Link count  . . . . . : 2
 F1=Help        F3=Exit        F4=Name
 F7=Backward    F8=Forward     F12=Cancel
s used also select an
 will be used.  Select
so be used for quick
ilename    Row 1 of 8
.
arry
ane
dapsrv
ogers
yslogd
ser1
```

Figure 8-19 Edit access control list panel to access ACL information

When you press Enter after specifying option 8, 9, or 10, you now have access to modify, add or the delete ACL entries, as shown in Figure 8-20. From this panel you can display, add, delete, copy, replace, or modify any of the permission settings (rwx) for the ACL entry.

This panel displays the current access ACL for directory harry that allows user ID JANE with a UID=10102 to have rwx access to the directory.

```
File  Directory  Special_file  Commands  Help
  Edit  Help
            Display File Attributes
Pathname : /u/harry
s used also select an
 will be used.  Select
            Access Control List: Access                Row 1 to 1 of 1
Command ===> ____________________________________ Scroll ===> PAGE

Type over read, write or execute permissions to make a change.
Clear the value to reset it, anything else will set it.
To delete, place a D in the S column for an entry or use command D * for
all entries. Use commands SORT ID or SORT NAME to reorder the table.

Option:    _  1. Add group  2. Add user   3. Copy       4. Replace

S          ID  Name      Read  Write  Execute  Type
_       10102  JANE       R      W       X     User
*************************** Bottom of data ******************************
```

Figure 8-20 Access Control List panel to change ACL definitions for Option 8

8.1.11 Create an ACL using the ISHELL

Select the permissions, using a forward slash (/) or other non-blank character, and enter either the numeric ID or the name. A name must have an associated UID or GID, and a UID or GID must have an associated name.

8.1.12 Example of ACL inheritance

The directory structure is changed, as shown in Figure 8-21 on page 139, by issuing the following command:

```
mkdir /u/harry/programx
```

The new directory, programx, inherits an access ACL from the directory default ACL of directory harry, and inherits the directory default ACL and file default ACL from directory harry, as shown in Figure 8-21.

Figure 8-21 ACL inheritance from directory harry ACLs

Using the getfacl command (shown in Figure 8-22) and the ISHELL (shown in Figure 8-23 and Figure 8-24 on page 140), you can see the inheritance from directory harry to directory programx.

```
HARRY @ SC65:/u/harry>getfacl -adf programx
#file:  programx/
#owner: HARRY
#group: SYS1
user::rwx
group::r-x
other::r-x
user:JANE:rwx
fdefault:user:JANE:r--
default:user:JANE:rwx
```

Figure 8-22 Display of ACLs for directory programx

```
  File  Directory  Special_file  Commands  Help
 ------------------------------------------------------------------------
                                Directory List
 Command ===> _____________________________________________________________

 Select one or more files with / or action codes.  If / is used also select an
 action from the action bar otherwise your default action will be used.  Select
 with S to use your default action.  Cursor select can also be used for quick
 navigation.  See help for details.
 EUID=0   /u/harry/
   Type  Perm  Changed-EST5EDT   Owner      ------Size  Filename    Row 1 of 3
 _ Dir   +700  2002-08-02 15:40  HARRY            8192  .
 _ Dir   +700  2002-08-02 15:40  HARRY            8192  ..
 _ Dir   +755  2002-08-02 15:40  HARRY            8192  programx
```

Figure 8-23 ISHELL display of directory programx showing ACLs exist

```
  File  Directory  Special_file  Commands  Help
_
  Edit  Help
C
                Display File Attributes
S
a  Pathname : /u/harry/programx                    s used also select an
                                                     will be used.  Select
                  Access Control List: Access                  Row 1 to 1 of 1
Command ===>                                                 Scroll ===> PAGE

Type over read, write or execute permissions to make a change.
Clear the value to reset it, anything else will set it.
To delete, place a D in the S column for an entry or use command D * for
all entries. Use commands SORT ID or SORT NAME to reorder the table.

Option:     _  1. Add group  2. Add user   3. Copy      4. Replace

S            ID  Name       Read  Write  Execute  Type
_         10102  JANE        R      W       X     User
****************************** Bottom of data ******************************
```

Figure 8-24 Display of an access ACL for directory programx

The directory structure is changed again, as shown in Figure 8-25, by issuing the following command that creates a file named myfile:

```
oedit /u/harry/programx/myfile
```

The new file, myfile, inherits only the file default ACL from directory programx (Figure 8-25).

Figure 8-25 ACL inheritance of an access ACL to file myfile

By using the getfacl command to display the ACLs for file myfile, Figure 8-26 on page 141 shows the file default ACL inherited from the directory programx (shown in Figure 8-22 on page 139).

```
HARRY @ SC65:/u/harry/programx>getfacl myfile
#file:  myfile
#owner: HARRY
#group: SYS1
user::rwx
group::---
other::---
user:JANE:r--
```

Figure 8-26 Display of ACL for file myfile

Using the ISHELL, Figure 8-27 and Figure 8-28 show the access ACL that was inherited by the file myfile from directory programx.

```
  File  Directory  Special_file  Commands  Help
                                 Directory List

Select one or more files with / or action codes.  If / is used also select an
action from the action bar otherwise your default action will be used.  Select
with S to use your default action.  Cursor select can also be used for quick
navigation.  See help for details.
EUID=10103   /u/harry/programx/
  Type  Perm  Changed-EST5EDT   Owner      ------Size  Filename    Row 1 of 3
_ Dir   +755  2002-08-02 16:30  HARRY            8192  .
_ Dir   +700  2002-08-02 15:40  HARRY            8192  ..
a File  +700  2002-08-02 16:31  HARRY              47  myfile
```

Figure 8-27 ISHELL display of file myfile showing an ACL exists

```
  File  Directory  Special_file  Commands  Help
_   Edit  Help
S                Display File Attributes                s used also select an
a                                                        will be used.  Select
w  Pathname : /u/harry/programx/myfile                  so be used for quick
              Access Control List: Access                Row 1 to 1 of 1

Type over read, write or execute permissions to make a change.
Clear the value to reset it, anything else will set it.
To delete, place a D in the S column for an entry or use command D * for
all entries. Use commands SORT ID or SORT NAME to reorder the table.

Option:     _   1. Add group  2. Add user   3. Copy       4. Replace

S           ID  Name      Read  Write  Execute  Type
_        10102  JANE       R      _       _     User
***************************** Bottom of data ******************************
```

Figure 8-28 Access control list window

User ID JANE now has read access to the file myfile, whose owner is user ID harry.

8.1.13 Other setfacl command options

The following options can be specified:

- In the following example, we have set for /u/user1/test directory, rwx permissions bits for user, r-x for group, r-x for others and we have created an access ACL entry for group SYS1 with rwx permits.

```
setfacl -s u::rwx,g::r-x,o::r-x,g:SYS1:rwx /u/user1/test
```

Note: It is possible to modify the permissions bits with the **setfacl -s** command instead of using the **chmod** command

- The -m/-M options allow you to modify an ACL extended definition. The base ACL is not needed with the modify option.

  In the following example we have created for directory /u/user1/test, a file default ACL with access rwx for user CBSYMSR1 and a directory default ACL with rwx access for group SYS1:

  ```
  setfacl -m f:user:CBSYMSR1:rwx,d:g:SYS1:rwx  /u/user1/test
  ```

- The -D option allows you to delete extended ACLs for the type specified, either an access ACL(**a**), a file default ACL(**f**), a directory default ACL(**d**), or all the extended ACL(**e**).

  The following example removes the file default ACL(**f**) for /u/user1/test directory:

  ```
  setfacl -D f /u/user1/test
  ```

- The -x/X options allow you to delete the ACL definition specified.

  The following example removes the ACL that has group SYS1 and has rwx access from the /u/user1/test directory:

  ```
  setfacl -x g:SYS1:rwx /u/user1/test
  ```

- To delete all of the extended ACL entries for all files and directories in the current working directory:

  ```
  setfacl -D e *
  ```

Command errors

Default ACLs can be only specified for directories; otherwise, the **setfacl** command fails with the following message:

```
FSUMF229 setfacl: warning: /u/user1/filetest is not a directory so Directory Default ACL
cannot be changed.
```

8.1.14 Modified commands with ACL support

The following commands have been modified in order to support ACL entries:

getconf This changed command returns configuration values associated with the file at the specified pathname, as follows:

_PC_ACL - Indicates whether an access control mechanism is supported by the file system owning the file specified by “pathname”. A value of 1 indicates that it is supported, as shown in Figure 8-29, and a value of 0 indicates it is not supported.

_PC_ACL_ENTRIES_MAX - Maximum number of entries in an ACL for a file or directory, as shown in Figure 8-29, which indicates a value of 1024.

```
@ SC63:/>getconf _PC_ACL /u/user1/test
1
@ SC63:/>getconf _PC_ACL_ENTRIES_MAX /u/user1/test
1024
```

Figure 8-29 Examples of the getconf command

ls command

- The **ls** command will indicate the existence of ACLs by adding a plus sign (+) character after the permission bits, as follows:

```
SC63:/>ls -l /u/user1
drwxr-xr-x+  2 HAIMO    SYS1        8192 May 26 09:42 test find
```

find command

The **find** command has new options that have been added for supporting ACL entries.

- The **find** command finds all files or directories with an ACL of a given type. In the next example, the command displays all the files/directories under /u/user1 directory which have any type of ACL (access, default file or default directory):

```
AYVIVAR @ SC63:/>find /u/user1 -acl a -o -acl d -o -acl f
/u/user1/test
```

- Find files with ACL entries for a specific user/group. In the next example, **find** displays all the files/directories under /u/user1 directory that have ACL entries for SYS1 group:

```
@ SC63:/>find /u/user1 -acl_group SYS1
/u/user1/test
```

- Find files with more than the specified amount of ACL entries:

```
@ SC63:/>find /u/user1 -acl_count +1
/u/user1/test
```

- In the following example, the find command is useful in command substitution, as it can produce file lists that are used as input to the **setfacl** command:

```
setfacl -m g:OMVSGRP:rwx $(find /u/user1 -acl_group SYS1)
```

cp command

The **cp -p** command preserves ACLs from source to target, if possible. The ACLs are not preserved if a file system does not support ACLs, or if you are copying files to MVS.

mv command

The **mv** command preserves an ACL from source to target.

pax command

When using the **pax** command, ACL data is automatically stored in USTAR formatted archives using special headers. The following options are not required:

- Extracted files will restore ACLs when **-p A** or **-p e** is specified.
- Copy preserves ACL when **-p A** or **-p e** is specified.
- Verbose output adds a plus sign (+) character after the permission bits when an extended ACL exists.

```
@ SC65:/u/user1>pax -vf test.pax
-rwx------   1 STC      SYS1      620000 May  3 15:28 /u/user1/test/file1
-rwx------+  1 STC      SYS1      660000 May  3 15:29 /u/user1/test/file2
```

tar command

The **tar -U** command with (USTAR format) will preserve ACLs in archives as follows:

- Extracted files will restore ACLs when `-A` is specified.
- For verbose output (`tar -v`), a + character is added to the end of the file permission bits for all files with extended ACLs (as for the pax command).

df command

The **df -v** command indicates whether the file system and security product supports ACLS, as shown in Figure 8-30 on page 144.

```
@ SC63:/>df -v /u/user1/test
Mounted on     Filesystem                Avail/Total    Files      Status
/u/user1       (OMVS.USER1.HFS)          14208/14400    4294967293 Available
HFS, Read/Write, Device:241, ACLS=Y
File System Owner : SC63        Automove=Y      Client=N
Filetag : T=off   codeset=0
```

Figure 8-30 Example of df -v command

Notes:

- ACLS=Y does *not* mean that the FSSEC class profile is active. It means that the file system will store ACLs and pass them to the security product.
- Using ACLs must be supported by the file system that the file or directory belongs to. It is supported in z/OS V1.3 by zFS and HFS. ACLs are not supported currently for a temporary file system (TFS) in z/OS V1R3.

8.1.15 USS Logical File System ACLs support

Several callable services have been modified in order to support ACLs entries:

- BPX1IOC(w_ioctl)/BPX1PIO(w_pioctl)

  These callable services (called, for example, by **setfacl** and **getfacl** commands) perform a device-specific command.

  Call BPX1IOC,(*file_descriptor,command,argument_length,argument,return_value,return_code,reason_code)*

  Support has been added in order to send and receive directory or file ACL structures to/from Physical File Systems (PFS) by:

  - Expand interface to allow larger *argument_length* from 1024 to 2,147,483,647
  - Adding new command codes for processing ACLs
    - SetfACL sets ACL for file or directory.
    - GetfACL gets ACL for file or directory.
- BPX1STA(stat/fstat) returns flags indicating presence and type of any ACLs on the specified file or directory. It is used by the shell **ls** command.

  Call BPX1STA,(*pathname_length,pathname,Status_Area_Length,Status_Area,return_value,return_code,reason_code)*

  BPXYSTAT mapping macro used adds information indicating id access ACL exists, default directory model exists or default file model exists
- BPX1PCF(pathconf/fpathconf) returns configurable variables associated with the file at the specified pathname. Is used by shell *getconf* command.

  Call BPX1PCF,(*pathname_length,pathname,name,return_value,return_code,reason_code)*

  New parm values have been added:

  _ACL indicates whether an access control mechanism is supported by the file system owning the file specified by “pathname”. Values can be TRUE or FALSE.

_ACL_ENTRIES_MAX specifies the maximum number of entries in an ACL for file or directory.

BPXYPCF mapping macro changes to support.

- BPX1GMN(w_getmntent) returns information on a mounted file system. It is used by the **df** shell command.

  `Call BPX1GMN,(Buffer_Length,Buffer,return_value,return_code,reason_code)`

  BPXYMNTE mapping macro changes to support.

- Other mapping macro changes:
  - BPXYVFSI - VFS Callable Service Interface includes ACL fields on the ATTR structure.

For more information about the changes made on the Callable Services, refer to *z/OS UNIX System Services Programming Assembler Callable Services Reference*, SA22-7803.

8.1.16 z/OS UNIX REXX support for ACLs

New services to get, create, update, replace or delete an ACL for a file or directory have been added:

- **aclinit** *variable.* Obtain resources necessary to process ACLs and associated those resources with *variable*. *Variable* is the name of a REXX variable that contains a token to access an ACL.
- **aclfree** *variable.* Releases resources associated with the ACL represented by variable that were obtained by the **aclinit** service.
- **acl get** *variable pathname acltype*. Read an ACL of specified *acltype* associated by the file identified by *pathname.* The ACL is associated with the specified *variable*. *Pathname* is the pathname of the file or directory the ACL is associated with. *Acltype* indicates the type of ACL (access, default file or default directory).
- **acldelete** *pathname acltype.* Deletes an ACL of specified *acltype* associated by the file identified by *pathname*.
- **acl set** *variable pathname acltype*. Replace the ACL associated by the file identified by *pathname* with the ACL represented by *variable*.
- **aclgetentry** *variable stem[index]*. Reads the ACL entry from the ACL represented by *variable.* The entry is identified by *index* if specified, otherwise the entry is identified by the type and ID specified by *stem*. If index is specified, stem is purely an output variable. Otherwise, stem is used for both input and output. The ACL entry is placed in stem.

  stem is the name of a stem variable that contains an ACL entry. STEM.0 contains a count of the number of variables set in the stem. The following variables may be used to access the stem variables:
  - **acl_entry_user** indicates user ACL.
  - **acl_entry_group** indicates group ACL.
  - **acl_id is** the uid or uid of the entry.
  - **acl read** indicates read access.
  - **acl_write** indicates write access.
  - **acl_execute** indicates execute/search access.
  - **acl_delete** indicates the ACL entry as deleted.

  index specifies the relative ACL entry to process. Indexing begins at 1.

- **aclupdateentry** *variable stem[index]*. Updates (or creates, if it does not already exist) an ACL entry from the ACL represented by *variable*. The entry is identified by *index* if specified; otherwise, the entry is identified by the type and ID specified by *stem.* If *index* is specified, *stem* is purely an output variable. Otherwise, stem is used for both input and output.
- **acldeleteentry** *variable stem*. Deletes an ACL entry from the ACL represented by *variable*. The entry is identified by the entry type and ID specified by *stem.*

The example in Figure 8-31 shows a REXX program used for displaying ACL for a requested input file.

Example: Display access ACL for input file

```
/* REXX */
parse arg path
call syscalls 'ON'
address syscall
'aclinit acl'
'aclget acl (path)' acl_type_access
do i=1 by 1
  'aclgetentry acl acl.' i
  if rc<0 | retval=-1 then leave
  parse value '- - -' with pr pw px
  if acl.acl_read=1    then pr='R'
  if acl.acl_write=1   then pw='W'
  if acl.acl_execute=1 then px='X'
  aclid=acl.acl_id
  if acl.acl_entry_type=acl_entry_user then type='UID='
   else
    if acl.acl_entry_type=acl_entry_group then type='GID='
     else
      type='???='
  say pr || pw || px type || aclid
end
'aclfree acl'
```

```
Output
RWX UID=11
RWX UID=12
RWX UID=13
R-- GID=500
```

Figure 8-31 Example REXX program

Other interface changes

New variables for ACL support have been added to the **STAT**, **FSTAT**, **LSTAT** syscall commands:

- ST_ACCESSACL- 1 if access ACL exits
- ST_DMODELACL- 1 if directory model ACL exists
- ST_FMODELACL- 1 if file model ACL exists.

New variables for ACL support have been added to the **PATHCONF** syscall command

- PC_ACL to test if ACLs are supported for the resource
- PC_ACL_MAX to query max number of allowed entries in an ACL.

For more information about REXX support for ACL, refer to *z/OS Using REXX and z/OS UNIX System Services*, SA22-7806.

8.1.17 LE Callable Services support for ACLs

LE Callable Services has been modified in order to support ACL entries, as follows:

- ACL Storage Management Functions

- Initialize ACL working Storage
  ```
  lacl_t acl_init(init count);
  ```
- Release memory allocated to an ACL Data Object
  ```
  int acl_free(acl_t obj_pp);
  ```

- Functions that manipulate complete entries in an ACL
  - Get an ACL entry
    ```
    int acl_get_entry(lacl_t acl_d,acl_entry_t*entry_p);
    ```
  - Return to Beginning of ACL Working Storage
    ```
    int acl_first_entry(lacl_t acl_d);
    ```
  - Validate an ACL
    ```
    int acl_valid(lacl_t acl_d,acl_entry_t*entry_p)
    ```
  - Add a new extended ACL entry to the ACL
    ```
    int acl_create_entry(lacl_t*acl_p,acl_entry_t entry_p,int version);
    ```
  - Delete the specified extended ACL entry from the ACL
    ```
    int acl_delete_entry(lacl_t acl_d,acl_entry_t entry_d):
    ```
  - Update the extended ACL entry
    ```
    int acl_update_entry(lacl_t acl_d,acl_entry_t entry_s,acl_entry_t entry_d,int
    version);
    ```
- Functions that manipulate the whole ACL object
  - Delete an ACL by File Descriptor
    ```
    int acl_delete_fd(int fd,acl_type_t type_d):
    ```
  - Delete an ACL by Filename
    ```
    int acl_delete_file(const char *path_p,acl_type_t type_d);
    ```
  - Get an ACL by File Descriptor
    ```
    int acl_get_fd(int fd,acl_type_t type_d,lacl_t acl_d,int *num);
    ```
  - Get ACL by filename
    ```
    int acl_get_file(const char *path_p,acl_type_t type_d,lacl_t acl_d,int *num);
    ```
  - Set an ACL by file descriptor
    ```
    int acl_set_fd(int fd,acl_type_t type_d,lacl_t acl_d,short OpType,acl_entry_t
    *entry_p);
    ```
  - Set an ACL by filename
    ```
    int acl_set_file(const char *path_p,acl_type_t type_d,lacl_t acl_d,short
    OpType,acl_entry_t*entry_p);
    ```
- Functions that convert between format of ACL
  - Convert an ACL to Text
    ```
    char * acl_to_text(const lacl_t acl_d,ssize_t*len_p,acl_type_t type_d,char
    delim);
    ```
  - Create an ACL from text
    ```
    int acl_form_text(const char *buf_p,short OpType,acl_all_t ptr,char **ret);
    ```
  - Sort the extended ACL entries (USER, GROUP low to high)
    ```
    int acl_sort(lacl_t acl_d);
    ```

Figure 8-32 on page 148 shows examples of using the functions.

Archived

To get an ACL from a file and set the same ACL on another file

acl_init() - to get the ACL buffer
acl_get_fd() or acl_get_file() - to get the ACL from a file
acl_set_fd() or acl_set_file() - to set the ACL on a file
acl_free() - to release storage

To get an ACL data from text and set the ACL on a file

acl_from_text() - creates ACL structure
acl_valid() - to check that entries are properly formed
acl_set_file() or acl_set_fd() - to set the ACL on a file
acl_free() - to release storage

To add, delete and update individual ACL entries

Use a combination of :
acl_get_entry() - to get a pointer to a specific acl entry
acl_delete_entry() - to delete an acl entry
acl_create_entry() - to add a new acl entry
acl_update_entry() - to change the content of an acl entry

Figure 8-32 Examples of using ACL functions

8.2 ISHELL enhancements

New features have been added to improve the ISHELL functionality on z/OS V1R3. The ISHELL is a user interface that allows users to work with menus rather than with sometimes cryptic commands. In z/OS V1R3, many new features that have been requested over the years to improve ISHELL functionality and panel navigation are implemented. These new features are a significant upgrade to the ISHELL and many of them are part of the Directory List options which lists files in a particular directory. Other enhancements include sorting and highlighting support, as well as easy ways to access information.

Many of the new features are part of the directory list, which lists files in a particular directory. The ISHELL enhancements include:

- Sorting
- Highlighting support
- Easy ways to access information, such as the name of a file whose name appears truncated

8.2.1 Directory list enhancements

Before going to the Directory List panel, you can choose which fields on the Directory List that are to be displayed. From the ISHELL panel **-> Options -> Option1 - Directory list...**, then choose which fields you want to display by typing a backslash (/) next to the option, as shown in the Directory Options List in Figure 8-33 on page 149.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
_
                    Directory List Options
C
  Select options and fields to be displayed with /
E
  /  File type    ( 4 columns)
  /  Permissions  ( 4 columns)
  /  Change time  (16 columns)
  /  Owner        ( 9 columns)
  /  File size    (10 columns)
R
  _  View/change sort options...                          More:     +
  _  View/change file name highlighting...
  /  Verbose directory list panel
E
   F1=Help      F3=Exit      F6=Keyshelp F12=Cancel
```

Figure 8-33 Directory List Options

Directory List panel enhancements

The Directory List panel is available through the ISHELL panel-> **Directory** -> **Option 1, List Directory,** or by specifying a file pathname (see Figure 8-34).

```
  File  Directory  Special_file  Commands  Help
                                Directory List
Command ===> _

Select one or more files with / or action codes.  If / is used also select an
action from the action bar otherwise your default action will be used.  Select
with S to use your default action.  Cursor select can also be used for quick
navigation.  See help for details.
EUID=0   /u/harry/programx/
  Type  Perm  Changed-EST5EDT   Owner      ------Size  Filename    Row 1 of 5
_ Dir   +755  2002-08-05 16:12  HARRY            8192  .
_ Dir   +700  2002-08-02 15:40  HARRY            8192  ..
_ File  +700  2002-08-02 16:31  HARRY              47  myfile
_ File  +700  2002-08-05 16:12  HARRY              87  myfilename.newfilename.p
_ File  +666  2002-08-05 12:54  HARRY           88390  o ishell
```

Figure 8-34 Directory List panel

The following changes are made to the Directory List panel shown in Figure 8-34:

- The effective user ID(EUID) is displayed on the panel, so you can know the authority you have at any particular moment. In the example shown in Figure 8-34, the EUID=10103 indicates that the current UID is accessing the ISHELL.
- Times are displayed in local time instead of Greenwich time, as in previous releases.
- The Action Bar can be removed by typing `noab` in the command line, as shown in Figure 8-35 on page 150. If you want to see the Action Bar again, type `ab` in the command line.

```
                                Directory List

Select one or more files with / or action codes.
EUID=10103   /u/harry/programx/
  Type  Perm  Changed-EST5EDT   Owner      ------Size  Filename    Row 1 of 4
_ File  +666  2002-08-05 12:54  HARRY           88390  o ishell
_ File  +700  2002-08-02 16:31  HARRY              47  myfile
_ Dir   +700  2002-08-02 15:40  HARRY            8192  ..
_ Dir   +755  2002-08-05 12:54  HARRY            8192  .
```

Figure 8-35 Changing the Directory List panel to a new format

8.2.2 Using the cursor on the directory list panel

Most areas of the Directory List panel are cursor-sensitive. In a “sensitive” area, you place the cursor under a value and press Enter, a new window appears that allows you to change the value of the selected field. The following actions can be done through the sensitive areas in Directory List pane.

Sorting by column header

By placing the cursor under any of the column headers on the Directory List panel shown in Figure 8-34 on page 149 and pressing Enter, the directory list will be sorted on that column. For example, placing the cursor on the Perm column header and pressing Enter sorts the permission bit settings, as shown in Figure 8-36.

```
  File  Directory  Special_file  Commands  Help
 ------------------------------------------------------------------------
                                Directory List

Select one or more files with / or action codes.  If / is used also select an
action from the action bar otherwise your default action will be used.  Select
with S to use your default action.  Cursor select can also be used for quick
navigation.  See help for details.
EUID=10103   /u/harry/programx/
  Type  Perm  Changed-EST5EDT   Owner      ------Size  Filename    Row 1 of 4
_ File  +666  2002-08-05 12:54  HARRY           88390  o ishell
_ File  +700  2002-08-02 16:31  HARRY              47  myfile
_ Dir   +700  2002-08-02 15:40  HARRY            8192  ..
_ Dir   +755  2002-08-05 12:54  HARRY            8192  .
```

Figure 8-36 Sorting by column header Perm

Sort fields in Directory List panel

Entries in the directory list panel can be sorted by any field (even if the field is not displayed) with a primary and secondary sort field. To set the sort options, you can access the Sorting Options panel shown in Figure 8-37 on page 151 by using one of the following options:

1. Type: `sort` on the command line from the Directory List panel.
2. From the Directory List panel, select **Commands -> Option 2 - Sort**.
3. From the ISHELL panel, select **Options -> Option1 - Directory List**. Then type: / on the column **View/change sort options...**.

```
  File  Directory  Special_file  Commands  Help
_ ┌──────────────────────────────────────────────────────────────────────┐ ──────────
  │                        Select sort fields                            │
S │ Select a primary sort field and optionally select a secondary        │ lect an
a │ sort field.  Note: performance for large directories may be          │   Select
w │ impacted if any field other than a file name is chosen.              │  quick
n │                                                                      │
E │ Primary sort:                  Secondary sort:                       │  1 of 5
  │ 5   1.  Default                1   1.  None                          │
_ │     2.  File name mixed case       2.  File name mixed case          │
_ │     3.  File name                  3.  File name                     │ ilename.p
_ │     4.  File type                  4.  File type                     │
_ │     5.  Permissions                5.  Permissions                   │
_ │     6.  Size                       6.  Size                          │
  │     7.  Owner                      7.  Owner                         │
  │     8.  Change time                8.  Change time                   │
  │                                                                      │
  │  F1=Help         F3=Exit         F6=Keyshelp     F12=Cancel          │
  └──────────────────────────────────────────────────────────────────────┘
```

Figure 8-37 Sort panel sort fields

Note: In Figure 8-37, selecting Option 5, Permissions, does exactly the same sort as shown in “Sorting by column header” on page 150.

Accessing files with cursor on Type header

By placing the cursor on the Type field for any file and pressing Enter, the default action will be taken on that file from the ISHELL panel; select **Options -> Option 2 - Default Actions**, as shown in Figure 8-38.

Notice that for a Regular file, the default action is to edit the file.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
_ ┌──────────────────────────┐ ──────────────────────────────────────────────
  │   Enter Action Defaults  │ ervices ISPF Shell
E │ Directory  . . . . . L   │ se:
  │ Regular file . . . . E   │
  │ Character special    A   │
  │ FIFO . . . . . . . . A   │
  │ Symbolic link  . . . A   │ mmand on the command line.
R │  F1=Help       F3=Exit   │ a different pathname.
  │  F6=Keyshelp  F12=Cancel │                                More:     +
  └──────────────────────────┘
  ──────────────────────────────────────────────────────────────────────────
  ──────────────────────────────────────────────────────────────────────────
  ──────────────────────────────────────────────────────────────────────────
EUID=10103
```

Figure 8-38 Default actions window settings

Therefore, when you place the cursor under the Type field (File) for file “myfile” in Figure 8-34 on page 149, the next panel displayed is an ISPF edit mode panel for file “myfile”, as shown in Figure 8-39 on page 152.

Archived

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       /u/harry/programx/myfile                        Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> PAGE
****** **************************** Top of Data ******************************
000001 This is a test of creating a file named myfile
****** **************************** Bottom of Data ***************************
```

Figure 8-39 Using the cursor to enter edit mode for a file

FIle mode bits of FSP

By placing the cursor on the permissions field for any file shown in Figure 8-34 on page 149 and pressing Enter, a window appears that allows you to change the permissions, setuid, setgid, or sticky bit by selecting option 1, as shown in Figure 8-40 and Figure 8-41.

In addition, you can also choose to modify the three ACL types from this window, but only if the ACL already exists. Refer to “Using ACLs from the ISHELL” on page 136 for more information about ACLs.

```
  File  Directory  Special_file  Commands  Help
_
_          Select Access Rights
C
  /u/harry/programx/myfile
S  1  1.  Mode bits                                  s.  If / is used also select an
a     2.  Access control list                        ult action will be used.  Select
w     3.  File model ACL                             ect can also be used for quick
n     4.  Directory model ACL
E
                                                     ---Size  Filename    Row 1 of 5
_                                                       8192  ..
_   F1=Help      F3=Exit      F6=Keyshelp               8192  .
_  F12=Cancel                                          88390  o ishell
_                                                        87  myfilename.newfilename.p
_ File  +700  2002-08-02 16:31  HARRY                    47  myfile
```

Figure 8-40 Select permission change window

When you select Option 1, Figure 8-41 appears to allow changes to the FSP fields.

```
  File  Directory  Special_file  Commands  Help
_
_          Change the Mode                          List
S  Change any values and press                      codes.  If / is used also select an
a  Enter.                                           default action will be used.  Select
w                                                  r select can also be used for quick
n  Permissions  . . . . 700
E  Set UID bit  . . . . 0                            ------Size  Filename    Row 1 of 4
   Set GID bit  . . . . 0                                8192  .
_  Sticky bit . . . . . 0                                 8192  ..
_                                                           47  myfile
_   F1=Help          F3=Exit                            88390  o ishell
_   F6=Keyshelp     F12=Cancel
```

Figure 8-41 Change permissions, setuid, setgid, or sticky bit

Change the owner

By placing the cursor under the owner field for any file shown in Figure 8-34 on page 149 and pressing Enter, a window is displayed that permits the assignment of a new file owner for the file, as shown in Figure 8-42 on page 153.

```
  File  Directory  Special_file  Commands  Help
_
                Change the File Owner
C
  /u/harry/programx/myfile
S
a  UID number . . . . . 10103
w  User ID  . . . . . . HARRY
n
E
   F1=Help       F3=Exit       F6=Keyshelp
_  F12=Cancel
_
_ File  +666  2002-08-05 12:54  HARRY
_ File  +700  2002-08-05 16:12  HARRY
_ File  +700  2002-08-02 16:31  HARRY
```

Figure 8-42 Change the file owner

Displaying the file attributes

By placing the cursor under either the Size or Changed-EST5EDT field for a file in Figure 8-34 on page 149, the file attributes are displayed, as shown in Figure 8-43. This is equivalent to selecting the **a** action code.

```
  File  Directory  Special_file  Commands  Help
_
    Edit  Help
S                Display File Attributes
a
w  Pathname : /u/harry/programx/myfile
n                                          More:     +
E  File type . . . . . . : Regular file
   Permissions . . . . . : 700
_  Access control list . : 1
_  File size . . . . . . : 47
_  File owner  . . . . . : HARRY(10103)
_  Group owner . . . . . : SYS1(2)
   Last modified . . . . : 2002-08-02 16:31:25
   Last changed  . . . . : 2002-08-02 16:31:25
   Last accessed . . . . : 2002-08-05 15:44:42
   Created . . . . . . . : 2002-08-02 16:30:54
   Link count  . . . . . : 1
    F1=Help         F3=Exit         F4=Name
    F7=Backward     F8=Forward      F12=Cancel
```

Figure 8-43 Displaying the file attributes using the cursor

Displaying the complete file name

By placing the cursor under a filename in Figure 8-34 on page 149, a new window appears that shows the full path name for that file. This is useful when a file name is truncated on the Directory list panel. The pathname to “myfilename.newfilename.programabc” has a file name that has been truncated in the Directory List panel. The complete pathname is displayed, as shown in Figure 8-44 on page 154.

```
File  Directory  Special_file  Commands  Help
                    Display the Selected Pathname
 The selected pathname is:
                                                          More:     +
    /u/harry/programx/myfilename.newfilename.programabc

  F1=Help       F3=Exit       F6=Keyshelp  F12=Cancel
```

Figure 8-44 Displaying the complete file name

8.2.3 Displaying colors on the Directory List panel

Entries in the Directory List can be highlighted with different colors based on a number of different methods. To set the highlighting options, use one of the following options:

- From the Directory List panel, type: `colors`.
- On the Directory List Panel, select **Commands -> Option 3 - Colors**.
- On the ISHELL panel, select **Options -> Option1 - Directory List**; then type: / on the column **View/change file name highlighting...**.

By choosing one of the options, the Highlighting Options Panel is displayed (Figure 8-45).

```
File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
                    Select colors for file names
 Make color choices and press Enter.
 Leave entry blank for default.

   _  Turn off highlighting

 1   2   3   4   5   6   7   8   9

 Directory  . . _        Extended attribute  . . _
 File . . . . . _        Setuid/setgid . . . . . _
 FIFO . . . . . _        Sticky  . . . . . _
 Char-spec  . . _        Execute permission  . . _
 Symlink  . . . _        Not owned by you  . . . _
                         Truncated name  . . . . _

  F1=Help       F3=Exit       F6=Keyshelp    F12=Cancel
```

Figure 8-45 Highlighting options panel

8.3 Shutting down z/OS UNIX without re-IPLing

z/OS V1R3 has introduced the ability to shut down and reinitialize the z/OS UNIX environment without the need to IPL the system. This new OMVS option will shut down z/OS UNIX and all the processes that are running under it.

OMVS shutdown allows you to do some reconfiguration that would otherwise have required an IPL, for example:

- Reconfiguring a system from a non-shared HFS system to a shared HFS system
- Implementing a new file structure

There are restrictions and limitations that are not allowed with the OMVS shutdown. In these cases, you will need to IPL the system as usual. These limitations include:

- During the cleanup of the resources for z/OS UNIX as part of the shutdown process, some internal failures cannot be resolved using this support, due to their severity.
- OMVS shutdown support cannot be used to install maintenance against z/OS UNIX, because some of the modules are maintained across the shutdown and restart process.
- Installations should avoid using the OMVS restart support as a way to shut down the system with a single command. This will cause some unexpected abnormal terminations of address spaces using UNIX System Services that are not shut down in the manner they recommend.
- OMVS restart support is not intended to be used in an unlimited manner to shut down and restart, because some system resources can be lost during the shutdown phase and because of the disruption it causes to the system.

In order to support OMVS shutdowns, new **Modify** command support for the OMVS address space has been introduced, providing the ability to shut down and then restart the z/OS UNIX environment with the following commands:

```
F OMVS,SHUTDOWN
F OMVS,RESTART
```

8.3.1 Registration support

New registration support has been introduced to allow an application to request special treatment when a shutdown is initiated and to request to receive a new SIGDANGER signal as a warning that shutdown has been initiated and is imminent.

The registration support allows requesting of special treatment in case of shutdown. Different kind of registrations can be implemented as follows:

- A process or job registered as permanent is not taken down across the shutdown and restart process. Its process-related resources are checkpointed at shutdown time and reestablished at restart time, so the registered permanent process or job can survive the shutdown.
- A process or job registered as blocking delays shutdown until it de-registers or ends. This gives the ability for an application to quiesce itself in a more controlled manner before UNIX System Service starts taking down all processes.
- A process or job registered for notification is notified that the shutdown process is being planned via SIGDANGER signal.

The following command has been modified to include information about what type of registration a specific process has:

```
D OMVS,A=ALL
```

As shown in Figure 8-46 on page 156, a character `P` or `B`, indicating permanent or blocked, has been included in the STATE field.

```
D OMVS,A=ALL
BPXO040I 10.02.18 DISPLAY OMVS 543
OMVS     000F ACTIVE          OMVS=(3A)
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS
OMVSKERN BPXOINIT 003C          1          0 MRI--- 07.58.59       .18
  LATCHWAITPID=         0 CMD=BPXPINPR
  SERVER=Init Process                     AF=    0 MF=00000 TYPE=FILE
STC      MVSNFSC5 003B   16908290          1 1R---- 07.59.13       .06
  LATCHWAITPID=         0 CMD=GFSCMAIN
STC      MVSNFSC5 003B   50462724          1 1R---- 07.59.12       .06
  LATCHWAITPID=         0 CMD=BPXVCLNY
STC      MVSNFSC5 003B   50462728          1 1A---- 07.59.14       .06
  LATCHWAITPID=         0 CMD=BPXVCMT
OMVSKERN SYSLOGD5 0041     131081          1 1FI--- 07.59.06       .13
  LATCHWAITPID=         0 CMD=/usr/sbin/syslogd -f /etc/syslog.conf
STC      RMFGAT   0046   84017164          1 1R---P 08.00.01     83.91
  LATCHWAITPID=         0 CMD=ERB3GMFC
TCPIPMVS TCPIPMVS 0043     131085          1 MR---B 08.00.06      8.35
  LATCHWAITPID=         0 CMD=EZBTCPIP
TCPIPMVS TCPIPMVS 0043     131086          1 1R---B 08.00.12      8.35
  LATCHWAITPID=         0 CMD=EZBTTSSL
TCPIPMVS TCPIPMVS 0043     131087          1 1R---B 08.00.12      8.35
```

Figure 8-46 Command now displays process registration

The registration process can be done using the new _shutdown_registration() C function. The BPX1ENV and BPX1SDD callable services have been updated to support shutdown registration.

Attention: Use the **F OMVS,SHUTDOWN** command carefully because this method will take down other system address spaces. As a result, some system-wide resources may not be completely cleaned up during a shutdown and restart.

Do not use this command to shut down and restart the z/OS UNIX environment on a frequent basis. (If you do so, you will eventually have to do a re-IPL.)

8.3.2 Shutting down z/OS UNIX

In order to grant a successful shutdown using OMVS shutdown support and to control the way in what processes are terminated, remember that the shutdown process will stop all the processes that are running. It is strongly recommended that the following steps are done prior to issuing the OMVS shutdown command:

- Quiesce your batch and interactive workloads.

  Once a shutdown request is accepted, jobs that subsequently attempt to use z/OS UNIX services for the first time will be delayed until the restart occurs, and jobs that are already using z/OS UNIX services as dubbed address spaces are sent termination signals and will end abruptly.

- Quiesce major application and subsystem workloads using z/OS UNIX services in the manner that each application or subsystem recommends.

  That will allow subsystems such as DB2, CICS and IMS, and applications like SAP, Lotus Domino, NetView®, and WebSphere to be quiesced in a more controlled manner. The **D OMVS,A=ALL** command can be used to determine the applications that require quiesceing.

- Unmount all remotely mounted file systems, such as those managed by NFS. Doing so will prevent these file systems from losing data.

> **Note:** As of z/OS V1R3, you can specify that file systems are to be automatically unmounted whenever a system leaves the sysplex.

- Shut down TCP/IP and all TCP/IP applications in the manner that TCP/IP recommends, as well as any colony address spaces. This would potentially include NFS and DFS.

> **Attention:** Failure to perform the necessary shutdown and quiesce of the z/OS UNIX workload prior to using this function may result in abnormal terminations for critical system functions (such as TCP/IP, NFS, DFS, and so on) when shutdown is subsequently done. This may cause many failures on the system that will reduce the likelihood that shutdown will succeed.

Starting the shutdown

The shutdown starts by issuing the **F OMVS,SHUTDOWN** command and it follows the process doing the following steps:

1. Once the shutdown command has been accepted, a BPXI055I is issued:

   ```
   *BPXI055I OMVS SHUTDOWN REQUEST ACCEPTED
   ```

   SIGDANGER signals are sent to all processes registered for receiving SIGDANGER signal.

2. If any blocking processes are found, shutdown is delayed until these processes end or deregister as blocking, or if a **F OMVS,RESTART** command is done to restart. If these blocking processes do not end or deregister in a reasonable amount of time, message BPXI064E is displayed to the console indicating shutdown is delayed.

   In our tests, twelve seconds after the shutdown command was accepted, the BPXI064E message was issued. Message BPXI060I was also issued for each process found to be holding up the shutdown. This message identified the job and address space involved, as shown in Figure 8-47.

```
*BPXI064E OMVS SHUTDOWN REQUEST DELAYED
BPXI060I TCPIPMVS RUNNING IN ADDRESS SPACE 0043 IS BLOCKING SHUTDOWN OF OMVS
BPXI060I TCPIPOE RUNNING IN ADDRESS SPACE 0044 IS BLOCKING SHUTDOWN OF OMVS
BPXI060I TCPIPB RUNNING IN ADDRESS SPACE 0052 IS BLOCKING SHUTDOWN OF OMVS
```

Figure 8-47 Shutdown messages for address spacing blocking the shutdown

3. Once all blocking processes have ended or deregister as blocking, the shutdown follows by sending a SIGTERM signal to each non-permanent process found and the following messages are received:

```
BPXP010I THREAD 10652BA800000000, IN PROCESS 67239946, WAS 684
TERMINATED BY SIGNAL SIGTERM, SENT FROM THREAD
1065383000000000, IN PROCESS 1, UID 0.
BPXP018I THREAD 1067C3D000000000, IN PROCESS 131109, ENDED 685
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 04EC6000,
AND REASON CODE 0000FF0F.
BPXP018I THREAD 1067AAC000000000, IN PROCESS 131107, ENDED 686
```

Figure 8-48 Messages received after a SIGTERM signal

If any of these processes do not end after receiving the SIGTERM signal, they are sent a SIGKILL signal and the following messages are received:

```
BPXP010I THREAD 106C2BA000000002, IN PROCESS 131198, WAS 789
TERMINATED BY SIGNAL SIGKILL, SENT FROM THREAD
1065383000000000, IN PROCESS 1, UID 0.
BPXP010I THREAD 1066EEC800000000, IN PROCESS 84017176, WAS 792
TERMINATED BY SIGNAL SIGKILL, SENT FROM THREAD
1065383000000000, IN PROCESS 1, UID 0.
```

Figure 8-49 Messages received after a SIGKILL signal

If, after both of these signals are sent and some of the processes still exist, they are terminated with a 422-1A3 ABEND.

```
IEF450I STEVEZ IKJACCT IKJACCNT - ABEND=S422 U0000 REASON=000001A3 952
        TIME=07.58.28
```

Figure 8-50 ABEND message for terminated address spaces

If, after all of these steps, some non-permanent processes still exist, the shutdown request is aborted and BPXI061E message is issued.

After non-permanent processes have been taken down, the shutdown process continues trying to checkpoint all the permanent processes.

A permanent process cannot be checkpointed; however, a permanent process found using any of these resources will cause shutdown to be aborted and message BPXI060I is issued, indicating what resource for which job is causing the problem.

- Shared libraries
- Memory mapped file services
- Map services
- SRB services
- Semaphore services
- Message queue services
- Shared memory services

4. After all non-permanent processes have ended, BPXOINIT is taken down with a 422-1A3 abend.
5. All file systems are unmounted and potentially moved to another system. If for some reason it is not possible to unmount some file systems, a BPXI066E message is issued and shutdown will proceed to the next phase of shutdown.

   In our tests, we noticed that only one BPXF063I message, which indicates that a file system has been unmounted, is issued. Apparently, it corresponds with the last file system with AUTOMOUNT=N and OWNER of the system being shut down. This can be displayed with the **D OMVS,F** command.
6. The last step in shutdown processing is to clean up all non-essential kernel and LFS resources, and then the following message is issued:

   ```
   BPXN001I UNIX SYSTEM SERVICES PARTITION CLEANUP IN PROGRESS FOR SYSTEM SC64
   ```

   When this is finished, a BPXI056E is issued indicating that shutdown is complete:

   ```
   *BPXI056E OMVS SHUTDOWN REQUEST HAS COMPLETED SUCCESSFULLY
   ```

Shutdown differences

In our tests, we found several differences in the shutdown process versus an IPL complete process related with unmount and movement of file systems, as follows:

- File systems mounted with the UNMOUNT keyword, and file systems mounted with the NOAUTOMOVE keyword, are unmounted on the shutdown process—whereas on a complete IPL, file systems mounted with the NOAUTOMOVE keyword remain mounted with no owner associated (and are thus inaccessible for other systems) and only file systems mounted with the UNMOUNT keyword are unmounted when the system is taken down.

  Refer to “New UNMOUNT option” on page 166 for more information about the UNMOUNT option on the **mount** command.

- In a complete IPL processing, file systems under an automount policy are kept mounted if they have other file systems mounted under it with the AUTOMOVE keyword specified. The shutdown process unmounts file systems mounted with the automount policy, even if they have other file systems mounted on it with AUTOMOVE keyword.

8.3.3 Restarting z/OS UNIX

The **F OMVS,RESTART** command restarts the z/OS UNIX environment. This involves the following:

1. Once the restart command has been accepted, the following message will appear:

   ```
   *BPXI058I OMVS RESTART REQUEST ACCEPTED
   ```

   The first step in the restart process is to reinitialize the kernel and LFS. This includes starting up all physical file systems, as shown in Figure 8-51.

```
BPXF026I FILE SYSTEM HFS.ZOSR03.Z03RD1.ROOT 349 WAS ALREADY MOUNTED.
IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00055
BPXF013I FILE SYSTEM HFS.SC64.DEV 351 WAS SUCCESSFULLY MOUNTED.
IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00056
BPXF013I FILE SYSTEM HFS.SC64.ETC 353  WAS SUCCESSFULLY MOUNTED.
IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00057
BPXF013I FILE SYSTEM HFS.SC64.VAR 355 WAS SUCCESSFULLY  MOUNTED.
BPXF013I FILE SYSTEM /SC64/TMP 356 WAS SUCCESSFULLY MOUNTED.
BPXF203I DOMAIN AF_UNIX WAS SUCCESSFULLY ACTIVATED.
BPXF203I DOMAIN AF_INET WAS SUCCESSFULLY ACTIVATED.
```

Figure 8-51 Restart messages following a system shutdown

2. BPXOINIT is restarted and it will reestablish itself as process ID 1.
3. BPXOINIT reestablishes the checkpointed processes as follows:
   a. All checkpointed processes that are still active are reestablished. Those that are not found are not reestablished and will have their checkpointed resources cleaned up.
4. After BPXOINIT completes its initialization, it will restart /etc/init or /usr/sbin/init to begin full function initialization of the z/OS UNIX environment. /etc/init performs its normal startup processing, invoking /etc/rc.
5. After /etc/init has completed full function initialization, a BPXI0041 message is issued indicating z/OS UNIX initialization is complete:

   ```
   BPXI004I OMVS INITIALIZATION COMPLETE
   ```

In order to monitor the shutdown/restart process, the **D OMVS** command has been modified to include information about the current state of the process. It shows whether OMVS: is shutting down; is shut down; or is restarting. During the shutdown process, the display command shows also a count indicating whether the shutdown request is still proceeding forward. As long as this count continues to increase, it means that shutdown is still processing, as shown in Figure 8-52.

```
D OMVS
BPXO042I 06.23.52 DISPLAY OMVS 221
OMVS     000F SHUTTING DOWN  27   OMVS=(3A)
```

Figure 8-52 D OMVS command showing current state of shutdown processing

8.4 Automount enhancements

z/OS V1R3 introduces several enhancements to help administrators to manage automount. Following is a summary of the changes:

- Display current automount policy
- Support the “#” character as a comment delimiter in the map file
- Allocate an HFS dynamically if needed
- Generic match only on lower case names
- Support system symbolics in the map file

8.4.1 Display current automount policy

A new option in the **automount** command has been introduced to display the current policy in effect. The policy is displayed in a normalized format suitable as input to the automount utility as the .map files with minor editing required, as shown in Figure 8-53.

```
@ SC65:/>/usr/sbin/automount -q
/u
name                 *
filesystem           OMVS.<uc_name>.HFS
type                 HFS
mode                 rdwr
duration             1440
delay                360
```

Figure 8-53 Example output for the display automount command

8.4.2 Support “#” as comment delimiter in map file

In order to provide a more common syntax with shell script files, z/OS V1R3 supports the “#” character as a comment delimiter in map files, as shown in Figure 8-54 on page 161.

```
####################
# Automount Map File for /u #
####################
name                *
type                HFS
filesystem          OMVS.<uc_name>.HFS
mode                rdwr
duration            1440
delay               360
```

Figure 8-54 Example map file with "#" as a comment delimiter

8.4.3 Dynamic HFS allocation in automount

Two new keywords have been introduced in the map file to allocate an HFS dynamically, if is not currently defined at the moment:

allocuser This keyword allocates an HFS only if HFS does not exist and the name matches the user ID.

allocany This keyword allocates an HFS if the HFS does not exist.

The format of these new keywords is as follows:

```
allocuser space_specifications string
allocany space_specifications string
```

Where the space_specifications string specifies typical allocation parameters, such as:

```
space(primary-alloc[,secondary alloc])
 cyl | tracks |  block(block size)
 vol(volser[,volser]...)
 maxvol(num-volumes)
 unit(unit-name)
 storclas(storage-class)
 mgmtclas(management-class)
 dataclas(data-class)
```

The following keywords are automatically added:

```
dsn(filesystem)
dsntype(hfs)
dir(1)
new
```

Map file example

In Figure 8-55 on page 162, the allocany keyword has been added in the map file. In our example, if the HFS does not exist at the moment of the reference of /u/uc_name, an HFS with a primary space of 10 tracks and a secondary space of 5 tracks will be allocated.

Archived

```
####################
# Automount Map File for /u #
####################
name               *
type               HFS
filesystem         OMVS.<uc_name>.HFS
mode               rdwr
duration           1440
delay              360
allocany           space(10,5) tracks
```

Figure 8-55 Example map file with the allocany keyword

The map file syntax is checked by the automount policy at load time. In case of an error, the key number in error is indicated and the policy fails. However, incorrect usage in the allocation specification is not checked at the time the automount policy is processed and will result in allocation failures on usage.

Incorrect map file example

Figure 8-56 shows what happens if an incorrect specification for allocation is used. In our example, we have created an automount policy including a nonexistent dataclas(nonexist). The policy is loaded correctly but at the moment the HFS must be allocated, the allocation fails. Allocation failure message IGD01011I is issued.

```
######################
# Automount Map File for /u #
######################
name            *
type            HFS
filesystem      <uc_name>.HFS
mode            rdwr
duration        nolimit
delay           10
allocany        space(10,5) tracks maxvol(3) dataclas(nonexist)

Messages on SYSLOG:

IEF196I IKJ56893I DATA SET USER1.HFS NOT ALLOCATED+
IKJ56893I DATA SET USER1.HFS NOT ALLOCATED+
IEF196I IGD01011I DATA SET ALLOCATION REQUEST FAILED -
IEF196I ACS DATACLAS ROUTINE RETURNED NONEXIST
IEF196I WHICH DOES NOT EXIST
IGD01011I DATA SET ALLOCATION REQUEST FAILED - 923
ACS DATACLAS ROUTINE RETURNED NONEXIST
WHICH DOES NOT EXIST
```

Figure 8-56 Example of allocation failure for automount

For more information, see *z/OS UNIX System Services Command Reference*, SA22-7802.

8.4.4 Generic match on lowercase names

z/OS V1R3 introduces the new keyword, lowercase, that ensures that a generic match will be done only for lowercase names.

Previously, when automount tried to resolve a lookup request, it attempted to find a specific entry. If a specific entry did not exist for the name being looked up, it attempted to use the generic entry (name *). The generic match could be in both lowercase or uppercase.

Then, automount automatically mounted the HFS data set based on the MapName policy that indicates the name of the HFS to be mounted. The name of the HFS to be mounted can include the following special symbols to provide name substitution:

<asis_name> This represents the exact name of the subdirectory to be “automounted”. If the name is in uppercase, the substitution name in the HFS name will be in uppercase. If the name is in lowercase, the substitution name will result in lowercase. That means that we are using <asis_name> keyword as shown in Figure 8-57.

> **Note:** The access to /u/testauto will result in a catalog error, whereas the access to /u/TESTAUTO will succeed.

<uc_name> This represents the name of the subdirectory to be “automounted” in uppercase characters. Note than in this case, /u/user1 and /u/USER1 mount point directories map the same file system.

> **Note:** The new keyword **`lowercase`** allows you to define if a generic entry will match names with lowercase or not. Two options are available: lowercase[YES] and lowercase[NO].

lowercase[YES] This indicates that only names in lowercase (special characters are also allowed) will match the * specification.

lowercase[NO] This is the default and indicates that *any* names will match the * specification.

```
######################
# Automount Map File for /u #
######################
name            *
type            HFS
filesystem      <asis_name>.HFS
mode            rdwr
duration        nolimit
delay           10
allocany        space(10,5) tracks
```

Figure 8-57 Automount map file

As an example, suppose we activate the policy described in Figure 8-58 on page 164.

```
####################
# Automount Map File for /u #
####################
name                *
type                HFS
filesystem          OMVS.<uc_name>.HFS
mode                rdwr
duration            1440
delay               360
allocany            space(10,5) tracks maxvol(3)
lowercase           yes
```

Figure 8-58 Automount map file with lowercase keyword

Because lowercase=yes is specified, access by a /u/USER1 will not match the generic entry and this results in a failure for the automount load. By accessing with /u/user1, we will be successful.

Note: Note that the `lowercase` keyword with <asis_substitution> in the HFS name will also result in an error when automount is requested.

8.4.5 Support system symbols in map file

z/OS V1R3 introduces the support of system symbols in the map file to provide name substitution in the file system name.

Figure 8-59 on page 165 shows the use of system symbols &SYSNAME and &SYSPLEX for the automount policy.

Note: Symbol substitution is done when the automount policy is loaded, not when the rule is used to resolve a mountpoint.

Archived

```
@ SC65:/>/usr/sbin/automount -q
/u2

name                 *
filesystem           OMVS.&SYSNAME..<uc_name>.HFS
type                 HFS
allocany                       space(10,5) tracks
mode                 rdwr
duration             1440
delay                360


/u

name                 *
filesystem           OMVS.&SYSPLEX..<uc_name>.HFS
type                 HFS
allocany                       space(10,5) tracks
mode                 rdwr
duration             1440
delay                360
```

Figure 8-59 Map files including system symbols

Attention: The use of <SYSTEM> will be withdrawn in a future release, so use &SYSNAME instead.

8.5 Copytree utility

A supported copytree utility REXX sample is included in /samples/copytree in z/OS V1R3 to ensure the appropriate version of the product.

Copytree provides the ability to copy a tree in a file system under another directory, preserving all file attributes (including ACLs), or to check a tree for structural integrity; it can run under TSO or the shell.

8.6 Shared HFS unmount option

Previous to z/OS V1R3, file systems could be mounted as AUTOMOVE and NOAUTOMOVE. If AUTOMOVE is specified, the file system is moved to another system in the event that the owning system is taken down. If NOAUTOMOVE is specified, the file system remains mounted when the owning system goes down, but the file system now has an unknown owner, as shown in Figure 8-60.

```
HFS           445 UNOWNED                    RDWR
  NAME=WTSCPLX2.SC64.SYSTEM.HFS
  PATH=/SC64
  OWNER=         AUTOMOVE=N CLIENT=Y
```

Figure 8-60 Display of file system showing unknown owner

When the failed system reinitializes, the file system will recover and become active again.

8.6.1 New UNMOUNT option

A new UNMOUNT option has been added in order to unmount file systems associated with a failed system. This allows for file systems that are required or desirable to not move to another system to be unmounted. This avoids either recovering or converting them to “unowned” status. Therefore, the options now are:

```
AUTOMOVE|NOAUTOMOVE|UNMOUNT
```

AUTOMOVE Specifies that ownership of the file system is automatically moved to another system. It is the default.

NOAUTOMOVE Specifies that the file system will not be moved if the owning system goes down and the file system is not accessible.

UNMOUNT Specifies that the file system will be unmounted when the system leaves the sysplex.

> **Note:** This option is not available for automounted file systems.

Mount commands changes

The new UNMOUNT option is now supported in the following ways:

- BPXPRMxx parmlib MOUNT statement

  The AUTOMOVE | NOAUTOMOVE | UNMOUNT parameters on the ROOT and MOUNT statements indicate what happens to the file system if the system that owns that file system goes down.

- TSO/E **MOUNT** command

  ```
  MOUNT filesystem(OMVS.HFS1.HFS) mountpoint('/u/vivarhfs') type(HFS) mode(rdwr)
  UNMOUNT
  ```

- **mount** shell command

  ```
  mount [-t fstype][-rv][-a yes|no|unmount][-o fsoptions][-d destsys][-s
  nosecurity|nosetuid] -f fsname pathname
  ```

- **SETOMVS** command

  ```
  SETOMVS FILESYS,FILESYS=filesystem,AUTOMOVE=YES|NO|UNMOUNT
  ```

- Shell **chmount** command:

  ```
  chmount [-R [-D |-d destsys][-a yes|no|unmount]pathname...
  ```

- The ISHELL mount interface in the Mount File System panel, as shown in Figure 8-61 on page 167, is accessed from the ISHELL by selecting **-> File System -> Option 3 - Mount)**. The new mount option is: **Automove unmount file system**.

Archived

```
File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
                          Mount a File System

Mount point:
                                                          More:     +
   /var/

File system name . . _
File system type . . _________  New owner  . . . . . _________
Owning system  . . . _________  Character Set ID . . _____

Select additional mount options:
_  Read-only file system          _  Do not automove file system
_  Ignore SETUID and SETGID       _  Automove unmount file system
_  Bypass security                _  Text conversion enabled

Mount parameter:
```

Figure 8-61 Mount a File System panel

Display commands changes

The following display commands have been modified to include information about the UNMOUNT option:

- **D OMVS,F** includes an U character in the AUTOMOVE field for “UNMOUNT” mounted file systems, as shown in Figure 8-62.

```
HFS           422 ACTIVE                    RDWR
  NAME=OMVS.TESTCD.HFS
  PATH=/TESTC/TESTCD
  OWNER=SC64     AUTOMOVE=U CLIENT=N
```

Figure 8-62 Display to show unmount option for a HFS data set

- shell **df -v** command

```
Mounted on     Filesystem              Avail/Total    Files      Status
/TESTC         (OMVS.TESTC.HFS)        14192/14400    4294967291 Available
HFS, Read/Write, Device:478, ACLS=Y
File System Owner : SC64        Automove=U      Client=N
Filetag : T=off   codeset=0
```

Figure 8-63 df -v command output to show unmount option

- **F BPXOINIT,FILESYS=D,ALL** and **F BPXOINIT,FILESYS=D,FILESYSTEM=filesystem** commands

Archived

```
OMVS.TESTC.HFS                                          478  RDWR
  PATH=/TESTC
  STATUS=ACTIVE                  LOCAL STATUS=ACTIVE
  OWNER=SC64         RECOVERY OWNER=SC64     AUTOMOVE=U PFSMOVE=Y
  TYPENAME=HFS       MOUNTPOINT DEVICE=       1
  MOUNTPOINT FILESYSTEM=WTSCPLX2.SYSPLEX.ROOT
  ENTRY FLAGS=90000000  FLAGS=40000018  LFSFLAGS=00000000
  LOCAL FLAGS=40000018  LOCAL LFSFLAGS=20000000
```

Figure 8-64 BPXOINIT commands to display unmount option

8.7 Mount table limit monitoring

In previous releases, users needed the capability to determine when the number of file system mounts in a shared HFS was approaching the configured limit. Before z/OS V1R3, there was no way to easily determine when the mount limit, specified in the BPXMCDS CDS shown in Figure 8-65, was being approached.

z/OS V1R3 introduces the possibility to monitor the shared HFS mount limits, specified in the CDS, by issuing a console message when the limit has almost been reached.

BPXMCDS couple data set

Shared HFS support uses a type BPXMCDS couple data set (CDS) to maintain data about mounted file system in the sysplex configuration. The primary and alternate CDS are formatted, using the IXCL1DSU utility, with a maximum number of mount entries as specified in the NUMBER value that specifies the number of mounts, as shown in Figure 8-66 on page 169.

```
ITEM NAME(MOUNTS) NUMBER(750)
 /* Specifies the number of MOUNTS that can be supported by OMVS.*/
   Default =  100
   Suggested minimum =   10
   Suggested maximum = 35000  */
 ITEM NAME(AMTRULES) NUMBER(50)
  /* Specifies the number of automount rules that can be supported by OMVS */
    Default =   50
    Minimum =   50
    Maximum =  1000        */
```

OMVS couple data set

Figure 8-65 Mount and automount entries for shared sysplex support

```
//STEP10   EXEC PGM=IXCL1DSU,REGION=0M
//STEPLIB  DD   DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  /* Begin definition for OMVS couple data set (1)                  */
     DEFINEDS SYSPLEX(SANDBOX)         /* Name of the sysplex in
                                          which the OMVS couple data
                                          set is to be used.         */
              DSN(SYS1.XCF.OMVS05) VOLSER(SBOX63) /* The name and
                                          volume for the OMVS
                                          couple data set. The
                                          utility will allocate a
                                          new data set by the name
                                          specified on the volume
                                          specified.                 */
              MAXSYSTEM(8)             /* Number of systems in the
                                          sysplex to be supported by
                                          this couple data set. Default
                                          value is eight.       @01A*/
CATALOG                  /* Default is not to CATALOG.
                                                                 @01C*/
          DATA TYPE(BPXMCDS)           /* The type of data in the
                                          data set being created is
                                          for OMVS. BPXMCDS is the
                                          TYPE for OMVS.             */
               ITEM NAME(MOUNTS) NUMBER(750) /* Specifies the number of
                                          MOUNTS that can be supported
                                          by OMVS.
                                          Default =   100
                                          Minimum =     1
                                          Maximum = 50000        @D1C*/
               ITEM NAME(AMTRULES) NUMBER(50) /* Specifies the number
                                          of automount rules that can
                                          be supported by OMVS.
                                          Default =    50
                                          Minimum =    50
                                          Maximum =  1000        @D1A*/
```

Figure 8-66 Job that creates the BPXMCDS couple data set

Displaying the mount table limit

Once the mount limit is reached, no more file systems can be mounted in the sysplex until a larger type BPXMCDS CDS is enabled. Mount table limit monitoring allows an installation to detect when a primary CDS is reaching its mount table limit in order to begin corrective actions before denial of service.

You can display the number of mount entries and the number in use by using the **F BPXOINIT,FILESYS=DISPLAY,GLOBAL** command, as shown in Figure 8-67 on page 170.

```
f bpxoinit,filesys=display,global
BPXF041I 2002/05/09 13.42.25 MODIFY BPXOINIT,FILESYS=DISPLAY,GLOBAL
221
SYSTEM   LFS VERSION ---STATUS--------------------- RECOMMENDED ACTION
SC64       1.  3.  1 VERIFIED                       NONE
SC63       1.  3.  1 VERIFIED                       NONE
SC65       1.  4.  1 VERIFIED                       NONE
CDS VERSION=  1        MIN LFS VERSION=  1.  3.  1
BRLM SERVER=N/A        DEVICE NUMBER OF LAST MOUNT=     706
MAXIMUM MOUNT ENTRIES=     500   MOUNT ENTRIES IN USE=     430
```

Figure 8-67 Display of mount table limits

BPXPRMxx parmlib member

Mount table limit monitoring is enabled by specifying the LIMMSG parameter in BPXPRMxx parmlib member, or dynamically by using **SETOMVS** command, with the values SYSTEM or ALL:

LIMMSG=SYSTEM Console messages are to be displayed for all processes that reach system limits. In addition, messages are to be displayed for each process limit of a process if:

The process limit or limits are defined in the OMVS segment of the owning user ID

The process limit or limits have been changed with a **SETOMVS PID=pid,proces_limit** command

LIMMSG=ALL In this case, console messages are to be displayed for the system limits and for the process limits, regardless of which process reaches a process limit.

For more information about LIMMSG parameter, see *z/OS V1R3 System Commands*, SA22-7627.

Display the LIMMSG parameter

Both LIMMSG values were defined in previous z/OS releases, and you can monitor the current value of LIMMSG parameter option using the **D OMVS,LIMITS** command, as shown in Figure 8-68 on page 171.

Archived

```
D OMVS,LIMITS
BPXO051I 10.43.10 DISPLAY OMVS 747
OMVS     000F ACTIVE          OMVS=(4A)
SYSTEM WIDE LIMITS:         LIMMSG=SYSTEM
                  CURRENT  HIGHWATER     SYSTEM
                    USAGE      USAGE      LIMIT
MAXPROCSYS             42         54        300
MAXUIDS                 1          2         50
MAXPTYS                 0          3        256
MAXMMAPAREA             0          0       4096
MAXSHAREPAGES           0          0   32768000
IPCMSGNIDS             10         10      20000
IPCSEMNIDS              0          0      20000
IPCSHMNIDS              0          0      20000
IPCSHMSPAGES            0          0    2621440
IPCMSGQBYTES          ---         72     262144
IPCMSGQMNUM           ---          6      10000
IPCSHMMPAGES          ---          0      25600
```

Figure 8-68 Display the BPXPRMxx parameter specifications

Mount table limit monitoring messages

Once the LIMMSG parameter is set to SYSTEM or ALL, a BPXI043E console message will be issued when the mount table limit reaches a critical value. The BPXI043E message has the following format:

```
BPXI043E MOUNT TABLE LIMIT HAS REACHED <limit_reached> OF ITS CURRENT CAPACITY OF
<current_limit>
```

Where:

<limit_reached> Has the value of 85, 90, 95, or 100

<current_limit> Indicates the mount NUMBER value in the BPXMCDS CDS.

The message is updated when the percentage, limit_reached field, has changed to a new value (from 85 to 90, from 90 to 95, or from 95 to 100), as shown in Figure 8-69. The message is deleted when the percentage decreases below 85%.

```
*BPXI043E MOUNT TABLE LIMIT HAS REACHED 85% OF ITS CURRENT CAPACITY OF 500
*BPXI043E MOUNT TABLE LIMIT HAS REACHED 90% OF ITS CURRENT CAPACITY OF 500
*BPXI043E MOUNT TABLE LIMIT HAS REACHED 100% OF ITS CURRENT CAPACITY OF 500
BPXI045I THE PRIMARY CDS SUPPORTS A LIMIT OF 700 MOUNTS AND A LIMIT OF 50 AUTOMOUNT
RULES.
BPXI044I RESOURCE SHORTAGE FOR MOUNT TABLE HAS BEEN RELIEVED.
```

Figure 8-69 Mount table limit monitoring messages

Mount table limit procedure

When the BPXI043E message has been issued, you must begin corrective actions before the limit reaches the 100% limit, which will provoke denial of service for new mounts. The corrective actions may consist of the following steps:

- Use the following steps to switch to an existing and enabled alternate CDS, which is presumably defined with more mount entries:
  - Format a new, larger type BPXMCDS by increasing the mount limits

– Once the CDS is defined, it can be enabled as the alternate CDS using the following command:

```
SETXCF COUPLE,TYPE=BPXMCDS,ACOUPLE=(alternate_name,alternate_volume)
```

– Finally, switch the alternate CDS to the primary CDS by using the following command:

```
SETXCF COUPLE,PSWITCH
```

Once the corrective actions have been made and the new larger primary CDS has been enabled, the following console messages are issued:

```
BPXI045I THE PRIMARY CDS SUPPORTS A LIMIT OF 700 MOUNTS AND A LIMIT OF 50 AUTOMOUNT
RULES.
BPXI044I RESOURCE SHORTAGE FOR MOUNT TABLE HAS BEEN RELIEVED.
```

Figure 8-70 Messages issued when a new BPXMCDS CDS is enabled

The BPXI045I message is issued when a PSWITCH occurs.

8.7.1 Shared HFS support for confighfs command

The **/usr/lpp/dfsms/bin/confighfs** shell command is used to perform certain functions directly with the HFS physical file system which include:

- Query HFS limits
- Query HFS global statistics
- Setting HFS virtual and fixed storage pool sizes

Previously, there was a restriction since OS/390 V2R9 implementation for shared HFS support that the **confighfs** command would only provide file system data for file systems that were mounted as RDWR if the command was issued from the owner system. Otherwise, the **confighfs** command failed with the following message:

```
Error issuing PFSCTL: RC=0 ERRNO=129(81) REASON=5B360105
HFS is not mounted on this system/LPAR
```

This restriction has been removed in z/OS V1R3 and support has been added so that the **confighfs** command can now be issued from any system for any active HFS file system.

Note: This command can now be issued from any system within a sysplex at z/OS V1R3 or later, assuming that the system on which the file system is mounted is also running z/OS V1R3 or later. Otherwise, the command will fail.

9

UNIX System Services enhancements in z/OS V1R4

In this chapter we discuss the enhancements introduced in z/OS V1R4 UNIX System Services. The following topics are described:

- Shared HFS enhancements, which include:
  - Automove system list
  - Byte-range locking (BRLM) enhancements
- Enhancements for unique UIDs and GIDs

Archived

© Copyright IBM Corp. 2003

9.1 Automove system list

z/OS V1R4 includes the capability to specify a prioritized automove system list to indicate which system will become the new owner for a file system in a shared file system environment, in the event of a loss of the owning system. In the shared sysplex environment shown in Figure 9-1, system SC63 is the owning system of the zFS file system OMVS.CMP01.ZFS.

Figure 9-1 Shared file system sysplex environment

The automove system list is defined using the AUTOMOVE parameter on any one of the following methods of mounting a file system:

- BPXPRMXX parmlib member MOUNT statement or TSO/E MOUNT command
- Shell command
- ISHELL panels
- C program, assembler program, or REXX program

The system list can be changed for a file system after it has been mounted and can also be displayed.

9.1.1 Automove system list specification

The automove system list can be specified in many different ways to automove a file system. The list begins with an indicator to either include or exclude, followed by a list of system names. The indicator can be abbreviated as “i” or “e”.

Specify the indicator as follows:

i Use with a system list to provide a prioritized list of systems to which the file system may be moved if the owning system goes down. The list of systems is in priority order and if none of the systems specified in the list can take over as the new owner, the file system will be unmounted.

e Use with a system list to provide a list of systems to which the file system may not be moved.

Note: Is not possible to define an include and exclude list at the same time for the same file system. One of the options will override the previously defined option.

BPXPRMxx parmlib specification

A new operand has been added to the AUTOMOVE keyword on the MOUNT statement, as shown in Figure 9-2.

```
AUTOMOVE(indicator,name1,name2,...,nameN)
```

```
mount filesystem(omvs.test1.hfs) mountpoint('/tmp/test1') type(hfs) mode(rdwr)
automove(i,sc64,sc65)
```

Figure 9-2 Mount statement showing the new AUTOMOVE options

Note: The automove system list is optional. If not specified, the system which will become the new server is randomly chosen.

Mount shell command

The **MOUNT** command issued from an OMVS shell session has been modified to include an automove system list specification, as follows:

```
mount [-t fstype] [-rv] [-a yes|include,sysname1,... sysnameN |exclude,sysname1,...
sysnameN |no|unmount] [-o fsoptions] [-d destsys] [-s nosecurity|nosetuid] -f fsname
pathname
```

An example of the new option:

```
mount -a i,SC64,SC65 -f OMVS.CMP01.ZFS /tmp/test
```

ISHELL panels

The ISHELL panel for mounts allows a selection to set the automove attribute. If selected, it displays a new panel to choose the automove type and specify a list of up 32 systems (refer to Figure 9-4 on page 176). This panel is accessed from the Main ISHELL Panel; select **File_systems** -> **Option 1 - Mount Table** -> **Modify**.

When you place an `M` for modify next to a mounted file system, the Select the attribute to change window is displayed; see Figure 9-3. This window is modified in z/OS V1R4, with Option 3 being new and replacing Option 3 and 4 from the previous releases.

Archived

```
                         Work with Mounted File Systems
C
                         Select the attribute to change
S
U   Select the attribute to change:
    3   1.  Change mount mode to R/O                                    8
        2.  Change Owning system from SC65
        3.  Change automove attribute...

    New owning system      _________

     F1=Help       F3=Exit       F6=Keyshelp  F12=Cancel
    OMVS.SC63.XML.HFS                                Available
    OMVS.SC64.USRLOCAL                               Available
    OMVS.SYSLOGD.HFS                                 Available
    OMVS.TWS810.TWSCTP.HFS                           Available
    WTSCPLX2.DOMINO.HFS                              Available
    WTSCPLX2.JAVA118.V001122.HFS                     Available
    WTSCPLX2.JAVA213.V010424.HFS                     Available
    WTSCPLX2.JAVA213.V010714.HFS                     Available
    WTSCPLX2.JAVA213.V010926.HFS                     Available
    WTSCPLX2.MQSERIES.MA88.HFS                       Available
    WTSCPLX2.SC63.SYSTEM.HFS                         Available
    WTSCPLX2.SC64.SYSTEM.HFS                         Available
    WTSCPLX2.SC65.SYSTEM.HFS                         Available
    WTSCPLX2.SYSPLEX.ROOT                            Available
```

Figure 9-3 Select the attribute to change window

Selecting Option 3 displays the new window shown in Figure 9-4 which allows you create a modify an automove system list. Select option 4 or 5 to either include or exclude the systems you specify by system names.

```
                         Work with Mounted File Systems
C
                         Select the attribute to change
S
U   S                          Set automove attribute
    3
    Select the automove attribute:
    1   1.  Yes
        2.  No
    N   3.  Unmount
        4.  Include systems
        5.  Exclude systems

    System names for Include or Exclude
    ________  ________  ________  ________  ________
    ________  ________  ________  ________  ________
    OMV ________  ________  ________  ________  ________
    OMV ________  ________  ________  ________  ________
    OMV ________  ________  ________  ________  ________
    OMV ________  ________  ________  ________  ________
    WTS ________  ________
    WTS
    WTS  F1=Help       F3=Exit       F6=Keyshelp  F12=Cancel
    WTS
    WTSCPLX2.JAVA213.V010926.HFS                     Available
    WTSCPLX2.MQSERIES.MA88.HFS                       Available
    WTSCPLX2.SC63.SYSTEM.HFS                         Available
    WTSCPLX2.SC64.SYSTEM.HFS                         Available
    WTSCPLX2.SC65.SYSTEM.HFS                         Available
    WTSCPLX2.SYSPLEX.ROOT                            Available
```

Figure 9-4 Set automove attribute window

C program

Using the callable service _mount(BPX2MNT), the syslist and indicator may be specified in the MNTE as input.

9.1.2 Changing an automove system list

After a file system is mounted, the AUTOMOVE attribute can be changed by using one of the following commands:

- **setomvs** command

  ```
  SETOMVS FILESYS,FILESYSTEM=filesystem,AUTOMOVE=YES | NO| UNMOUNT |
  indicator(sysname1,sysname2,...,sysnameN)
  ```

  ```
  setomvs filesys,filesystem='omvs.cmp01.zfs',automove=e(sc64)
  ```

- **chmount** shell command

  ```
  chmount [-R][-r][-w][-D | -d destsys] [-a yes | no | unmount |
  indicator,sysname1,..., sysnameN] pathname
  ```

  ```
  chmount -a i,SC64,SC65 /tmp/test
  ```

Commands to display the automove system list

Console and shell display commands have been modified in order to give information about the automove system list.

- The **DISPLAY OMVS,FILE** command displays the system list and indicator, if the system list has been defined; see Figure 9-5 on page 177.

```
d omvs,file
HFS           126 ACTIVE                    RDWR
  NAME=OMVS.CMP01.ZFS
  PATH=/SC63/tmp/test1
  OWNER=SC63 AUTOMOVE=I CLIENT=N
  INCLUDE SYSTEM LIST:  SC64     SC65
```

Figure 9-5 z/OS command to display the automove system list

- The **df -v** command provides system list information and the indicator, if the system list has been defined, see Figure 9-6.

```
@ SC65:/>df -v /tmp/test1
Mounted on     Filesystem                Avail/Total    Files      Status
/SC65/tmp/test1 (OMVS.TEST1.HFS)          14224/14400    4294967294 Available
HFS, Read/Write, Device:126, ACLS=Y
File System Owner : SC65        Automove=I      Client=N
System List (Include) : SC64       SC63
Filetag : T=off   codeset=0
```

Figure 9-6 OMVS shell command to display the automove system list

- **F BPXOINIT,FILESYS=DISPLAY,ALL** has been modified to display syslist information; see Figure 9-7 on page 178.

Archived

```
OMVS.TEST1.HFS                                          126  RDWR
  PATH=/SC63/tmp/test1
  STATUS=ACTIVE                 LOCAL STATUS=ACTIVE
  OWNER=SC63        RECOVERY OWNER=SC63     AUTOMOVE=I PFSMOVE=Y
  TYPENAME=HFS      MOUNTPOINT DEVICE=      72
  MOUNTPOINT FILESYSTEM=/SC63/TMP
  ENTRY FLAGS=91000000  FLAGS=40000000  LFSFLAGS=00000000
  LOCAL FLAGS=40000000  LOCAL LFSFLAGS=20000000
  SYSLIST STS=00000000  SYSLIST VALID=00000000
  INCLUDE SYSTEM LIST (2 SYSTEM(S) IN LIST):
    SC64     SC65
```

Figure 9-7 z/OS command to display the automove system list

9.2 Byte-range locking in a shared HFS environment

With z/OS V1R4, you can lock all or part of a file that you are accessing for read-write purposes by using the byte range lock manager (BRLM). As a default, the lock manager is initialized on only one system in the sysplex. The first system that enters the sysplex initializes the BRLM and becomes the system that owns the manager. This is called a “centralized BRLM”.

In a sysplex environment, a single BRLM handles all byte-range locking in the shared HFS group. If the BRLM server crashes, or if the system that owns the BRLM is partitioned out of the sysplex, the BRLM is reactivated on another system in the group. All locks that were held under the old BRLM server are lost. An application that accesses a file that was previously locked receives an I/O error, and has to close and reopen the file before continuing.

Distributed BRLM

You can choose to have distributed BRLM initialized on every system in the sysplex. Each BRLM is responsible for handling locking requests for files whose file systems are mounted locally in that system. Use distributed BRLM if you have applications which lock files that are mounted and owned locally.

For distributed BRLM to be activated, the z/OS UNIX couple data set (BPXMCDS) must be updated as shown in Figure 9-8, and the supported code must be installed and running on each system. See APAR OW52293 for more information.

```
ITEM NAME(DISTBRLM) NUMBER(1)
     /*Enables conversion to a distributed BRLM.
       1, distributed BRLM enabled,
       0, distributed BRLM is not enabled during next sysplex IPL
       Default = 0  */
```

OMVS couple data set

Figure 9-8 Update BPXMCDS for BRLM

This support allows you to change to using distributed BRLM (rather than a single, central BRLM) in the sysplex. With distributed BRLM, each system in the sysplex runs a separate BRLM, which is responsible for locking files in the file systems that are owned and mounted on that system. Because most applications (including cron, inetd, and Lotus Domino) lock local files, the dependency on having a remote BRLM up and running is removed. Running with distributed BRLM is optional.

z/OS R1V4 implements the first phase of moveable BRLM in a sysplex. Moveable BRLM provides the capability of maintaining the byte range locking history of applications, even when a member of the sysplex dies. This first phase will focus on distributing the locking history across all members of the sysplex. As a result, many applications that lock files that are locally mounted will be unaffected when a remote sysplex member dies. Movement away from a centralized to a distributed BRLM will provide greater flexibility and reliability.

BRLM and callable services

If you use the BPX1FCT or BPX1VLO callable services or the **fcntl()** or **lockf()** C functions to do byte-range locking in a shared HFS sysplex environment, you should be aware of the recovery scenario that was introduced with the shared HFS support.

> **Note:** The following C functions use byte-range locking internally, and can result in the same recovery scenario: **endutxent(), getutxent(), getutxid(), getutxline(), pututxline(), setutxent(), __utmpname(),** and **__utxtrm()**.

BRLM coexistence and maintenance

Distributed BRLM support is added in the following PTFs for downlevel systems:

- UW85157 (OS/390 V2R9)
- UW85155 (OS/390 V2R10)
- UW85156 (z/OS V1R2).

Additional support is added in PTFs; when the BRLM server crashes, a default SIGTERM is issued against any process that has used byte-range locking and has an open file that was locked. Users can specify a preferred signal to be used instead of the default SIGTERM.

- UW75787 (V2R9)
- UW75786 (V2R10)

9.3 Shared HFS availability enhancement

The following updates have been made for shared HFS:

- The type BPXMCDS couple data set has changed to hold additional data; you must reformat the OMVS couple data set, DATA TYPE(BOXNCDS), using the V1R4 level of the SYS1.MIGLIB.
- The size of the PARM parameter on the MOUNT statement in the BPXPRMxx parmlib member has been reduced to 500 characters.

9.4 Enhancements for UID/GID support

Enhancements in z/OS V1R4 have been made in the way that UIDs and GIDs can be assigned by RACF. They can be automatically assigned to new users and then either prevented from being shared, or allowed to be shared.

New keywords have been added to the **SEARCH** command in order to determine the set of users assigned to a given UID, or the groups assigned to a given GID.

The RACF UNIXMAP class is currently used to allow the system to quickly look up a user ID from a UID, or a group name from a GID. An enhanced RACF database organization can now perform this conversion quickly without requiring the mapping profiles in the UNIXMAP class. Use the IRRIRA00 utility, if desired, to convert the RACF database to the new organization.

9.4.1 RACF database and AIM

To use a new RACF profile, SHARED.IDS, and the new SEARCH keywords, the RACF database must have application identity mapping (AIM) stage 2 or 3 implemented. You can convert your RACF database to stage 3 of application identity mapping by using the IRRIRA00 conversion utility. See the *z/OS Security Server RACF System Programmer's Guide*, SA22-7681 for information about running the IRRIRA00 conversion utility.

If your installation is new to RACF and you are not running any releases prior to OS/390 Version 2 Release 10, you will automatically take advantage of application identity mapping at the stage 3 level without running the IRRIRA00 conversion utility, and you will not need to use VLF and UNIXMAP to achieve improved performance.

IRRIRA00 utility

This utility was new in OS/390 V2R10. It converts an existing RACF database to application identity mapping functionality using a four-staged approach.

With a database created at OS/390 V2R10 or higher, RACF's application identity mapping uses an alias index to associate user and group names with:

- Lotus Notes® for z/OS
- Novell Directory Services for OS/390
- z/OS UNIX identities

RACF typically uses mapping profiles for this purpose with databases created before Release 10. However, you can use the IRRIRA00 utility to convert the database to work with an alias index instead. The conversion consists of a series of stage transitions from zero to three. RACF uses the database mapping information differently for each stage.

Note: Systems with an older release are not aware of an alias index or the application identity mapping conversion stages, so use care when sharing a database between older systems and a system at Release 10 or higher.

AIM stage 3

In stage 3, RACF locates application identities, such as UIDs and GIDs, for users and groups by using an alias index that is automatically maintained by RACF. This allows RACF to more efficiently handle authentication and authorization requests from applications such as z/OS UNIX than was possible using other methods, such as the UNIXMAP class and VLF. Once your installation reaches stage 3 of application identity mapping, you will no longer have UNIXMAP class profiles on your system, and you can deactivate the UNIXMAP class and remove VLF classes IRRUMAP and IRRGMAP.

Important: Associating RACF user IDs and groups to UIDs and GIDs has important performance considerations. If your installation shares the RACF database with systems running releases prior to OS/390 Version 2 Release 10, it is important to use the VLF classes IRRUMAP and IRRGMAP and the UNIXMAP class to improve performance by avoiding sequential searches of the RACF database for UID and GID associations.

If your installation shares the RACF database with only systems running z/OS, or OS/390 Version 2 Release 10 or above, you may be able to achieve improved performance without using UNIXMAP and VLF. However, before you can avoid using UNIXMAP and VLF, you need stage 3 of application identity mapping by running the IRRIRA00 conversion utility.

9.4.2 Search enhancements to map UIDs and GIDs

The use of the new SEARCH keywords, USER and GROUP, requires at least AIM Stage 2 installed and it does not require any particular authority. When a UID or GID is specified, all other keywords, except CLASS, are ignored. The RACF search command is as follows:

```
SEARCH CLASS(USER) UID(0)
```

Use the UID keyword, shown in Figure 9-9, to obtain a list of all *users* assigned to the UID specified.

```
SEARCH CLASS(USER) UID(1)
LDAPSRV
MSYSLDAP
```

Figure 9-9 Display all users with a UID=1

Use the GID keyword, shown in Figure 9-10, to obtain a list of all *groups* assigned to the GID specified.

```
SEARCH CLASS(GROUP) GID(1)
OMVSGRP
```

Figure 9-10 Display all groups with a GID=1

RACF SEARCH panel

If you prefer to use RACF panels, the SEARCH panel for both user and group profiles has been modified to support this new enhancement on the SEARCH command, as shown in Figure 9-11 on page 182, with the UID selection criteria.

```
                         RACF - SEARCH FOR USER PROFILES
COMMAND ===>

ENTER OPTIONAL SELECTION CRITERIA:

   MASK1    ________     Selects profiles with names that begin with the
                         specified character string.

   MASK2    ________     Selects profiles with names that contain the
                         specified string somewhere after MASK1.

   FILTER   ________     Selects profiles with names that match the
                         specified string.

   AGE      ____         Selects users that have not accessed the system
                         in the number of days specified.

   USERID   ________     Selects the profiles this user is authorized to see
                         (administrators only).

   UID      1________    Selects profiles which have this UID defined in
                         the OMVS segment (other options will be ignored).
                         Valid values are 0 - 2147483647.

   ___ Enter YES to generate a TSO clist
       (Command Direction is inactivated for SEARCH with clist)

   ___ Enter YES to specify additional SEARCH criteria
```

Figure 9-11 RACF search for user profiles panel

If you searched for UID(1) users by entering a 1 for the UID (as shown in Figure 9-11), you should receive the following response:

```
 BROWSE - RACF COMMAND OUTPUT----------------------- LINE 00000000 COL 001 080
 COMMAND ===> _                                          SCROLL ===> HALF
******************************** Top of Data ********************************
LDAPSRV
MSYSLDAP
****************************** Bottom of Data ******************************
```

Figure 9-12 RACF search response for UID(1)

9.4.3 Shared UID prevention

In order to prevent several users from having the same UID number, a new RACF SHARED.IDS profile has been introduced in the UNIXPRIV class. This new profile acts as a system-wide switch to prevent assignment of an UID that is already in use.

The use of the SHARED.IDS profile requires AIM stage 2 or 3, and its uniqueness is guaranteed within the scope of the GRSplex.

To enable shared UID prevention, you must define a new SHARED.IDS profile in the UNIXPRIV class as follows:

```
RDEFINE UNIXPRIV SHARED.IDS UACC(NONE)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Once the SHARED.IDS profile has been defined and the UNIXPRIV class refreshed, it will be not allow a UID to be assigned if the UID is already in use. The same is true for GIDs; it will not allow a GID to be shared between different groups.

Shared UID examples

We created USER1 with UID(7). Then we tried to define USER2 with the same UID(7), but received the following error message:

```
IRR52174I Incorrect UID 7.  This value is already in use by USER1.
```

You will get the following error message if you try to specify more than one user in an ADDUSER command request:

```
IRR52185I The same UID cannot be assigned to more than one user.
```

Existing shared UIDs

The use of this new functionality does not affect pre-existing shared UIDs; they will remain as shared once you install the new support. If you want to eliminate sharing of the same UID, you must clean them up separately. The release provides a new IRRICE report to find the shared Allowing shared UIDs

Even if the SHARED.IDS profile is defined, you may still require some UIDs to be shared and others not to be shared. For example, you may require multiple superusers with a UID(0). It is possible to do this using the new SHARED keyword in the OMVS segment of the **ADDUSER**, **ALTUSER**, **ADDGROUP**, and **ALTGROUP** commands.

To allow an administrator to assign a non-unique UID or GID using the SHARED keyword, you must grant that administrator at least READ access to SHARED.IDS profile, as follows:

```
PERMIT SHARED.IDS CLASS(UNIXPRIV) ID(ADMIN) ACCESS(READ)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Once user ID ADMIN has at least READ access to the SHARED.IDS profile, ADMIN will be able to assign the same UID or GID to multiple users, using the SHARED KEYWORD, as follows:

```
ADDUSER (USERA USERB) OMVS(UID(7) SHARED)
```

Note: To specify the SHARED operand, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS profile in the UNIXPRIV class.

9.4.4 Automatic UID/GID assignment

UIDs and GIDs can be assigned automatically by RACF to new users, making it easier to manage the process of assigning UIDs and GIDs to users. (Previously, this was a manual process and guaranteed the uniqueness of the UID and GID for every user.)

Now, by using a new AUTOUID keyword with the **ADDUSER** and **ATLUSER** commands, an unused UID will be assigned to the new or modified user. Using the AUTOGID keyword on **ADDGROUP** and **ALTGROUP** commands, a GID will be automatically assigned to the new or modified group.

The use of automatic UID/GID requires the following:

- AIM stage 2 or 3.

  Otherwise, the automatic assignment attempt fails and an IRR52182I message is issued:

  ```
  IRR52182I Automatic UID assignment requires application identity mapping to be
  implemented
  ```

- A SHARED.IDS profile must be defined.

Otherwise, the attempt fails and an IRR52183I message will be issued:

```
IRR52183I Use of automatic UID assignment requires SHARED.IDS to be implemented
```

The SHARED.IDS must be defined as follows:

```
RDEFINE UNIXPRIV SHARED.IDS UACC(NONE)
```

This command is explained in 9.4.3, “Shared UID prevention” on page 182.

- The BPX.NEXT.USER facility class profile must be defined and RACLISTed.

  Otherwise, the attempt will fail and an IRR52179I message will be issued:

  ```
  IRR52179I The BPX.NEXT.USER profile must be defined before you can use automatic UID
  assignment.
  ```

  The definition of the BPX.NEXT.USER FACILITY class profile has the following syntax:

  ```
  RDEFINE FACILITY BPX.NEXT.USER APPLDATA(UID/GID)
  ```

  Where APPLDATA consists of two qualifiers separated by a forward slash(/). The qualifier on the left of the slash character specifies the starting UID value or range of UID values. The qualifier on the right of the slash character specifies the starting GID value or range of GID values. Qualifiers can be null or specified as NOAUTO to prevent automatic assignment of UIDs or GIDs.

  The starting value is the value RACF attempts to use in ID assignment, after determining that the ID is not in use. If it is in use, the value is incremented until an appropriate value is found.

  The maximum value valid in the APPLDATA specification is 2,147,483,647. If this value is reached or a candidate UID/GID value has been exhausted for the specified range, subsequent automatic ID assignment attempts fail and message IRR52181I is issued.

  Note: Keep in mind that APPLDATA is verified at the time of *use*, not when it is defined. If a syntax error is encountered when the auto assignment is used, an IRR52187I message is issued and the attempt fails.

Automatic assignment example

In the following example, we have defined the APPLDATA for a range of values from 5 to 70000 for UIDs, and from 3 to 30000 for GIDs. USERA and USERB are added using the automatic assignment of UID. The range of automatic UID assignment starts with 5, so USERA is assigned to UID(5), which was free. UID(6) and UID(7) were already assigned before we started our examples the first following free UID is 8. USERB is assigned to UID(8).

```
RDEFINE FACILITY BPX.NEXT.USER APPLDATA('5-70000/3-30000')

ADDUSER USERA OMVS(AUTOUID)
IRR52177I User USERA was assigned an OMVS UID value of 5.

ADDUSER USERB OMVS(AUTOUID)
IRR52177I User USERB was assigned an OMVS UID value of 8.
```

RACF extracts the APPLDATA from the BPX.NEXT.USER and parses out the starting value. It checks if it is already in use and if so, the value is incremented and checked again until an unused value is found. Once a free value is found, it assigns the value to the user or group and replaces the APPLDATA with the new starting value, which is the next potential value or the end of the range.

In our example, that means that if UID(6) becomes free after UID(7) is assigned to USERB, RACF will start checking from UID(8) in the next assignment, so it will not assign UID(6). However, you can change the APPLDATA and modify the starting value. The APPLDATA can be changed using the following command:

```
RALTER FACILITY BPX.NEXT.USER APPLDATA('2000/500')
```

Note: Automatic assignment of UIDs and GIDs fails if you specify a list of users to be defined with the same name, or if you specify the SHARED keyword. Also, AUTOUID or AUTOGID is ignored if UID or GID is also specified.

APPLDATA examples

Following are examples of correct and incorrect APPLDATA specifications:

Good data
1/0
1-50000/1-20000
NOAUTO/100000
/100000
10000-20000/NOAUTO
10000-20000/

Bad data
123B
/
2147483648 /* higher than max UID value */
555/1000-900

If you have an incorrect specification and attempt to use AUTOUID on an **ADDUSER** command, the following message is issued:

```
IRR52187I Incorrect APPLDATA syntax for the BPX.NEXT.USER profile.
```

Automatic assignment with RACF panel

You may use the RACF panels to define the OMVS segment. Figure 9-13 on page 186 indicates how to use the automatic assignment by using the AUTOUID field.

Archived

```
                          RACF - CHANGE USER JANE
                          OMVS PARAMETERS
COMMAND ===>

 Delete ALL OMVS information      (NOOMVS)  ___    Enter YES to DELETE

    -- OR --

 Choose to CHANGE or DELETE, then press ENTER.
                                                                 More:     +
 Specify new User Identifier         (UID)  __________  0 - 2147483647
 Allow shared use of this UID     (SHARED)  _           Enter any character
    -- or --
 Assign a unique UID             (AUTOUID)  _           Enter any character
    -- or --
 Delete User Identifier            (NOUID)  _           Enter any character

 Change Initial Path Name           (HOME)  _           Enter any character
 Delete Initial Path Name         (NOHOME)  _           Enter any character

 Change Program Path Name        (PROGRAM)  _           Enter any character
 Delete Program Path Name      (NOPROGRAM)  _           Enter any character

 Specify CPU Time             (CPUTIMEMAX)  __________  7 - 2147483647
 Delete  CPU Time           (NOCPUTIMEMAX)  _           Enter any character

 Specify Address Space Size    (ASSIZEMAX)  __________  10485760 -
                                                           2147483647
 Delete  Address Space Size  (NOASSIZEMAX)  _           Enter any character
```

Figure 9-13 RACF panel to set OMVS parameters for automatic assignment

Automatic assignment in an RRSF configuration

In an RRSF configuration (see Figure 9-14 on page 187) you may wish to avoid UID and GID duplications. This can be done by using non-overlapping APPLDATA ranges.

You may also wish to make RACF automatically suppress propagation of internal updates. This can be done by specifying the ONLYAT keyword to manage the BPX.NEXT.USER profile, as follows:

```
RDEFINE BPX.NEXT.USER APPLDATA('5000-10000/5000-10000') ONLYAT(NODEA.MYID)
RDEFINE BPX.NEXT.USER APPLDATA('10001-20000/10001-20000') ONLYAT(NODEB.MYID)
```

For more information about automatic assignment in a RRSF configuration, refer to *z/OS V1R4.0 Security Server RACF Security Administrator's Guide*, SA22-7683.

Archived

Figure 9-14 Automatic assignment in an RRSF configuration

9.4.5 Group ownership option

By default, the system assigns the UID and GID of a file when it is created:

- The owning UID is taken from the effective UID of the owning process.
- The owning GID is taken from the parent directory.

Figure 9-15 Group ownership example without UNIXPRIV profile

New group ownership option

The new group ownership option introduced in z/OS V1R4 allows you to specify that the group owner of a new file is to come from the effective GID of the creating process.

To enable this new option, you must perform the following tasks:

- Define a FILE.GROUPOWNER.SETGID profile in the UNIXPRIV class and do a refresh of the UNIXPRIV class:

```
RDEFINE UNIXPRIV FILE.GROUPOWNER.SETGID
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

- Turn off the set-gid bit for the directory (it is the default) if it is not already off.

```
chmod g+s /u/jane/MyDir
```

Once the new group ownership option is enabled, the new assignment is shown in Figure 9-16.

Figure 9-16 New group ownership option with UNIXPRIV profile

Note: After a new file system is mounted, you must turn on the set-gid bit of its root directory if you want new objects within the file system to have their group owner set to that of the parent directory.

Archived

10

zFS file system enhancements

In this chapter we discuss the enhancements introduced in z/OS V1R3 and z/OS V1R4 for Distributed File Services zSeries File System. The following enhancements are described from z/OS V1R3:

- zFS file system support UNIX ACLs

The following enhancements are described from z/OS V1R4:

- Dynamic configuration
- Dynamic aggregate extension
- Grow option
- Duplicate file system names in different aggregates
- System symbols in IOEFSPRM
- Metadata backing cache
- Log file cache

© Copyright IBM Corp. 2003

10.1 zFS file systems

The z/OS Distributed File Service (DFS) zSeries File System (zFS) is a z/OS UNIX file system that can be used in addition to the Hierarchical File System (HFS). zFS provides significant performance gains in accessing files approaching 8K in size that are frequently accessed and updated. The access performance of smaller files is equivalent to that of HFS.

Note the following:

- zFS provides reduced exposure to loss of updates by writing data blocks asynchronously and not waiting for a sync interval.
- zFS is a logging file system. It logs metadata updates. If a system failure occurs, zFS replays the log when it comes back up to ensure that the file system is consistent.
- zFS is a Physical File System (PFS) that is started by UNIX System Services (USS) during IPL. A physical file system is the part of the operating system that handles the actual storage and manipulation of data on a storage medium.

Figure 10-1 ZFS physical file system (PFS)

10.1.1 zFS supports z/OS UNIX ACLs

In order to provide better granularity of access control for z/OS UNIX files and directories, access control lists have been introduced with z/OS V1R3. You can use access control lists (ACLs) to control access to files and directories by individual UIDs and GIDs. This provides the means to allow specific users and groups dedicated and different types of access.

To manage an ACL for a file, you must have one of the following security access controls:

- Be the file owner
- Have superuser authority (UID=0)
- Have READ access to SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class

Beginning with z/OS V1R3, ACLs are supported by the HFS and zFS file systems. You must also know whether your security product supports ACLs and what rules are used when determining file access.

ACL support works similar to the way access to MVS data sets is permitted, although the implementation is different. The ACL is a part of the File Security Packet (FSP), which is maintained by the Physical File System (PFS).

Note: For more information on ACLs, see "Access control list (ACL) support for V1R3" on page 122 and also refer to *z/OS UNIX System Services Planning*, GA22-7800.

10.2 Dynamic configuration

zFS has an IOEFSPRM configuration file which specifies the processing options for the ZFS PFS, as well as some definitions for multi-file aggregates. Prior to z/OS V1R4, in order to change any configuration file parameters, you had to do the following:

1. Modify the IOEFSPRM file.
2. Shut down and restart the ZFS PFS.

Doing this causes unmounts and potential moves of zFS file systems in a sysplex environment, which could be disruptive to applications and is administratively involved.

zFS provides utility programs and z/OS UNIX commands to assist in the customization of the aggregates and file systems. These utilities and commands are to be used by system administrators. For more detailed information on the **zfsadm** command options and usage, see *z/OS Distributed File Service zSeries File System Administration*, SC24-5989.

10.2.1 New zfsadm commands

Two new **zfsadm** commands have been added with z/OS V1R4 in order to change configuration values dynamically without a shutdown, restart the ZFS PFS, and display the current value of the zFS configuration options. Table 10-1 on page 192 shows the zfsadm command with its subcommands.

The issuer of the zfsadm command must have READ authority to the data set that contains the IOEFSPRM file and must be root or have READ authority to the SUPERUSER.FILESYS.PFSCTL profile in the UNIXPRIV class.

To issue the commands shown in Table 10-1, you must have RACF authorization, as follows:

IOEFSPRM The command issuer must have access to a RACF data set profile to the IOEFSPRM data set.

SU mode The command issuer must have superuser authority.

Table 10-1 The zfsadm command and subcommands

Command/subcommand	Command description	IOEFSPRM	SU mode
zfsadm aggrinfo	Obtain information on attached aggregate	Read	No
zfsadm apropos	Display first line of help entry	Read	No
zfsadm attach	Attach an aggregate	Read	Yes
zfsadm clone	Clone a file system	Read	Yes
zfsadm clonesys	Clone multiple file systems	Read	Yes
`zfsadm config`	Modify current configuration options	Read	Yes
`zfsadm configquery`	Display current configuration options	Read	Yes
zfsadm create	Create a file system	Read	Yes
zfsadm define	Define a VSAM linear data set	Read	No
zfsadm delete	Delete a file system	Read	Yes
zfsadm detach	Detach an aggregate	Read	Yes
zfsadm format	Format a VSAM LDS as an aggregate	Alter	Yes
zfsadm grow	Grow an aggregate	Read	Yes
zfsadm help	Get help on commands	Read	No
zfsadm lsaggr	List all currently attached aggregates	Read	No
zfsadm lsfs	List all file systems on a aggregate or all	Read	No
zfsadm lsquota	Show quotas for file systems & aggregates	Read	No
zfsadm quiesce	Quiesce an aggregate and all file systems	Read	Yes
zfsadm rename	Rename a file system	Read	Yes
zfsadm setquota	Set the quota for a file system	Read	Yes
zfsadm unquiesce	Make aggregate and file systems available	Read	Yes

zfsadm config command

The **`zfsadm config`** command changes the value of zFS configuration options in memory that were specified in the IOEFSPRM file (or defaulted).

The format of the zfsadm config command is shown in Figure 10-2 on page 193.

```
zfsadm config [-admin_threads number]
          [-user_cache_size number]
          [-meta_cache_size number]
          [-log_cache_size number]
          [-sync_interval number]
          [-vnode_cache_size number]
          [-nbs {on|off}][-fsfull threshold,increment]
          [-aggrfull threshold,increment]
          [-trace_dsnPDSE_dataset_name]
          [-tran_cache_size number]
          [-msg_output_dsn Seq_dataset_name]
          [-user_cache_readahead {on|off}]
          [-metaback_cache_size number]
          [-fsgrow increment,times]
          [-aggrgrow {on|off}]
          [-allow_dup_fs {on|off}]
          [-level]
          [-help]
```

Figure 10-2 zfsadm config command parameters

The issuer must have READ authority to the data set that contains the IOEFSPRM file and must be a superuser or have READ authority to the SUPERUSER.FILESYS.PFSCTL profile in the UNIXPRIV class.

The example shown in Figure 10-3 changes the value of the sync_interval to 60 seconds.

```
@ SC65:/>zfsadm config -sync_interval 60
IOEZ00300I Successfully set -sync_interval to 60
```

Figure 10-3 Command to change the value of the sync_interval

Note: If you want the configuration specification to be permanent, you need to update the IOEFSPRM file.

zfsadm configquery command

The **zfsadm configquery** command displays the current value of zFS configuration options retrieved from the zFS address space memory, rather than from the IOEFSPRM file.

The format of the zfsadm configquery command is shown in Figure 10-4 on page 194.

```
zfsadm configquery [-adm_threads]
          [-aggrfull]
          [-aggrgrow]
          [-all]
          [-allow_dup_fs]
          [-auto_attach]
          [-cmd_trace]
          [-code_page]
          [-debug_dsn]
          [-fsfull]
          [-fsgrow]
          [-log_cache_size]
          [-meta_cache_size]
          [-metaback_cache_size]
          [-msg_input_dsn]
          [-msg_output_dsn]
          [-nbs]
          [-storage_details]
          [-sync_interval]
          [-trace_dsn]
          [-trace_table_size]
          [-tran_cache_size]
          [-user_cache_readahead]
          [-user_cache_size]
          [-usercancel]
          [-vnode_cache_size]
          [-level]
          [-help]
```

Figure 10-4 zfsadm configquery command parameters

The example in Figure 10-5 displays all current values of zFS configuration options.

```
@ SC65:/>zfsadm configquery -all
IOEZ00317I The value for config option -adm_threads is 10.
IOEZ00317I The value for config option -aggrfull is <no value>.
IOEZ00317I The value for config option -aggrgrow is OFF.
IOEZ00317I The value for config option -allow_dup_fs is OFF.
IOEZ00317I The value for config option -auto_attach is ON.
IOEZ00317I The value for config option -cmd_trace is OFF.
IOEZ00317I The value for config option -debug_dsn is <no value>.
IOEZ00317I The value for config option -fsfull is <no value>.
IOEZ00317I The value for config option -fsgrow is <no value>.
IOEZ00317I The value for config option -log_cache_size is 64M.
IOEZ00317I The value for config option -meta_cache_size is 32M.
IOEZ00317I The value for config option -metaback_cache_size is <no value>.
IOEZ00317I The value for config option -msg_input_dsn is <no value>.
IOEZ00317I The value for config option -msg_output_dsn is <no value>.
IOEZ00317I The value for config option -nbs is OFF.
IOEZ00317I The value for config option -sync_interval is 30.
IOEZ00317I The value for config option -trace_dsn is <no value>.
IOEZ00317I The value for config option -trace_table_size is 256K.
IOEZ00317I The value for config option -tran_cache_size is 2000.
IOEZ00317I The value for config option -user_cache_readahead is ON.
IOEZ00317I The value for config option -user_cache_size is 256M.
IOEZ00317I The value for config option -vnode_cache_size is 8192.
```

Figure 10-5 Display of all the IOEFSPRM values

10.3 Dynamic aggregate extension

Before this change in z/OS V1R4, if aggregates become full they could only be grown by using the **zfsadm grow** command. You needed to specify a larger size or specify zero for the size to get a secondary allocation size extension.

z/OS V1R4 introduces the possibility to dynamically grow an aggregate if it becomes full. The aggregate is extended automatically when an operation cannot complete because the aggregate is full. If the extension is successful, the operation will be redriven transparent to the application.

> **Important:** To dynamically grow an aggregate when it becomes full, the VSAM LDS must have a secondary allocation and have space on the volume(s).

10.3.1 Implementing dynamic aggregate extension

Dynamic aggregate extension can be enabled in the following ways:

- In the IOEFSPRM configuration file, you can dynamically extend an aggregate when it becomes full by specifying one of the following options:
  - You can specify a new option: aggrgrow=on | off. The default value is off.

    > **Note:** The option specified here is the default if none of the following ways of specifying the aggrgrow | noaggrgrow options are used.

  - You can specify either aggrgrow | noaggrgrow as a suboption on the define_aggr option for a multi-file system aggregate, as shown in the following definition:

    ```
    define_aggr R/W attach aggrgrow cluster(OMVS.TEST.ZFS)
    ```

- Using the **mount** command, in the PARM keyword you can specify either aggrgrow or noaggrgrow, as shown in the following example

  ```
  mount filesystem('omvs.test.zfs') mountpoint('/tmp/test') type(zfs) mode
  (rdwr) parm('aggrgrow')
  ```

  > **Note:** This aggrgrow | noaggrgrow option can only be used with compatibility mode aggregates.

- Using the **zfsadm attach** command for attaching a multi-file system aggregate, you can specify either the -aggrgrow or -noaggrgrow option, as shown in the following example

  ```
  zfsadm attach -aggregate OMVS.TEST.ZFS -aggrgrow
  ```

- Using the **zfsadm config** command, you can dynamically change the configuration file option, aggrgrow on | off. This becomes the new default if no other option specification is in use.

Aggregate extension processing

When an aggregate fills and dynamic aggregate extension has been specified using one of the options, the aggregate is extended using secondary allocation extensions, the extension(s) taken are formatted, and it becomes available transparently to the application. The messages that are issued indicating the process are shown in Figure 10-6 on page 196.

```
IOEZ00312I Dynamic growth of aggregate OMVS.TEST.ZFS in progress, (by user AYVIVAR).
IOEZ00329I Attempting to extend OMVS.TEST.ZFS by a secondary extent.
IOEZ00324I Formatting to 8K block number 360 for secondary extents of OMVS.TEST.ZFS
IOEZ00309I Aggregate OMVS.TEST.ZFS successfully dynamically grown (by user AYVIVAR).
```

Figure 10-6 Messages issued when dynamically growing an aggregate

10.3.2 Dynamic file system quota increase

The maximum size of a file system is known as its *quota*. This is a logical number that is compared against each time additional blocks are allocated to the file system. A quota can be smaller than, equal to, or larger than the space available in the aggregate. When the quota is reached, the file system indicates that it is full.

z/OS V1R4 introduces several ways to dynamically increase the file system quota if the file system becomes full. Dynamic file system quota increase can be specified in the following ways:

- A new option in the IOEFSPRM configuration file, fsgrow=(increment,times), specifies whether file systems in a multi-file aggregate can have their quota dynamically extended. The value that is specified in this file becomes the default if no other way to specify this option is made, where:

  increment The number of k-bytes to increase the quota. The maximum value that can be specified is 2147483647.

  times The number of times to extend the quota

- Using the **mount** command, in the PARM keyword you can specify the file system quota extension of 500 KB for a maximum of four times, as fsgrow(increment,times), as shown in the following example:

  ```
  mount filesystem(zfstest) mountpoint('/tmp/zfstest') type(zfs) mode(rdwr)
  parm('fsgrow(500,4)') noautomove
  ```

- Using the **zfsadm config** command, you can dynamically change the configuration file option by specifying, -fsgrow increment times, as shown in the following example

  ```
  zfsadm config -fsgrow 500,4
  ```

  For example, fsgrow(500,4) means grow the quota by 500 K bytes up to 4 times.

Displaying the quota

Figure 10-7 shows the file system quota to be 1159, which was defined for a compatibility mode aggregate.

```
$> zfsadm lsquota -filesystem OMVS.TEST.ZFS
Filesys Name          Quota    Used  Percent Used  Aggregate
OMVS.TEST.ZFS         1159       9     0    11 = 146/1296 (zFS)
```

Figure 10-7 Display quota information about file systems and aggregates

Compatibility mode aggregates

For a compatibility mode aggregate, only the aggrgrow specification is used to extend an aggregate size. The fsgrow option is ignored if it is specified. The quota increases by the size of the extension. For compatibility mode aggregates, the file system quota is dynamically increased based on a new aggregate size once it becomes full, as shown in Figure 10-8 on page 197.

```
Before dynamic extension:
Filesys Name          Quota    Used  Percent Used  Aggregate
OMVS.TEST.ZFS         1159       9     0    11 = 146/1296 (zFS)
After dynamic extension:
Filesys Name          Quota    Used  Percent Used  Aggregate
OMVS.TEST.ZFS         1807    1577    87    88 = 1714/1944 (zFS)
```

Figure 10-8 Compatibility mode aggregate dynamic extension

Multi-file mode aggregates

For a multi-file system aggregate, there may be multiple file systems in the aggregate. Therefore, if a file system becomes full, equal to its quota, an fsgrow option must be in place for the file system to then use the additional physical space that is available in the aggregate, following the dynamic increase of the individual quota for the file system.

10.3.3 Displaying dynamic aggregate and quota extensions

Values set for dynamic aggregate or quota extensions can be displayed if you want to know which values have been assigned in each case.

Global parameters defined in IOEFSPRM or modified later by the **zfsadm config** command can be displayed by using the **zfsadm configquery** command. As shown in Figure 10-9, you can obtain the current values for both the aggrgrow and fsgrow parameters.

```
@ SC65:/>zfsadm configquery -aggrgrow -fsgrow
IOEZ00317I The value for config option -aggrgrow is ON.
IOEZ00317I The value for config option -fsgrow is (500,4).
```

Figure 10-9 Display the current aggrgrow and fsgrow options

Specific values that are set by using the **mount** command for a zFS file system can be displayed by using the **df -v** command, as shown in Figure 10-10. Alternatively, you can use the z/OS **d omvs,f** command, as shown in Figure 10-11 on page 198.

```
@ SC65:/>df -v /tmp/zfs3
Mounted on     Filesystem                Avail/Total    Files      Status
/SC65/tmp/zfs3 (ZFS3)                    2310/5000      4294967291 Available
ZFS, Read/Write, Device:178, ACLS=Y
fsgrow(500,4)
File System Owner : SC65        Automove=N      Client=N
Filetag : T=off   codeset=0
Aggregate Name : OMVS.TEST7.ZFS

@ SC65:/>df -v /tmp/test8
Mounted on     Filesystem                Avail/Total    Files      Status
/SC65/tmp/test8 (OMVS.TEST8.ZFS)          460/3614       4294967291 Available
ZFS, Read/Write, Device:170, ACLS=Y
aggrgrow
File System Owner : SC65        Automove=N      Client=N
Filetag : T=off   codeset=0
Aggregate Name : OMVS.TEST8.ZFS
```

Figure 10-10 Display the current values of the aggrgrow and fsgrow options

```
d omvs,f
..............
ZFS           170 ACTIVE                      RDWR
  NAME=OMVS.TESTC.ZFS
  PATH=/SC65/tmp/testc
  AGGREGATE NAME=OMVS.TESTC.ZFS
  MOUNT PARM=aggrgrow
  OWNER=SC65     AUTOMOVE=N CLIENT=N
ZFS           178 ACTIVE                      RDWR
  NAME=ZFSTEST
  PATH=/SC65/tmp/zfstest
  AGGREGATE NAME=OMVS.TEST.ZFS
  MOUNT PARM=fsgrow(500,4)
  OWNER=SC65     AUTOMOVE=N CLIENT=N
................
```

Figure 10-11 Display options using the d omvs,f z/OS command

The **f bpxoinit,filesys=display,filesystem=filesystem** command also shows information about the aggrgrow and fsgrow parameters, as shown in Figure 10-12.

```
f bpxoinit,filesys=display,filesystem=zfsa
BPXM027I COMMAND ACCEPTED.
BPXF035I 2002/05/24 14.26.12 MODIFY BPXOINIT,FILESYS=DISPLAY
--------------NAME-------------------------        DEVICE  MODE
ZFSA                                                  198  RDWR
  AGGREGATE NAME=OMVS.TESTA.ZFS
  PATH=/SC65/tmp/testa
  PARM=fsgrow(500,4)
  STATUS=ACTIVE                LOCAL STATUS=ACTIVE
  OWNER=SC65        RECOVERY OWNER=SC65     AUTOMOVE=N PFSMOVE=Y
  TYPENAME=ZFS      MOUNTPOINT DEVICE=      72
  MOUNTPOINT FILESYSTEM=/SC65/TMP
  ENTRY FLAGS=90060000  FLAGS=40000010  LFSFLAGS=00000000
  LOCAL FLAGS=40000010  LOCAL LFSFLAGS=20000000
BPXF040I MODIFY BPXOINIT,FILESYS PROCESSING IS COMPLETE.
```

Figure 10-12 Display aggrgrow and fsgrow options using the f bpxoinit command

10.4 New -grow option

The VSAM linear data set must be formatted to be used as a zFS aggregate. There are two options available to format an aggregate:

- IOEAGFMT format utility
- **zfsadm** command

Both options use the same parameters, as follows:

Format parameters

```
zfsadm format -aggregate name [-initialempty blocks] [-size blocks] [-logsize blocks]
[-overwrite] [-compat] [-owner {uid | name}] [-group {group_id | name}]
[-perms decimal | octal | hex_number] [-level] [-help]
```

-size option description

-size After you have allocated the space for an aggregate, the default size is the number of 8 K blocks that fits into the primary allocation. You can specify a -size option giving the number of 8 K blocks for the aggregate.

If you specify a number that is less than (or equal to) the number of blocks that fits into the primary allocation, the primary allocation size is used. If you specify a number that is larger than the number of 8K blocks that fits into the primary allocation, the VSAM LDS is extended to the size specified. This occurs during its initial formatting.

10.4.1 Formatting aggregates with -grow

When an aggregate is initially formatted using the IOEAGFMT format utility or the **`zfsadm format`** command, the formatting takes place as follows:

- The default size formatted is the number of blocks that will fit in the primary allocation of the VSAM LDS
- Using the **-size** parameter, if the number of blocks to be formatted is less than the default, it is rounded up to the default.
- If a number greater than the default is specified, a single extend of the VSAM LDS is attempted after the primary allocation is formatted.

> **Note:** The size of the secondary extension is proportional to the number of blocks to be formatted and rounded up to the next multiple of the secondary size specified in the VSAM LDS definition.

-grow option

Since the **-**size parameter specified is only able to be allocated in the primary allocation and one extension, if you wanted to format a three-volume aggregate with the IOEAGFMT utility, this is not possible because it requires a primary and at least two extensions. Therefore, a new -grow option is provided with z/OS V1R4 for the IOEAGFMT utility and the **`zfsadm format`** command to allow specification of the increment that can be used for extension of the aggregate when **-**size is larger than the primary allocation, as shown in Figure 10-13. This allows the extension by the -grow amount until -size is satisfied.

-grow Specifies the number of 8K blocks that zFS uses as the increment for an extension when the -size option specifies a size greater than the primary allocation.

Figure 10-13 New -grow option

-grow example

To illustrate the before and after using the -grow option, the following example has a VSAM LDS defined with 2 cylinders of primary space and 1 cylinder of secondary space, as shown in Figure 10-14 on page 200.

```
//AYVIVAR2 JOB CLASS=J,MSGCLASS=A,NOTIFY=AYVIVAR
/*JOBPARM S=SC65
//P010 EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER(NAME(OMVS.TESTA.ZFS) VOLUMES(SBOX43) -
  LINEAR CYL(2,1) SHAREOPTIONS(2))
//
```

Figure 10-14 Allocating a VSAM LDS for a zFS aggregate

Format the aggregate

The VSAM LDS is formatted as a compatibility mode aggregate, shown in Figure 10-15 on page 200, using a -size value of 276 8K blocks.

```
//AYVIVAR2 JOB CLASS=J,MSGCLASS=V,NOTIFY=AYVIVAR
/*JOBPARM S=SC65
//FORMAT EXEC PGM=IOEAGFMT,REGION=0M,
//       PARM=('-aggregate OMVS.TESTA.ZFS -size 276 -compat')
//SYSPRINT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
//*
```

Figure 10-15 Formatting the aggregate

Listcat output of aggregate

If we look at the listcat output, shown in Figure 10-16, we can see that the file has two extents; the first one corresponds to the primary space specified in the define process (30 tracks), and the second one has 30 tracks.

```
ALLOCATION
  SPACE-TYPE------CYLINDER     HI-A-RBA---------2949120
  SPACE-PRI--------------2     HI-U-RBA---------2949120
  SPACE-SEC--------------1
VOLUME
  VOLSER------------SBOX44     PHYREC-SIZE---------4096     HI-A-RBA---------2949120     EXTENT-NUMBER----------2
  DEVTYPE------X'3010200F'     PHYRECS/TRK-----------12     HI-U-RBA---------2949120     EXTENT-TYPE--------X'40'
S  SYSTEM SERVICES                                          TIME: 19:31:08
  VOLFLAG------------PRIME     TRACKS/CA-------------15
  EXTENTS:
  LOW-CCHH-----X'02080000'     LOW-RBA----------------0 TRACKS----------------30
  HIGH-CCHH----X'0209000E'     HIGH-RBA---------1474559
  LOW-CCHH-----X'020A0000'     LOW-RBA----------1474560 TRACKS----------------30
  HIGH-CCHH----X'020B000E'     HIGH-RBA---------2949119
```

Figure 10-16 Listcat of the VSAM LDS aggregate

Using the -grow option

With the new -grow parameter, you are allowed to specify the increment that will be used for an extension size larger than the primary allocation. That is, after the primary space is allocated, multiple extensions of the amount specified by the -grow parameter rounded up to a multiple of the secondary space defined will be attempted until the total number of blocks specified by the -size parameter are satisfied, as shown in Figure 10-13 on page 199.

Replacing the example shown in Figure 10-15, we used the -grow parameter on the format process, as shown in Figure 10-17 on page 201.

```
//AYVIVAR2 JOB CLASS=J,MSGCLASS=V,NOTIFY=AYVIVAR
/*JOBPARM S=SC65
//FORMAT EXEC PGM=IOEAGFMT,REGION=0M,
//       PARM=('-aggregate OMVS.TESTA.ZFS -size 276 -grow 90  -compat')
//SYSPRINT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
```

Figure 10-17 Format of VSAM LDS aggregate using -grow

Listcat output using -grow

The listcat output, shown in Figure 10-18,shows that now the VSAM LDS has three extensions; the first one corresponding to the primary space specified, and the next ones by the -grow amount.

```
ALLOCATION
  SPACE-TYPE------CYLINDER     HI-A-RBA---------2949120
  SPACE-PRI--------------2     HI-U-RBA---------2949120
  SPACE-SEC--------------1
VOLUME
  VOLSER------------SBOX15     PHYREC-SIZE---------4096     HI-A-RBA---------2949120     EXTENT-NUMBER----------3
  DEVTYPE------X'3010200F'     PHYRECS/TRK-----------12     HI-U-RBA---------2949120     EXTENT-TYPE--------X'40'
S  SYSTEM SERVICES                                          TIME: 19:42:56
  VOLFLAG------------PRIME     TRACKS/CA-------------15
  EXTENTS:
  LOW-CCHH-----X'003F0000'     LOW-RBA----------------0     TRACKS--------------30
  HIGH-CCHH----X'0040000E'     HIGH-RBA---------1474559
  LOW-CCHH-----X'00410000'     LOW-RBA----------1474560     TRACKS--------------15
  HIGH-CCHH----X'0041000E'     HIGH-RBA---------2211839
  LOW-CCHH-----X'00420000'     LOW-RBA----------2211840     TRACKS--------------15
  HIGH-CCHH----X'0042000E'     HIGH-RBA---------2949119
```

Figure 10-18 Listcat output using the -grow option

10.5 Duplicate file system names

Prior to z/OS V1R4, file system names are required to be unique among all attached aggregates on a system. Although it is possible to create the same file system name on two different aggregates, you need to ensure that they are not attached at the same time—if a second aggregate is attached, an error message is issued and the duplicate file system becomes unavailable.

With z/OS V1R4, a new IOEFSPRM configuration file option allows duplicate file system names to be in different aggregates. The new option is:

```
allow_duplicate_filesystems=on
```

Commands that specify zFS file systems can now specify an aggregate name to qualify it, if more than one file system name exists. However, if a file system name is ambiguous (meaning that the name is a duplicate and an aggregate name is not specified), then the command fails.

If you specify **allow_duplicate_filesystems=off** (which is the default), and try to create a duplicate file system name, the request will be denied and the following error message will be issued:

```
IOEZ00097E File system ZFSA already exists.
```

In previous versions (before the new option in z/OS V1R4), it is possible to create the same file system name on two different aggregates by not having them attached at the same time. When the second aggregate is attached, the attach is successful but an error message is issued for the duplicate file system name and it becomes unavailable, as follows:

```
IOEZ00314E The file system name ZFSA is not unique. Its aggregate name must also be
specified.
```

10.6 System symbols in the IOEFSPRM file

System symbols can now be specified for data set names in the IOEFSPRM configuration file, to make it easier to share a single IOEFSPRM file in a sysplex.

Figure 10-19 illustrates uses of system symbols in IOEFSPRM configuration options.

```
trace_dsn=OMVS.&SYSNAME..ZFS.TRACEOUT
define_aggr R/W attach aggrgrow cluster(OMVS.&SYSNAME..AGGR1)
```

Figure 10-19 IOEFSPRM configuration file using system symbols

10.7 Metadata backing cache and log file cache

The performance of zFS can be influenced by controlling the size of the caches used to hold file system and log data. There are three caches that can be monitored and controlled to reduce I/O rates, as follows:

- User file cache

  This cache is used for all user files and performs I/O for all user files greater than 7 KB.
- Metadata cache

  User files that are smaller than 7 KB have the I/O from this cache.
- Log file cache

  This cache is used to write file record transactions that describe changes to the file system.

Figure 10-20 on page 203 shows the cache prior to z/OS V1R4.

Archived

Figure 10-20 Default cache sizes

10.7.1 Metadata cache storage

The metadata cache is used to contain all file system metadata, which includes the following:

- All directory contents
- File status information, which includes atime, mtime, size, permission bits, and so on
- File system structures
- Caching of data for files smaller than 7K

New metadata backing cache

An optional metadata backing cache that contains an extension to the meta cache can be specified. This new backing cache resides in a data space and is used as a “paging” area for metadata. Therefore, it allows a larger meta cache for workloads that need large amounts of metadata.

Note: This optional cache is only needed if the metadata cache is constrained.

This option can be enabled by specifying a metaback_cache_size option in the IOEFSPRM configuration file. A value between 1 MB and 2048 MB is allowed. You can specify values by using a “K” or an “M” following the value (indicating kilobytes or megabytes, respectively). It is also possible to specify a fixed option that indicates the pages are permanently fixed for performance, as follows:

```
metaback_cache_size=64M,fixed
```

10.7.2 Log file cache

Prior to z/OS V1R4, the log file cache is shared among all aggregates and is stored in the primary address space. With z/OS V1R4, the log file cache is moved to a dataspace, as shown in Figure 10-21 on page 204. This cache defaults to 64 MB.

For each aggregate that is attached, the log file cache is grown dynamically by adding one 8K buffer. Each aggregate always has one 8K buffer to record its most recent changes to file system metadata.

Figure 10-21 Log file cache grows dynamically

Log file cache in a data space

The log file cache is now moved into a dataspace, which frees up space in the zFS address space and allows for space for the other caches.

Archived

11

Security Server RACF enhancements

This chapter describes the changes to the Security Server RACF and contains the following topics:

- Security Server components
- Access control lists (ACLs)
- IBM Policy Director Authorization Services
- Enterprise Identity Mapping Services (EIM)
- z/OS UNIX security management usability enhancements
- Program control and program access to data sets (PADS)

Archived

© Copyright IBM Corp. 2003

11.1 Security Server components

Security Server consists of following components:

DCE Security Server	Last changed in OS/390 V2.9.
Firewall Technologies	Changed in z/OS V1R4.
LDAP Server	Changed in z/OS V1R4.
Network Authentication Service	Changed in z/OS V1R4.
Open Cryptographic Enhanced Plug-ins (OCEP)	Last changed in OS/390 V2R10.
RACF	Changed in z/OS V1R3 and in z/OS V1R4.
Public Key Infrastructure (PKI) Services	New in z/OS V1R3 and changed in z/OS V1R4.

In OS/390 V2R9, the word “SecureWay” was added to the beginning of this feature’s name. As of z/OS V1R3, the word “SecureWay” was dropped.

In this chapter, we discuss the enhancements of the RACF component. We describe the security enhancements of the other components in subsequent chapters.

11.2 RACF enhancements in z/OS V1R3

In z/OS V1R3, the following new RACF functions are introduced:

- Access control lists (ACLs) for UNIX System Services
- IBM Policy Director Authorization Services for z/OS and OS/390 support

11.2.1 Access control lists (ACLs) for UNIX System Services

UNIX file security on z/OS uses permission bits to control access to files, in accordance with the POSIX standard. However, the permission bit model does not allow for granting and denying access to specific users and groups, such as is possible using RACF profiles.

This support for access control lists (ACLs) allows you to control access to files and directories by individual user (UID) and group (GID). ACLs are contained within the file system, therefore file security is portable. You can manage ACLs by using UNIX setfacl/getfacl commands. To manage an ACL for a file, you must either be the file owner or have superuser authority (that is, have UID=0 or have READ access to SUPERUSER.FILESYS.CHANGEPERMS in the UNIXPRIV class). These are the same requirements for changing the current permission bits.

This function is provided by the introduction of access control lists (ACLs) in the UNIX file system. An ACL is a SAF-owned construct which resides within the file system. The RESTRICTED attribute of a user will now be applicable to file and directory access.

For more information about managing ACLs, see “Access control list (ACL) support for V1R3” on page 122.

New RESTRICTED.FILESYS.ACCESS profile

If the new RESTRICTED.FILESYS.ACCESS profile in the UNIXPRIV class is defined, RESTRICTED users cannot be granted file access via the “other” bits, whether or not an ACL exists. You can use RACF commands to define the RESTRICTED.FILESYS.ACCESS UNIXPRIV class profile as follows:

```
RDEFINE UNIXPRIV RESTRICTED.FILESYS.ACCESS UACC(NONE)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Note that RESTRICTED.FILESYS.ACCESS is checked for RESTRICTED users regardless of whether an ACL exists, so this function can be exploited regardless of whether you use ACLs or not. And using UACC(READ) on RESTRICTED.FILESYS.ACCESS does not work, since a RESTRICTED user cannot be granted access via UACC.

In any case, you can permit the RESTRICTED user (or one of its groups) to RESTRICTED.FILESYS.ACCESS. This permit does not grant the user access to any files. It only allows the “other” bits to be used in access decisions for this user. To permit the RESTRICTED user to RESTRICTED.FILESYS.ACCESS profile, do the following:

```
PERMIT RESTRICTED.FILESYS.ACCESS CLASS(UNIXPRIV) ID(RSTDUSER) ACCESS(READ)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

SUPERUSER.FILESYS still applies to RESTRICTED users regardless of the existence of a RESTRICTED.FILESYS.ACCESS profile.

New SUPERUSER.FILESYS.ACLOVERRIDE profile

When the new SUPERUSER.FILESYS.ACLOVERRIDE profile in the UNIXPRIV class is defined, it supersedes the check to SUPERUSER.FILESYS when a matching ACL entry (or entries, in the case of groups) is found, but does not grant access.

You can use this to provide a mechanism of scoping UNIXPRIV access authority to certain file system subtrees. In some cases, you can permit the user or group to SUPERUSER.FILESYS.ACLOVERRIDE with whatever access level would have been required for SUPERUSER.FILESYS. The following commands describe how you can permit a user to SUPERUSER.FILESYS.ACLOVERRIDE profile:

```
RDEFINE UNIXPRIV SUPERUSER.FILESYS.ACLOVERRIDE UACC(NONE)
PERMIT SUPERUSER.FILESYS.ACLOVERRIDE CLASS(UNIXPRIV) ID(BIGADMIN) ACCESS(READ)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Note that SUPERUSER.FILESYS authority will still be required when an ACL does not exist for the file. SUPERUSER.FILESYS.ACLOVERRIDE is checked only if an ACL entry match is found for the user or one of its groups, but no ACL entry granted the requested access. If SUPERUSER.FILESYS.ACLOVERRIDE does not exist or there was no matching ACL entry, SUPERUSER.FILESYS is checked as it is prior to z/OS V1R4.

Refer to “SUPERUSER.FILESYS.ACLOVERRIDE profile” on page 124 for more information.

11.2.2 Policy Director Authorization Services for z/OS and OS/390 support

Distributed applications have their own security administration tools, their own authentication models, and their own authorization policies. All these components need to agree on who a given user is. If they do not, many steps are required as a user's role changes. Figure 11-1 on page 208 shows the distributed applications that have their own tools, models, and policies.

Figure 11-1 Example of the distributed applications

Authorization API support

Policy Director Authorization Services for z/OS and OS/390 provides authorization API support for Policy Director-enabled applications on z/OS and OS/390. It combines features of both Local and Remote modes; the authorization daemon that is ported to USS and manages a local policy database, and PD-enabled applications invoke new SAF services for authorization decisions.

SAF support is a re-implementation, not a port. It allows z/OS and OS/390 to plug-and-play in an existing Tivoli® Policy Director secure domain.

Policy Director Authorization Services for z/OS and OS/390 extends the value of Tivoli Policy Director to the z/OS and OS/390 platforms. It allows customers to leverage existing Tivoli Policy Director security policy definitions and LDAP user registry. And it establishes a relationship between native identity and Policy Director identity. New SAF authorization interfaces allow a user to be identified via host identity. It provides core functionality for Tivoli Policy Director blades on z/OS and OS/390.

Tivoli Policy Director

Tivoli Policy Director is designed to unite core security technologies around common security policies. It maintains a central user registry that contains users, groups, and authentication information. Tivoli Policy Director also maintains a model of the Protected Objectspace, which is hierarchically organized. It defines permitted actions on objects by using access control list templates that are attached to entries in objectspace. Tivoli Policy Director provides an API for making authorization queries and provides a number of “blades” that use it.

Tivoli Policy Director depends on two databases of registry information: a user registry (commonly an LDAP directory), and an authorization policy database. The Protected Objectspace is defined as a hierarchy and policy templates, or access control lists (ACLs), which can be defined at any level.

An ACL represents a relationship between a user, or a group of users, and a resource. Applications may exploit an Open Group ratified standard programming model to query policies, and Tivoli also provides a number of offerings, or blades, which provide key security functionality:

- NetSEAL, a solution for securing TCP/IP communication
- WebSEAL, for managing access control for resources such as URLs, HTML files, and Java servlets
- PD for MQ series, a solution for securing MQ series queues

RACF support for Policy Director Authorization Services

RACF provides support for new SAF callable services exploited by Policy Director Authorization Services:

- R_cacheserv
- R_proxyserv

The RACF SMF Unload utility is enhanced to unload Policy Director Authorization Services audit records. The SAF interface is extended to provide support for aznCreds and aznAccess, and SAF trace is enhanced not only for R_cacheserv and R_proxyserv, but also aznCreds and aznAccess.

11.3 Security Server RACF enhancement in z/OS V1R4

In z/OS V1R4, the following new and changed RACF functions are introduced:

- Enterprise Identity Mapping Services (EIM)
- UNIX security management usability enhancements
- Program access to data sets (PADS)

11.3.1 Enterprise Identity Mapping Services (EIM)

Enterprise Identity Mapping Services (EIM) defines a set of services and extensions to LDAP. It will be available on all eServer platforms - iSeries™ (OS/400), zSeries (z/OS), pSeries (AIX) and Linux. EIM is an infrastructure that user administration applications, servers, operating systems, and audit tools can leverage to provide complete solutions to two classes of problems that customers may experience:

- Transforming the user identity associated with a work request as it moves between systems through a multi-tiered application
- User administration in a heterogeneous environment

RACF support for EIM in z/OS V1R4 enables customers to configure z/OS and servers to use an EIM domain. The services required to find mappings between userids and/or enterprise identifiers and to administrate those mappings have not been supported yet.

This support adds new fields to the RACF template IRRTEMP1. IRRMIN00 must be run with PARM=UPDATE to add the changed templates to existing RACF databases. The changes will occur in the next IPL.

EIM architecture

The approach most commonly used today is that each system registry provides its own mappings of system user identity-to-new user identity. For example, RACF userids map to distinguished names in digital certificates and UNIX System Services UIDs. Administrators have a reasonably easy job as long as the computing enterprise is made of all z/OS systems that use RACF data sharing or RRSF.

However, when another platform such as OS/400 is added, administrators face a growing workload, because now they have to manage—for each person or entity— the relationship to two system identities and duplicate the mappings of the system identity to a distinguished name and UIDs.

In contrast, EIM architecture stores mappings in a centralized, distributed registry, LDAP. Information is stored in LDAP to allow a server or application to map one user identity to another as long as the two identities belong to the same enterprise user or entity.

A mapping consists of two parts: a source user identity in a registry to a unique enterprise identifier, and the unique enterprise identifier to a user identity in the same or different registry. The mapping operation can be visualized as Figure 11-2.

```
Source identity --> EIM identifier --> Target identity
```

Figure 11-2 The mapping operation in EIM

The enterprise identifier can represent an individual or entity within an organization. There can be more than one source identity in the same or different registry that maps to a given identifier, and an identifier may map to one or more target identities in the same or different registry.

EIM keeps relationships between enterprise identifiers and user identities in registries. Source associations indicate a relationship's source identity to the EIM identifier, and target associations indicate a relationship's EIM identifier to the target identity, as shown in Figure 11-3.

This EIM data is stored in LDAP. EIM also has a C/C++ programming interface which administrates EIM domains and performs lookups of mappings.

Figure 11-3 EIM architecture

The C/C++ programming interface that performs lookups of mappings looks as follows:

```
eimGetTargetFromSource(registry 1, user ID 1, registry 2)=user ID 2
```

Retrieving mappings from an EIM domain

Once an EIM domain has been configured, then servers may use mapping lookup C/C++ EIM services. Figure 11-4 illustrates how a multitiered application might use EIM mapping services.

Figure 11-4 Multitier application using EIM

Starting with the client application, the flow indicated in Figure 11-4 proceeds as follows:

1. The identity is authenticated using a local authentication mechanism (userid/password).
2. The client application, or the system on the client's behalf, makes an eapi (EIM API) lookup call to the EIM domain controller contained in an LDAP server. EIM uses the identity provided by the client to find the identity in local client registry that shares the same enterprise identifier. EIM returns it to the client application.
3. The local credential is created from the local identity.
4. The client prepares to make a call to another server. It uses EIM to transform the local identity into the identity required by the network protocol (OS/400 userid).
5. The retrieved OS/400 userid is then passed to the server, where the process repeats itself.

Configuring z/OS for EIM

System-wide EIM settings are set through the following commands:

```
RDEFINE FACILITY IRR.PROXY.DEFAULTS EIM(DOMAINDN('ibm-eimDomain=Joes
Domain.o=ibm.c=us')) OPTIONS(ENABLE)) PROXY(LDAPHOST(ldap://some.big.host)
BINDDN('cn=EIM Lookup') BINDPW('secret'))
-or-
RDEFINE LDAPBIND IRR.EIM.DEFAULTS EIM(DOMAINDN('ibm-eimDomain=Joes Domain.o=ibm.c=us'))
OPTIONS(ENABLE)) PROXY(LDAPHOST(ldap://some.big.host) BINDDN('cn=EIM Lookup')
BINDPW('secret'))
```

Server/user-specific settings are set through the following commands. This method can be used to assign domain/bind information to an administrator's user ID, as follows:

```
RDEFINE LDAPBIND BUCKSDOMAIN EIM(DOMAINDN('ibm-eimDomain=Bucks Domain.o=ibm.c=us'))
OPTIONS(ENABLE)) PROXY(LDAPHOST(ldap://another.big.host) BINDDN('cn=EIM Application
Lookups') BINDPW('secret'))
ADDUSER SERVERID EIM(LDAPPROF(BUCKSDOMAIN))
```

The default local registry name is activated with the following commands:

```
RALTER FACILITY IRR.PROXY.DEFAULTS EIM(LOCALREGISTRY('SAF/RACF Pok1'))
SETROPTS EIMREGISTRY or ipl the system
```

The following command de-activates the local registry name temporarily:

```
SETROPTS NOEIMREGISTRY
```

The following command de-activates the local registry name permanently:

```
RALTER FACILITY IRR.PROXY.DEFAULTS EIM(NOLOCALREGISTRY)
SETROPTS EIMREGISTRY or ipl the system
```

Also, the ADDUSER/ALTUSER/LISTUSER and RDEFINE/RALTER/RLIST commands introduce an EIM segment and fields.

11.3.2 z/OS UNIX Security Management Usability enhancements

The main objective of the z/OS UNIX Security Management Usability enhancements item is to aid the RACF administrator in ensuring that RACF users and groups run with a unique UNIX identity.

- A system-wide setting prevents assignment of a UID or GID value that is already in use. To handle exceptions to the “one user ID/one UID” rule, a RACF command keyword is provided that overrides the system setting and allows assignment of a shared UID or GID.
- A SEARCH enhancement allows an administrator to determine the set of users or groups assigned a given UID or GID.
- A mechanism is provided to automatically assign an unused UID or GID value to a user or group, which helps reduce manual steps and potential administrative error.

SEARCH enhancement

Prior to OS/390 V2R10, profiles in the UNIXMAP class were used to map UIDs to user IDs and GIDs to group names. UNIXMAP profiles automatically maintained by RACF commands and RLIST UNIXMAP Unnn ALL shows all users with UID(nnn). But in OS/390 V2R10, customers migrating to stage 3 of application identity mapping lose that capability.

The **SEARCH** command is enhanced to map UIDs and GIDs, as follows:

```
SEARCH CLASS(USER) UID(0)
OMVSKERN
BPXOINIT
SUPERGUY

SEARCH CLASS(GROUP) GID(99)
RACFDEV

SEARCH CLASS(USER) UID(1234567)
NO ENTRIES MEET SEARCH CRITERIA
```

This support requires at least application identity mapping stage 2; otherwise, the following message will be displayed:

```
SEARCH CLASS(USER) UID(0)
The UID keyword requires application identity mapping to be implemented.
```

This support does not require any particular authority. When UID or GID is specified in SEARCH command, all other keywords (except CLASS) are ignored.

Prevention of shared UID/GIDs

The POSIX standard does not require UIDs and GIDs to be unique. So RACF does not require UIDs and GIDs to be unique, although keeping them unique is recommended. You can use the IRRICE report against DBUNLOAD output to find duplicate UIDs, but there is nothing real-time to prevent or warn you about non-unique UIDs or GIDs.

Now, the new SHARED.IDS profile in the UNIXPRIV class acts as a system-wide switch to prevent assignment of an ID that is already in use. Generic characters are not allowed in the profile name, and a discrete profile name must be used.

This support requires application identity mapping stage 2 or 3. Uniqueness is guaranteed within the scope of a GRSplex. It does not affect pre-existing shared IDs, so customers must clean those up separately, if desired. There is no pre-requisite for using this support; customers can use the IRRICE report to find shared UIDs, and a new IRRICE report is shipped to find shared GIDs.

The following commands show the prevention of shared UID/GIDs with this support:

```
RDEFINE UNIXPRIV SHARED.IDS UACC(NONE)
SETROPTS RACLIST(UNIXPRIV) REFRESH

ADDUSER MARCY OMVS(UID(12))
IRR52174I Incorrect UID 12.  This value is already in use by BRADY.

ADDGROUP ADK OMVS(GID(46))
IRR52174I Incorrect GID 46.  This value is already in use by PATS.
```

If you are not at least at application identity mapping stage 2, you will see the following message:

```
ADDUSER MARCY OMVS(UID(11))
IRR52176I SHARED.IDS is defined, but application identity mapping is not implemented.
```

If specifying more than one user/group, you will see the following message:

```
ADDUSER (HOUGH  MACOMB) OMVS(UID(12))
IRR52185I The same UID cannot be assigned to more than one user.
```

Shared keyword

Keep in mind that there are valid reasons to assign a non-unique UID/GID. For example, it is necessary to assign UID(0) to started task user IDs. If you do so, you can now use the new SHARED keyword in the OMVS segment of the ADDUSER, ALTUSER, ADDGROUP, and ALTGROUP commands. The new SHARED keyword requires SPECIAL, or at least READ authority, to the SHARED.IDS profile.

Profile-level audit settings can be used to log successes and failures to SHARED.IDS. RACF can generally only log successes in the UNIXPRIV class. There is no FIELD class checking for the SHARED operand because there is no corresponding field in the RACF database, as follows:

```
PERMIT SHARED.IDS CLASS(UNIXPRIV) ID(UNIXGUY) ACCESS(READ)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

```
AU OMVSKERN OMVS(UID(0) SHARED)
AG (G1 G2 G3) OMVS(GID(9) SHARED)
```

If you do not have the required authority for the SHARED keyword, you will see the following message:

```
AU MYBUDDY OMVS(UID(0) SHARED)
IRR52175I You are not authorized to specify the SHARED keyword.
```

The SHARED keyword is ignored (and also no authorization check to SHARED.IDS) when one of the following conditions occurs:

- The UID or GID keyword is omitted.
- The SHARED.IDS profile is not RACLISTed.
- The specified UID or GID value is identical to the current UID or GID value.
- The specified UID or GID value is unique.

Automatic UID/GID assignment

Apart from UID(0), the UID and GID values are entirely arbitrary from an authorization perspective—but they should really be unique. When defining OMVS segments, using the LISTUSER * or LISTGROUP * command is very inefficient. Otherwise, it need a manual process, for example, use employee serial number.

There is a new AUTOUID keyword in the OMVS segment of the ADDUSER and ALTUSER commands. There is also a new AUTOGID keyword in the OMVS segment of the ADDGROUP and ALTGROUP commands is added. Derived UID/GID values are guaranteed to be unique, as follows:

```
ADDUSER MELVILLE OMVS(AUTOUID)
IRR52177I User MELVILLE was assigned an OMVS UID value of 4646.

ADDGROUP WHALES OMVS(AUTOGID)
IRR52177I Group WHALES was assigned an OMVS GID value of 105.
```

This support does not require additional authority beyond what is required for UID or GID. FIELD checking for UID or GID still applies. But there is no FIELD class checking for the AUTOUID or AUTOGID operand because there is no corresponding field in the RACF database.

You can use the APPLDATA of the new BPX.NEXT.USER profile in the FACILITY class to derive candidate UID/GID values. The APPLDATA consists of two qualifiers separated by a forward slash (/). The left qualifier specifies the starting UID value, or range of UID values. The right qualifier specifies the starting GID value, or range of GID values. The qualifiers can be null or specified as NOAUTO to prevent automatic assignment of UIDs or GIDs. APPLDATA is verified at time of use, not when it is defined. The FACILITY class does not need to be active or RACLISTed.

The following example shows RDEFINE FACILITY BPX.NEXT.USER APPLDATA('data'):

```
RDEFINE FACILITY BPX.NEXT.USER APPLDATA('data')
```

The following example shows valid APPLDATA values:

```
1/0
1-50000/1-50000
NOAUTO/100000
/100000
10000-20000/NOAUTO
10000-20000/
```

The following example shows invalid APPLDATA values and a message that will be displayed when adding a user with AUTOID with invalid APPLDATA.

```
/
123B
2147483648     /* higher than max UID value */
555/1000-900

ADDUSER MARQUEZ OMVS(AUTOUID)
IRR52187I Incorrect APPLDATA syntax for the BPX.NEXT.USER profile.
```

When AUTOUID or AUTOGID is issued, RACF extracts the APPLDATA from BPX.NEXT.USER and parses out the starting value. Then it checks to see if the value is already in use. If so, the value is incremented and checked again until an unused value is found; then RACF assigns the value to the user or group. Finally, RACF replaces the APPLDATA with the new starting value. Administrators can change the APPLDATA at any time by using the RALTER command.

If candidate UID or GID values have been exhausted, the following message will be displayed:

```
IRR52181I The BPX.NEXT.USER profile has run out of possible UID values.
```

This support must be enforcing uniqueness with SHARED.IDS in order to use AUTOUID/AUTOGID. And it requires application identity mapping stage 2 or 3. You cannot specify AUTOUID/AUTOGID without first defining BPX.NEXT.USER. You cannot use automatic UID assignment with a list of names, for example, ADDUSER (TOM TROY ADAM) OMVS(AUTOUID). AUTOUID/AUTOGID and SHARED keywords are mutually exclusive.

11.3.3 Program control and program access to data sets (PADS)

Program control authorizes users to programs via PROGRAM class profiles. With program control, programs can be protected. Program access to data sets (PADS) authorizes users to data sets while running a particular program via DATASET profiles. With PADS, data sets can be protected by restricting access to specified users only when running particular programs.

Prior to z/OS V1R4, when specifying a program name in the conditional access list, the name of the program that actually did the loading needed to be known. Situations where the user invokes one program, which actually opens another data set, required you to know both program names rather than just the high level program name. With this enhancement, you only have to know the high level name.

New enhanced security mode for PADS

In the RACF profile IRR.PGMSECURITY, in the RACF FACILITY class, a new enhanced security mode can be specified which provides improved usability and increased security when using PADS.

Using IRR.PGMSECURITY, the APPLDATA specifies whether RACF will operate in basic, enhanced, or enhanced-warning PGMSECURITY mode, as specified by using the APPLDATA keyword. The new modes are:

- If the APPLDATA is exactly 'ENHANCED', then RACF will run in enhanced PGMSECURITY mode.
- If the APPLDATA is exactly 'BASIC', then RACF will run in basic PGMSECURITY mode.
- If the APPLDATA is empty or contains any other value, RACF will run in enhanced PGMSECURITY mode—but in warning mode, rather than failure mode 'ENHWARN'.

Recommendation: Use the ENHANCED-WARNING program security mode as part of your implementation of ENHANCED program security mode.

With ENHANCED-WARNING mode, RACF ensures that programs accessing data sets through PADS, or running execute-controlled programs, meet the added restrictions of ENHANCED mode. However, if they do not meet the added restrictions, RACF still allows the access if it would have worked in BASIC mode. This allows you to test your setup to make sure it is suitable for ENHANCED mode, while continuing to operate like BASIC mode while you adjust your profiles.

When you migrate to the new mode, you will have some profiles defined in the PROGRAM class but probably none of them specify APPLDATA('MAIN') or APPLDATA('BASIC'), as those specifications don't mean anything in BASIC program security mode. Therefore, specify the IRR.PGMSECURITY profile defined in the FACILITY class and use the APPLDATA to specify your desired mode

For example, to use the new ENHANCED-WARNING mode, do the following:

1. Use the RDEFINE command to define the IRR.PGMSECURITY profile in the FACILITY class, and specify an APPLDATA value other than ENHANCED or BASIC; for example:
   RDEFINE FACILITY IRR.PGMSECURITY APPLDATA('ENHWARN')
2. Issue the SETROPTS REFRESH command to change modes.

   ETROPTS WHEN(PROGRAM) REFRESH

To ease migration from BASIC to ENHANCED program security mode, the mode switch does not affect systems running any release earlier than z/OS V1R4. It also does not affect jobs, started tasks, or TSO sessions that are already running. For this reason, you should IPL the system at least once while in ENHANCED-WARNING mode to ensure that you test any jobs, started tasks, and TSO users that started before you migrate from BASIC to ENHANCED program security mode.

While running in ENHANCED-WARNING mode, you may receive messages ICH427I or ICH430I to indicate the need for further necessary changes. After receiving the messages, making the relevant changes, and allowing a sufficient test period of running in ENHANCED-WARNING mode without getting further messages, you can switch to ENHANCED program security mode.

For additional information on this new enhancement, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683.

The mode becomes effective at SETR WHEN(PROGRAM) or SETR WHEN(PROGRAM) REFRESH. The default mode is BASIC.

Program control and PADS before this support are not changed in customer's current operation if no FACILITY IRR.PGMSECURITY profile or if FACILITY class not activated.

RDEFINE PROGRAM defines each program control. You can add APPLDATA to specific PROGRAM class profiles. ADDMEM is still needed for library data. Following is the definition of specific program control:

```
RDEFINE PROGRAM pgmname APPLDATA('value')
```

The APPLDATA values are as follows:

MAIN Trusted enhanced mode program

BASIC Program exempted from enhanced PGMSECURITY and it overrides ENHANCED mode.

anything else Not trusted in enhanced mode

SPECIFIC profiles are only valid. First program must be specified as MAIN or BASIC for authorization. MAIN applies only to first program in // EXEC PGM=program or TSOEXEC program. BASIC applies to first program of any TCB and to all daughter TCBs. BASIC allows use of old security programs with ENHANCED mode. You should realize that BASIC weakens security in ENHANCED mode.

For new PADS, You should specify any first program for any mother TCB rather then OPENing program. First program describe in // EXEC PGM=program or TSOEXEC program.

Let us consider some example of this support. When you use // EXEC PGM=A and A LINKs to B which does OPEN, prior to z/OS V1R4, you must specify B in conditional access list, and may need to specify A unless defined as NOPADCHK. Now you can specify either A or B. If ENHANCED mode, A must be MAIN or BASIC. You still need to specify the other program unless defined as NOPADCHK.

When you use // EXEC PGM=A and module A ATTACHs to module B which does OPEN, prior to z/OS V1R4, you must specify B in conditional access list, and may need to specify A unless defined as NOPADCHK. Now you can specify either A or B. If ENHANCED mode, A must be MAIN or BASIC or B must be BASIC. You still need to specify the other program unless defined as NOPADCHK.

Archived

Archived

12

Security Server PKI Services

This chapter contains a description of the new component of z/OS V1R3, Security Server PKI Services, which includes:

- Digital certificates
- Browser certificates
- Server certificates
- PKI Services architecture

We also describe the z/OS V1R4 enhancements to the Security Server PKI Services, which are as follows:

- Sysplex support
- Event notification via e-mail
- Distinguished name qualifier support
- LDAP password encryption
- PKCS#7 certificate chain support
- Key generation via PCICC
- CERTAUTH certificate defaults

Archived

© Copyright IBM Corp. 2003

12.1 Security Server PKI Services in z/OS V1R3

The PKI Services component allows you to establish a PKI infrastructure and serve as a certificate authority for your internal and external users, issuing and administering digital certificates in accordance with your own organization's policies. Your users can use a PKI Services application to request and obtain certificates through their own browsers, while your authorized PKI administrators approve, modify, or reject these requests through their own Web browsers. The Web applications provided with PKI Services are highly customizable, and a programming exit is also included for advanced customization.

You can allow automatic approval for certificate requests from certain users, and add host IDs, such as RACF user IDs, to certificates you issue for certain users in order to provide additional authentication. You can also issue your own certificates for browsers, servers, and other purposes, such as virtual private network (VPN) devices, smart cards, and secure e-mail.

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

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

This support for PKI Services includes certificate support for the following:

- Administrative approval processes using a Web-based interface for the selective approval, rejection, and revocation of certificates (life cycle management). Clients can also renew and revoke their own certificates.
- Certificates created are posted to an LDAP directory.
- Certificate revocation lists (CRLs) are maintained and posted to an LDAP directory.
- Coexistence with the RACF V2R10 SPE. The installation chooses the certificate generation provider:
  - SAF - uses the existing limited Security Server (RACF) certificate support
  - PKI - uses the new PKI Services component
- The R_PKIServ SAF service is extended to provide additional functions that support the programmatic request of certificates and certificate management used by the Web user and Web administrator functions.

12.1.1 New component of z/OS Security Server

PKI Services is a new component of the z/OS V1R3 Security Server. It is always enabled, with or without RACF, but is closely tied to RACF because PKI Services has a SAF callable service interface front-ended by RACF or an equivalent security product. It is a complete Certificate Authority (CA) package, and has the following functions to provide full certificate life cycle management:

- User request-driven via customizable Web pages
- Automatic or administrator approval process
- End user/administrator revocation process

The user Web page interface is based on the PKISERV RACF SPE delivered in the OS/390 V2.10 time frame (downloadable from the RACF Web site). But PKI Services delivers much more function than the PKISERV SPE.

12.1.2 Digital certificate

Figure 12-1 shows a basic digital certificate. A digital certificate is a binding of name to public key. Certificates are created by certificate authorities (CAs); the CA declares that subject A owns public key XYZ. And certificates are signed by CA to prevent tampering.

Figure 12-1 Basic digital certificate

Certificates have several uses, but the most common is server-side SSL. The following examples are certificates used for server authentication and data encryption in servers on zSeries or other platforms:

- SSL Webservers
- VPNs
- Internet Routers

The following examples are used for clients:

- SSL client authentication (for example, accessing protected Web sites)
- Message encryption and/or signing (S/MIME e-mail)
- File encryption

12.1.3 Certificate life cycle

Figure 12-2 on page 222 shows a typical certificate life cycle. It starts with a certificate request that is created by the person desiring the certificate. The request is called a PKCS#10 encoding. For client certificates, it may be created by the browser as directed by a Web page. For server certificates, it may be created by proprietary software running on the

server itself (for example, RACF or gskkyman for z/OS servers). Then the request is submitted for approval. If approved either manually or automatically, a certificate is issued to and then used by the requestor until it expires or is revoked. If not revoked, it may be renewed and the cycle repeats.

Figure 12-2 Certificate life cycle

12.1.4 Browser certificates

Figure 12-3 on page 223 shows typical usage as it applies to browser certificates.

Archived

Figure 12-3 Browser certificate process

In the example shown in Figure 12-3, the Web user submits the PKCS#10 certificate request (1), which is queued for approval by the administrator (2). The administrator reviews the request and approves or rejects (3).

If approved (4), it is issued and stored. The certificate is returned to the Web user when queried (6), and published to an LDAP directory (7). The certificate revocation list (CRL) is also published to LDAP on a continuous basis.

The Web user then uses the certificate to authenticate to an SSL client authentication-protected Web site (8). The SSL handshake will validate the certificate and check the CRL (9). If OK, the user gains access (10).

12.1.5 Server certificates

Figure 12-4 on page 224 is typical usage as it applies to Webserver certificates.

Figure 12-4 Server certificate process

In the example shown in Figure 12-4, the Webserver administrator uses server-specific software to generate a PKCS#10 request (1a). This is copied (1b) and pasted (1c) into the certificate request Web page and submitted. Steps for queuing/approving/issuing/retrieving the certificate are identical to the preceding browser flow (2-7).

The Webserver administrator installs the certificate into the Webserver (8) and brings it online. Web users may now visit the SSL-protected Web site (9). If client authentication is enabled, the client's certificate would be validated using the CRL in LDAP (10). If OK, the user gains access (11)

12.1.6 z/OS PKI Services architecture

A complete PKI Services system is made up of several components.

The z/OS HTTP server provides the end-user and administrator interface. Customizable Web page contents are defined in the templates file. CGIs read file and submit requests to R_PKIServ. An optional customer exit pkiexit is provided. Finally, the HTTP server invokes z/OS PKI Services through the SAF interface R_PKIServ.

R_PKIServ is an SAF callable service backed by RACF or an equivalent product. End-user functions of the service are request, retrieve, verify, revoke, or renew a certificate. Administrator functions of the service are query, approve, modify, or reject certificate requests, query and revoke issued certificates. Requests are verified by RACF and submitted to the PKI Services daemon. And it may create SMF auditing records.

The PKI Services daemon has multiple service and background threads. It maintains requests and issued certificates in VSAM data sets. Services threads are for incoming requests and background threads are for certificate/certificate revocation list (CRL) issuance. There are two VSAM DBs, one for requests (ObjectStore), and the other for the issued certificate list (ICL).

The Open Cryptographic Services Facility (OCEP) and the Open Cryptographic Enhanced Plug-ins (OCEP) provide the cryptographic facilities for PKI Services. OCEP is used for accessing the CA certificate and private key in RACF. OCSF is used for BSAFE or ICSF (Hardware) cryptographic engines.

The LDAP Server is used as the public repository for issued certificates and CRLs; there are no special schema requirements.

Figure 12-5 shows a view of the architecture. Customers may provide a PKI Exit to enable additional processing. CGIs are written in REXX, thus requiring a RACF glue routine because REXX cannot create the structure parameters required by the callable service. A RACF setup REXX exec is also provided to create the RACF environment needed by PKI Services.

Figure 12-5 PKI Services architecture

12.1.7 Prerequisite products

The following products must be installed prior to configuring PKI Services:

- IBM z/OS HTTP Server
  - Working in at least non-SSL mode.
- LDAP Directory
  - The z/OS LDAP Server is recommended; if used, then the TDBM back-end is required.
  - The PKIX schema is required. The minimum schema shipped with the z/OS LDAP Server (schema.user.ldif) is PKIX-compliant.
- Cryptographic Services OCSF and OCEP
  - ICSF (optional).
- RACF (or equivalent)
  - Currently there is no RACF equivalent.

12.2 Security Server PKI Services enhancement in z/OS V1R4

This support for PKI Services includes the following:

- Support of e-mail notification for completed certificate requests and expiration warnings.
- Support of MAIL, STREET and POSTALCODE distinguished name qualifiers.
- The RACDCERT command and the R_PKIServ callable service to support PKCS#7 certificate chains are enhanced.
- Removal of clear text LDAP passwords from the pkiserv.conf file by storing them in RACF profiles
- Use of the PCI cryptographic coprocessor to generate key pairs, thus eliminating software key exposures.
- Update of the list of default CERTAUTH certificates in RACF. Providing VSAM RLS for the ICL and ObjectStore VSAM data sets in support of SYSPLEX enablement.

12.2.1 Sysplex support

In z/OS V1R3, there is no Parallel Sysplex support. Therefore, multiple instances of PKI services in a sysplex would all be independent. For example, databases (VSAM data sets) needed to be separated.

In z/OS V1R4, multiple instances can share databases by accessing via VSAM record level sharing (RLS). VSAM RLS requires CF, couple data sets, and the appropriate storage class. For more information about VSAM RLS, see *z/OS DFSMSdfp Storage Administration Reference*.

New sample JCL IKYMVSAM is added to migrate existing data sets to the new storage class. Specifying `SharedVSAM=T` in configuration file tells PKI Services to use VSAM RLS.

Figure 12-6 on page 227 is a view of a possible Webserver configuration. Each image in the sysplex has one instance of PKI Services front-ended by a Webserver. One master Webserver manages which image to send the work to. All images share one Coupling Facility, one set of VSAM data sets, and one LDAP directory.

Archived

Figure 12-6 Webserver configuration

To enable sysplex sharing, we move some work files into the VSAM request data set. If starting z/OS V1R4 PKI Services for the first time, those files are moved automatically. Therefore, the first image to start must have read/write access to the files listed.

The HFS files moved to the VSAM request data set:

- Scheduler file (/var/pkiserv/pkica.j*)
- Serial file (/var/pkiserv/pkica.j*)

Old files are renamed with the .MIGRATED extension.

A new SYSTEMS ENQs is used to serialize images and a major name is SYSZPKI2. Migration and some normal processing are serialized by new ENQs.

12.2.2 Event notification via e-mail

PKI Services now has the capability to notify end users when their certificate requests are complete and when their certificates are about to expire. A new request named field “NotifyEmail” allows end users to supply a notification e-mail address. The notification e-mail notifies you when the request is complete (ready or rejected), and when the certificate is about to expire. The NotifyEmail address is stored in the LDAP directory as a MAIL attribute. The Communication Server's sendmail utility is used to send e-mail messages. The customer supplies message forms, which are specified by pathname via a configuration file; see Example 12-1 on page 228.

Example 12-1 User-specified message forms

```
ReadyMessageForm=/etc/pkiserv/readymsg.form
RejectMessageForm=/etc/pkiserv/rejectmsg.form
ExpiringMessageForm=/etc/pkiserv/expiringmsg.form
```

Sample message forms are shipped in the samples directory. Example 12-2 shows a sample ready message. PKI Services recognizes the following four variables, and will substitute accordingly.

%%transactionid%% The unique value returned when a certificate request has been submitted.

%%requestor%% The name that the user wishes to be known by; usually it is the same as the common name, except when requesting a server certificate.

%%dn%% The subject's distinguished name.

%%notafter%% The certificate expiration date/time.

Example 12-2 Sample ready message

```
From:dime-o-cert PKI
Subject:Certificate Ready For Pick Up

Attention - Please do not reply to this message as it was automatically sent by a   service
machine.

Dear %%requestor%%,

Thank you for choosing dime-o-cert PKI. The certificate you requested for subject %%dn%% is
now ready for pickup. Please visit http://www.dimeocert.com/PKIServ/camain.rexx to retrieve
your certificate. You will need the transaction ID listed below and your passphrase that
you entered when you submitted the request.

%%transactionid%%
```

12.2.3 Additional distinguished name (DN) qualifier support

The subject's distinguished name (DN) is specified via qualifier name/value pairs when a request is submitted (for example, DN is CN=Jim Sweeny,O=IBM,C=US). Qualifiers are named fields in the Web pages, and can be use- supplied or hardcoded.

In z/OS V1R3, the following DN qualifiers are supported:

- CommonName
- Title
- OrgUnit
- Org
- Locality
- StateProv
- Country

In z/OS V1R4, the following qualifiers are added:

- Email
- PostalCode
- Street

If both Email and NotifyEmail are specified, they must be equal.

12.2.4 LDAP password encryption

PKI Services posts information to LDAP. It needs bind dn and password to do so.

In z/OS V1R3, the bind password is specified in the clear, as the configuration parameter AuthPWDn.

Otherwise, the LDAPBIND Class profile specified in the RACF_entity parameter must have a PROXY segment previously created through a RDEFINE or RALTER command. If the RACF_entity is not specified, the IRR.PROXY.DEFAULTS profile in the FACILITY Class must have a PROXY segment previously created through a RDEFINE or RALTER command.

12.2.5 PKCS#7 certificate chain support

When setting up a secure server (in particular an SSL Webserver), the server needs to be loaded with the entire certificate hierarchy, from the root CA down to the end entity (the end entity in this case is the server itself). For short hierarchies, loading individual certificates manually is not a real problem; however, for longer hierarchies, this is a labor-intensive and error-prone task.

PKCS#7 certificate packages contain the entire chain, and software that can read the entire chain eliminates the manual work. Figure 12-7 shows the certificate chains.

Figure 12-7 PKCS#7 certificate chains

RACF now has full support for PKCS#7 certificate packages. The RACF command to add certificates is RACDCERT. For PKCS#7 packages, the certificates are added in order from top CA down to the subject (end entity).

The RACDCERT ADD command will add CA certificates to CERTAUTH if user-authorized. It requires CONTROL authority to the FACILITY class profile IRR.DIGTCERT.ADD or RACF SPECIAL authority.

RACDCERT's “trust value” is an optional on/off switch and certificates marked NOTRUST cannot be used. If not specified on the command, RACDCERT will determine the trust of the top CA dynamically.

TRUST If self-signed, or if the signer is already trusted.

NOTRUST If NOTRUST, trust of subsequent certificates inherited from signer. Inconsistencies cause NOTRUST (expired or unknown signature algorithm).

12.2.6 Key generation via PCICC

Prior to z/OS V1R4, the RACDCERT GENCERT command generated an RSA key pair using software.

In z/OS V1R4, PCI Cryptographic Coprocessor (PCICC) is used for generation if a new PCICC keyword is specified.:

- <no keyword specified> - key generated in software, then stored in RACF DB
- ICSF specified - key generated in software, then stored in PKDS
  - Fails in z/OS V1R4 if ICSF's PKA features are not active (instead of saving as a software key, as in z/OS V1R3)
- PCICC specified - key generated using PCICC, then stored in PKDS
  - Fails if ICSF PKA features and/or PCICC are not active

12.2.7 Additional default CERTAUTH

The following three new CERTAUTH certificates are added to the default list:

- Verisign Class 1 Individual CA
- Verisign Class 2 Individual CA
- Verisign International Svr CA

The following two expiring CERTAUTH certificates are replaced:

- Verisign Class 2 Primary CA
- Verisign Class 3 Primary CA

The following defunct CERTAUTH certificate is no longer added:

- IBM World Registry™

Archived

13

Security Server LDAP server

This chapter describes the Security Server LDAP server enhancements to z/OS V1R4 as follows:

- DIGEST-MD5 and CRAM-MD5 authenticate support
- Transport layer security (TLS) support
- IBM-entryuuid support
- Modify DN operation
- Access control list (ACL) support
- Activity logging
- RDBM and JNDI removal

Archived

© Copyright IBM Corp. 2003

13.1 Security Server LDAP Server enhancements in z/OS V1R4

For z/OS V1R4, the z/OS LDAP server provides the following new capabilities:

- z/OS Managed System Infrastructure (msys) for Setup provides a configuration wizard and property sheets to be used for setting up a new LDAP server or managing an existing LDAP server.
- Both the (C/C++) client and server are updated with new authentication methods: DIGEST-MD5 and CRAM-MD5. These authentication methods are prescribed IETF RFC 2829 and RFC 2831. Interoperability is improved for any applications that make use of these methods.
- Transport Layer Security (TLS) support allows an application to control which LDAP operations are secured with SSL/TLS. Support for TLS Version 1 is provided by the LDAP client and server's use of System SSL, including new cipher specifications introduced with TLS Version 1.
- Expanded support for renaming directory entries allows you to rename or move any entry as long as the DN is still managed by the same TDBM back-end.
- Entry Universal Unique Identifier (UUID) support identifies an entry uniquely within a server, even if the entry's name changes. A utility is provided to add entry UUIDs to each entry already existing in an LDAP directory (TDBM backend) that is migrated from a previous release.
- ACL enhancements to allow attribute-level access control and the ability to explicitly deny access to information.
- Improved server performance.
- A new server activity log allows a system administrator to produce a log of server activity. This support is similar to logging capabilities that are provided by other popular LDAP servers.

The following function is *removed* in z/OS V1R4:

- The RDBM back-end and its associated parts (TDBM is the replacement).
- IBM's JNDI implementation (Sun's JNDI is the replacement).

13.1.1 DIGEST-MD5 and CRAM-MD5 authenticate support

The password is not passed in the clear between the LDAP server and client while attempting to do a CRAM-MD5 or DIGEST-MD5 bind. This prevents a hacker from intercepting the password and then attacking the server.

DIGEST-MD5 and CRAM-MD5 binds to the z/OS LDAP server do not require any additional products or software to be installed and running on the system. These binds are automatically available.

Note the following:

- DIGEST-MD5 is now required to be implemented for all LDAP v3-compliant servers and clients.
- If native authentication is turned on in a subtree, and a DIGEST-MD5 or CRAM-MD5 bind is done to an entry under this subtree, the bind will *no*t be successful.
- CRAM-MD5 and DIGEST-MD5 binds are not supported to the SDBM backend. These binds are capable only to the TDBM backend.
- CRAM-MD5 and DIGEST-MD5 binds are not supported on replication.

- The userpassword attribute must be either in clear-text or two-way encryption format (DES with OCSF).
- On DIGEST-MD5 authentication, an authzid of an unspecified user ID is not supported on the z/OS client or server.

13.1.2 Transport layer security (TLS) support

RFC 2830 defines a protocol that allows a client communicating over a non-secure connection to switch to and from secure communication. The LDAP Server supports this protocol. The LDAP client APIs have not been updated. The LDAP Server and client APIs support both SSL V3 and TLS V3 protocol for secure communication.

The client connects to the LDAP Server over a non-secure connection (for example, port 389). During course of communication, the client transmits a Start TLS Extended Operation. The server accepts or declines the request. If accepted, the server expects the next communication from the client to be an SSL handshake.

Communication continues in a secure (SSL-protected) session until the client causes SSL to terminate with a TLS closure alert. After TLS is terminated, the client connection remains open and additional communication can occur. The client's authentication is reset to the anonymous state.

This support requires System SSL for z/OS V1R4.

13.1.3 IBM-entryuuid support

Enterprise Identity Mapping (EIM) needed a way to track an identity from its creation to its deletion that is constant throughout the LDAP collection regardless of where it is moved to or what it is renamed to. The identity must be unique across all LDAP servers, must not be modifiable, and must be created automatically by the server.

An attribute is added to LDAP entry which contains a unique identifier. The operational attribute is “ibm-entryuuid”.

LDAP operations and utilities

ldap_add adds a unique ibm-entryuuid to an entry. ldap_modify adds a unique ibm-entryuuid to an entry if one does not already exist for the entry.

tdbm2ldif TDBM utility unload entries include ibm-entryuuid, if it is present in an entry. The ldif2tdbm TDBM utility adds a unique ibm-entryuuid to an entry if the ldif input does not include one.

Figure 13-1 on page 234 shows how the LDAP server generates a unique uuid for each entry added to the directory. If an ldap_modify operation is performed against an entry and the entry does not contain the ibm-entryuuid attribute, then the attribute will be automatically added to the entry.

Figure 13-1 LDAP ibm-etnryuuid overview

New ldapadduuids utility

ldapadduuids adds an ibm-entryuuid to each entry in the directory tree. The command line syntax is similar to ldapsearch. The utility can specify a search argument, and only matching entries will have the ibm-entryuuid attribute added—and the utility does not add to entries that already have one. Example 13-1 shows a command that adds uuids to all entries in a tree under o=ibm,c=us that have H as the first letter.

Example 13-1 ldapadduuids utility example

```
ldapadduuids -D "cn=ldap administrator" -w password -b "o=ibm,c=us" "cn=H*"
```

13.1.4 Modify DN operation

The Modify DN operation is now fully implemented, as defined in RFC 2251. It also provides for co-existence with previous releases, to whatever extent possible. It has a implementation of renaming for both leaf and non-leaf nodes, subtree relocation, and DN realignment, with support for replication of these operations to z/OS replica servers (the previous release provided support for leaf renames only).

Existing client utilities (including non-IBM utilities) continue to work with the new implementation, if they worked with the old implementation. LDAP APIs exploit the new function. A new ldapmodrdn client utility is provided for user interaction with the new features (requests for non-leaf renames, subtree moves, and submission of DN realignment and timelimit controls).

There is no need to migrate change data created by previous releases. Once an z/OS V1R4 LDAP server has stored change data in a given DB2 database instance, prior server versions may no longer use that database. If one or more replica servers cannot be determined to contain the requisite support for the new features, then Modify DN operations are disabled in the master server for rename of non-leaf nodes, subtree relocations, and for requests accompanied by timelimit or realignment controls.

Performance considerations

Note that the Modify DN operation can potentially modify many (or all) directory entries in one transaction, and its magnitude of changes may reduce concurrency with other operations due to DB2 locking. DB2 logging is done for all changed data, so performance may suffer when many entries are affected.

Its transactional nature means Modify DN operation is all-or-none; for example, if 99,999 entries are modified and #100,000 fails for some reason, *all* modifications are discarded. Therefore, the time needed to back out in-flight changes affects concurrency with other operations.

13.1.5 Access control list (ACL) enhancement

This enhancement provides attribute-level access control; it also provides grant/deny control for access classes and attributes.

The aclEntry attribute now has expanded syntax. Figure 13-2 shows the prior aclEntry attribute value syntax.

```
[access-id: | group: | role: ] <subjectDN> [ ":" <access-class> ":" [ permissions ] ]
```

Figure 13-2 Prior aclEntry attribute value syntax

Figure 13-3 shows the new aclEntry attribute value syntax.

```
[access-id: | group: | role: ] <subjectDN> [ ":" <access-class | "at."<attributetype> >
[ ":" < "grant" | "deny" > ] ":" [ permissions ] ]
```

Figure 13-3 New aclEntry attribute value syntax

Figure 13-4 shows an example of an aclEntry value that gives Tim read, write, search, and compare on “normal” attributes; denies all access to “sensitive” attributes, and grants write access only to the userpassword attribute. .

```
aclEntry:cn=tim,o=tworld:normal:rwsc:sensitive:deny:rwsc:at.userpassword:w
```

Figure 13-4 aclEntry attribute value

Migration considerations

This support matches capabilities in the IBM Directory Server, so it can be replicated between the z/OS LDAP Server and IBM NT/Unix LDAP servers.

Note: When running in a sysplex, all servers need to be at the z/OS V1R4 level before changing the database version. The co-existence PTF for OS/390 V2.10, z/OS V1R1, and z/OS V1R2 is required.

13.1.6 Activity logging

Activity logging provides a way to log server activity for load analysis. You can specify a log file to record selectable, time stamped server activity.

The logfile option in a configuration file can be either an HFS file or MVS data set. The default is /etc/ldap/gldlog.output.

Activity logging is invoked by the modify command, F jobname,appl=log,xxx, where xxx is one of the following:

writeops Logs start of add, delete, modify, modrdn, extop

allops writeops + search, compare

time Add ending log for all enabled ops logs

notime Stop writing ending logs

stop Stop writing any logs

flush Make sure log records are written to DASD

summary Write only accumulated statistics on an hourly basis

msgs Write messages to log

nomsgs WTORs writing messages to log

Figure 13-5 shows sample output from an activity log.

```
Fri Mar 22 15:14:28 2002 total operations started = 0
Fri Mar 22 15:14:28 2002 total operations completed = 0
Fri Mar 22 15:14:28 2002 total search entries sent = 0
Fri Mar 22 15:14:28 2002 total bytes sent = 0
Fri Mar 22 15:14:28 2002 total connections processed = 0
Fri Mar 22 15:14:28 2002 current connections = 0
Fri Mar 22 15:14:28 2002 connection high water mark = 0
Fri Mar 22 15:15:14 2002 Modify: connid = 1, DN = cn=schema, o=university of michigan,c=us
Fri Mar 22 15:15:35 2002 Add: connid = 2, DN = o=University of Michigan,c=US
Fri Mar 22 15:15:44 2002 Add: connid = 3, DN = cn=Karen Lang,o=university of michigan,c=us
Fri Mar 22 15:15:50 2002 Modify: connid = 4, DN = cn=Karen Gdaniec,o=university of michigan,c=US
Fri Mar 22 15:16:28 2002 Search: connid = 5, base = , filter = (objectclass=*)
Fri Mar 22 15:16:37 2002 Search: connid = 6, base = , filter = (objectclass=*)
Fri Mar 22 15:17:03 2002 Search: connid = 7, base = o=university of michigan,c=us, filter = (objectclass=*)
Fri Mar 22 15:17:15 2002 Modrdn: connid = 8, DN = cn=Karen Lang, o=university of michigan, c=US
Fri Mar 22 15:17:23 2002 Search: connid = 9, base = o=university of michigan,c=us, filter = (objectclass=*)
Fri Mar 22 15:18:26 2002 total operations started = 23
Fri Mar 22 15:18:26 2002 total operations completed = 23
Fri Mar 22 15:18:26 2002 total search entries sent = 5
Fri Mar 22 15:18:26 2002 total bytes sent = 1090
Fri Mar 22 15:18:26 2002 total connections processed = 9
Fri Mar 22 15:18:26 2002 current connections = 0
Fri Mar 22 15:18:26 2002 connection high water mark = 1
```

Figure 13-5 Activity log sample output

13.1.7 RDBM and JNDI removal

TDBM, which was introduced in OS/390 V2.10, is a higher-scalable back-end data store for LDAP than RDBM is. Regarding JNDI, Sun now provides JNDI in JDK, and the IBM and Sun JNDI implementations differ, which causes customer confusion. Therefore, RDBM and its related code and the IBM JNDI implementation are no longer shipped.

For RDBM, data must be migrated from RDBM to TDBM. For JNDI, you must use IBM's distribution of Sun's support; refer to *z/OS Security Server LDAP Server Administration and Use*, SC24-5923, for more information.

14

Security Server Network Authentication Service

This chapter describes the enhancements to the Security Network Authentication Service in z/OS Version 1 Release 4 as follows:

- IPv6 support
- Security registry support for Kerberos in the NDBM database

Archived

© Copyright IBM Corp. 2003

14.1 Security Server Network Authentication Service

Security Server Network Authentication Service was new in OS/390 V2R10. It is licensed with the base operating system and can be used without ordering or enabling Security Server. Prior to z/OS V1R2, this component was named Network Authentication and Privacy Service.

Security Server Network Authentication Service is a IBM z/OS program based on Kerberos Version 5. The Kerberos authentication system was developed at MIT in the 1980s. It is a trusted third-party authentication system whose main purpose is to allow clients and processes (principals) to prove their identity across an network without an interchange of secret information.

All entities in Kerberos must be registered with an authentication server. In other words, each must have a secret key (password) which is shared only with an authentication server (AS). In order to obtain services from an application server, a client (principal) must acquire a ticket from the AS. Then the client presents this ticket to the application server, which can now perform the verification or validation of the client's identity. For more information, see *z/OS Security Server Network Authentication Service Administration,* SC24-5926.

14.2 Enhancements in z/OS V1R4

Release 4 of Network Authentication Service principally provides two new items: IPv6 support, and support for the NDBM database. The Kerberos NDBM database uses UNIX System Services database support rather than RACF. For this reason, the database is stored in HFS files.

The advantage of using NDBM instead of RACF for the Kerberos database is that the NDBM database supports the Kerberos **kadmin** command for remote Kerberos administration and can be shared with non-z/OS Kerberos platforms (that is, the KDC can run on z/OS and non-z/OS platforms). If you use RACF for the Kerberos database, then the KDC runs only on z/OS and only RACF commands can be used to administer the Kerberos information.

14.2.1 IPv6 support

IPv6 is a new function for z/OS V1R4 Communication Server IP Services. An IPv6 address has 16 bytes, while an IPv4 address has 4 bytes. The socket definition for IPv4 is AF_INET, and for IPv6 it is AF_INET6. For more information about IPv6, see *z/OS Communication Server IPv6 Network and Application Design Guide,* SC31-8885.

There are no changes to installation or configuration in Network Authentication Service. But there are changes in Kerberos APIs and a GSS-API.

Change in Kerberos APIs

The krb5_address_compare API can have address lists that contain both IPv4 and IPv6 addresses. An IPv6 address that maps an IPv4 address is considered to be equal to the IPv4 address.

APIs that generate addresses

Generated address lists can contain both IPv4 and IPv6 addresses. An IPv6 address that represents a mapped IPv4 address will be generated as an IPv4 address.

APIs that return addresses

Returned address lists can contain both IPv4 and IPv6 addresses. An IPv6 address that represents a mapped IPv4 address will be returned as an IPv4 address.

APIs that store addresses

Address lists in tickets can contain both IPv4 and IPv6 addresses. A mapped IPv6 address will be stored in the ticket as the corresponding IPv4 address.

Changes to GSS-API

Channel bindings address type GSS_C_AF_INET6 is added for DARPA Version 6 Internet address (IPv6), while GSS_C_AF_INET is for DARPA Version 4 Internet address (IPv4).

14.2.2 New support for Kerberos registry in NDBM

The Kerberos security server now supports two types of security registries: SAF and NDBM. The NDBM registry is new in z/OS V1R4 and is a Kerberos registry in the HFS. It can be used instead of an SAF (RACF) registry.

NDBM registry database

The key distribution center (KDC) maintains its own registry database using UNIX System Services. Full Kerberos administration support is provided in the NDBM registry, and the realm can contain both z/OS KDC and non-z/OS KDC instances.

The NDBM registry uses the database support provided by UNIX System Services. The database files are located in the /var/skrb/krb5kdcdirectory. Kerberos database propagation is used to synchronize these files between systems in the same sysplex, and between systems in different sysplexes.The file system containing the /var/skrb/krb5kdc directory must be large enough to contain two copies of the registry database files, plus a complete database dump file.

SAF registry database

Kerberos information is integrated with the z/OS system authorization profiles. The SAF (RACF) registry database can be shared within the sysplex. System-authenticated userids can eliminate the use of Kerberos passwords and key tables when obtaining and decrypting tickets. And it has scales to support a large number of principals.

Kerberos principals must be mapped to system userids. No Kerberos administration support is provided due to semantic differences between the SAF database and the Kerberos administration wire protocols. All KDC instances in the realm must share the same SAF database.

Configuration and administration for NDBM registry

The **krb5_ndbm** command is used to create the initial registry database files.

The Kerberos security server supports two database propagation protocols: full replacement, and individual updates. The full replacement protocol sends the entire Kerberos database to each secondary Kerberos security server. This is the only propagation protocol supported by MIT Kerberos.

The propagation occurs at timed intervals specified by the SKDC_KPROP_INTERVAL environment variable. A propagation does *not* occur if there have been no changes to the database since the last database propagation.

The NDBM registry database is not shared by each security server in the realm (the file system containing the /var/skrb/krb5kdc directory must not be shared between systems). Instead, each security server maintains its own NDBM database and receives updates from the primary security server through the database propagation protocol.

The same SAF registry database must be shared by *all* of the security servers in the realm. This means you do not need to repeat the RACF (or other external security manager) commands previously described when you configure a secondary security server. Database propagation is not used by the Kerberos security server for a SAF registry database, since the external security manager is responsible for any required propagation.

If the /etc/skrb file system is not shared between systems, copy the/etc/skrb/krb5.conf and /etc/skrb/home/kdc/envar files from the primary system to the secondary system. You do not need the /etc/skrb/home/kdc/kadm5.acl configuration file because Kerberos administration services are not available for the SAF registry database.

For additional information, refer to *z/OS Network Authentication Service Administration*, SC24-5926.

Archived

15

Cryptographic services

This chapter describes the enhancements to cryptographic services in z/OS Version 1 Release 3 as follows:

- ICSF TSO panel enhancement
- Unique key per transaction (UKPT) and PKCS#1V2 support
- Advanced Encryption Standard (AES) support
- ICSF sample JCL in SYS1.SAMPLIB
- CSFEUTIL utility enhancement
- Hardware requirements for ICSF V1R3

It also describes enhancements to System SSL in z/OS Version 1 Release 4 as follows:

- gskkyman utility
- Certificates with private keys in ICSF
- Advanced encryption standard (AES) support
- IPv6 network address support
- Performance enhancements Session ID caching in a sysplex
- Serviceability enhancements
- GSKSRVR started task changes

© Copyright IBM Corp. 2003

15.1 Cryptographic services components

Cryptographic services consists of the following components:

Integrated Cryptographic Service Facility (ICSF)	Changed in z/OS V1R3.
Open Cryptographic Services Facility (OCSF)	Last changed in z/OS V1R2.
System Secure Sockets Layer (SSL)	Changed in z/OS V1R4.

15.2 ICSF enhancements in z/OS V1R3

The ICSF TSO panels have been updated to enhance usability. ICSF provides enhanced support for Unique Key Per Transaction (UKPT) key derivation for PIN services and PKCS #1V2 OAEP block formatting for key management services. Also, ICSF provides services to perform encryption using the AES algorithm.

ICSF RMF is enhanced for DES and SHA-1 instruction/service usage.

15.2.1 TSO panel enhancement

The ICSF TSO panels are changed to enhance usability, as follows:

- Coprocessor management functions are combined onto one panel
- Master key management and CKDS functions are combined onto one panel
- TKE TSO utilities are combined onto one panel
- The primary panel is simplified
- There is a new utility to generate master key values from a pass phrase

15.2.2 Unique key per transaction (UKPT) and PKCS#1V2 support

ICSF provides enhanced support for unique key per transaction (UKPT) key derivation for PIN services and support for PKCS #1V2 OAEP block formatting for key management services.

In z/OS V1R3, the rule_array parameter is changed for the PIN services, as follows:

- Encrypted PIN Translate (CSNBPTR) has been enhanced to support UKPT keywords KPTIPIN, UKPTOPIN, and UKPTBOTH.
- Encrypted PIN Verify (CSNBPVR) has been enhanced to support the UKPT keywordUKPTIPIN.

The RSA PKCS #1V2 OAEP has a new rule_array keyword, PKCSOAEP, for key export in z/OS V1R3 as follows:

- Symmetric Key Export (CSNDSYX). APAR OW50507 is available on HCR7703 (OS/390 V2 R10 and z/OS V1R1) and HCR7704 (z/OSV1R2).
- Symmetric Key Export (CSNDSYX). APAR OW50507 is available on HCR7703 (OS/390 V2R10 and z/OS V1R1) and HCR7704 (z/OS V1R2).
- Symmetric Key Import (CSNDSYI). APAR OW50507 is available on HCR7703 (OS/390 V2R10 and z/OS V1R1)and HCR7704 (z/OS V1R2).

Other callable services enhanced in z/OS V1R3 are:

- Symmetric Key Decipher (CSNBSYD). This callable service deciphers data in an address space or a dataspace using the cipher block chaining or electronic code book modes. The

Symmetric Key Decipher service (AES support) is also available on z/OS V1R2 through APAR OW51349 (refer to 15.2.3, “Advanced Encryption Standard (AES) support” on page 243 for more information).

- Symmetric Key Encipher (CSNBSYE). This callable service enciphers data in an address space or a dataspace using the cipher block chaining or electronic code book modes. The Symmetric Key Encipher service (AES support) is also available on z/OS V1R2 through APAR OW51349.

15.2.3 Advanced Encryption Standard (AES) support

ICSF provides services to perform encryption using the AES algorithm. Only clear key support will be provided. Services provided are symmetric key decipher and symmetric key encipher.

The AES algorithm is implemented in software. System availability will be the same as triple DES. Clear keys of 128-bit, 192-bit, and 256-bit lengths are supported, and CBC and ECB encryption modes are supported.

Symmetric key decipher callable service (CSNBSYD) decrypts a text block using the AES algorithm and a clear key. Symmetric key encipher callable service (CSNBSYE) encrypts a text block using the AES algorithm and a clear key.

15.2.4 ICSF setup and CSFEUTIL utility enhancements

Batch process setup for ICSF has been added. Sample JCL has been added to SYS1.SAMPLIB.

A new CSFEUTIL option for pass phrase initialization is introduced. CSFUTIL now can be used to initialize DES and PKA master keys using a pass phrase. Parameters which are need for this option are CKDS name, optional pass phrase, and PPINIT keyword; see Example 15-1.

Example 15-1 New CSFEUTIL option

```
//STEP EXEC PGM=CSFEUTIL, PARM='CSF.CKDS,PPINIT'
//STEP EXEC PGM=CSFEUTIL, PARM='CSF.CKDS,This is my pass phrase, PPINIT'
```

15.2.5 Hardware requirements

z/OS V1R3 ICSF executes on all systems with a CCF, or on all systems with both CCFs and PCICCs/PCICAs. A PCICC (with appropriate LIC level) is required for the following new function in z/OS V1R3 ICSF:

- UKPT support
- PKCS #1V2 support

15.3 System SSL enhancements in z/OS V1R4

In Release 4, the following functional and performance enhancements have been made to System SSL:

- The gskkyman utility has been restructured to allow for clearer presentation of certificate information. It has also been enhanced to support exporting/importing certificates in PKCS #12 Version 3 and PKCS #7 format, as well as modification of certificate labels and creation of Digital Signature Standard certificates (FIPS 186-1).

- In addition to the APIs being provided so that applications can securely communicate over an open communication network using the SSL or TLS protocols, a new suite of APIs has been introduced to allow application writers the ability to exploit functions (other than typical SSL functions), including:
  - The ability to create/manage key database files in a way similar to the SSL gskkyman utility.
  - Use certificates stored in the key database file or key ring for purposes other than SSL.
  - Basic PKCS #7 message support has been added to provide application writers a mechanism to communicate with another application through the PKCS #7 standard. These APIs build and process PKCS #7 messages.
- Key ring support has been enhanced to allow private keys to be stored in ICSF and applications to use key rings owned by other userids.
- System SSL has added AES cipher support to its SSL V3.0 and TLS V1.0 implementations. In order to exploit the AES ciphers, Security Level 3 Feature of System SSL is required.
- Support for IPv6 network addresses has been added.
- An in-storage caching mechanism has been added where retrieved Certificate Revocation Lists (CRLs) will be cached for a period of time. This will optimize the fetching done to retrieve CRL information from the LDAP server during certificate validation.

  A sysplex session cache has been added to make SSL server session information available across the sysplex. An SSL session established with a server on one system in the sysplex can be resumed using a server on another system in the sysplex, as long as the SSL client presents the session identifier obtained for the first session when initiating the second session. The sysplex session cache can be used to store SSL V3.0 and TLS V1.0 server session information.
- Component trace and enhanced debug granularity of trace information has been added.

15.3.1 System SSL gskkyman utility

gskkyman is a z/OS shell-based program that creates, fills in, and manages a z/OS HFS file that contains PKI private keys, certificate requests, and certificates. This z/OS HFS file is called a key database and, by convention, has a file extension of .kdb.

The **gskkyman** command was first introduced in OS/390 V2R7 as a method for doing certificate management. As part of the work being done in V1R4, the command is being restructured and enhanced. The gskkyman continues to be a interactive dialog command where the issuer select entries from a list and is prompted for information to perform the task.

Functional enhancements for gskkyman are as follows:

- Support for the government FIPS 186-1 standard was added for Digital Signature Standard certificates.
- Prior to V1R4, the PKCS #12 support in System SSL was based on a draft version (PFX 1.0). This level is extremely backlevel with the current level used by the industry. Most of the industry will not work with the PFX level. gskkyman has been enhanced to support the industry standard, as well as to continue to support the PFX version, to ensure interoperability among the OS/390 and z/OS releases.
- Support has been added to import and export certificates in the PKCS #7 format. Current implementation supports a single certificate (no certificate chains). CA certificates must be in the key database file prior to the PKCS #7 certificate being imported.

- Support for changing certificate labels. Certificate labels are the unique identifiers for certificates within the key database file. Changing labels becomes very helpful when changing the purpose of the certificate or providing more meaningful names when certificates were received from a PKCS#12 formatted file that contained more than one certificate.
- Certificate creation supports certificates with RSA key sizes between 512 and 1024, and a DSA key size between 523 and 2048. Existing certificates with key sizes of 512 will continue to be supported, as well as the importation of certificates with key sizes of 512.
- Certificate creation supports X.509 Version 3 certificates only. Existing Version 1 and Version 2 certificates will continue to be supported, as well as the importation of certificates based on version 1 and 2.
- gskkyman no longer supports the migration of mkkf certificate files to key database files. mkkf files have not been used since OS/390 Release 7.
- When creating self-signed certificates, the Key Usage extension identifies how the certificate can be used. In prior releases, this extension was never created. In V1R4, new certificates will take advantage of this extension. When creating a self-signed certificate to be used by a client or server application, the certificate must be created as an end user certificate. This will allow it to do key encipherment and digital signature. When the certificate is to act as a signing certificate authority, it must be created as a CA certificate. This will update the Key Usage with CRL and signing capabilities.

15.3.2 Certificates with private keys in ICSF

Today, certificates created in RACF can either have their private keys stored in the RACF or ICSF. System SSL was originally written to work only with keys not stored in ICSF. As of December 2001, support has been retrofitted back to OS/390 V2.10 and z/OS V1R2 through APAR OW52700. This support is in the base for z/OS V1R4.

15.3.3 Shared SAF key rings

In z/OS V1R4, support has been added to allow applications to work with key rings that are not owned by the user ID executing the application. This is done by specifying the userid/keyring together, either through the GSK_KEYFILE environment variable or the gsk_attribute_set_buffer API prior to initializing the SSL environment. The user of the keyring needs to have UPDATE authority to the RACF IRR.DIGTCERT facility class. Certificates that do not belong to the user can only be used for certificate validation, because the private key is not returned when the certificate is read from the key ring.

15.3.4 Advanced Encryption Standard (AES) support

System SSL has been enhanced to support the Advanced Encryption Standard (AES). The new AES provides a better combination of security and speed than DES. Using 128-bit secret keys, AES offers higher security against a brute force attack than the old 56-bit DES keys, and AES can use larger 256-bit keys.

In order to exploit the AES, the System SSL security level 3 feature must be installed. System SSL AES support is provided through a software implementation. New cipher specification values have been added to the SSL V3 and TLS V1.0 ciphersuites, “2F” for 128-bit and “35” for 256-bit. Cipher specification values are specified through the gsk_attribute_set_buffer API.

15.3.5 IPv6 Network Address support

The goal for System SSL in regard to IPv6 Network Addresses is to allow SSL applications to exploit both IPv4 and IPv6 addresses. System SSL applications are responsible for creating and maintaining the sockets. Therefore. System SSL contains sensitivity to network addresses when building its in-storage session id cache and when getting the peer name through the default-supplied getpeername function.

The default getpeername function has been updated to be sensitive to whether the “getpeername” function has returned an AF_INET or AF_INET6 structure. And session id cache entries have been enlarged to accommodate IPv6 addresses.

15.3.6 Performance enhancements

Certificate Revocation Lists (CRL) provide a mechanism to revoke a certificate prior to the certificate's expiration time. System SSL supports CRLs that are stored in an LDAP directory. In prior releases of System SSL, when the application has specified that it wants certificate validation to include CRL checking, the code would bind to the LDAP server (if not already done and connection had not been lost) and request a list of CRLs associated with the certificate being validated. This is very expensive and adds a great deal of time to the validation process.

In z/OS 1R4, a storage caching mechanism is provided where retrieved CRLs will be cached for a period of time. This time period is a configurable value. By default, the cache time is 24 hours.

15.3.7 Session ID caching across a sysplex

Sysplex session cache support makes SSL server session information available across the sysplex. An SSL session established with a server on one system in the sysplex can be resumed using a server on another system in the sysplex as long as the SSL client presents the session identifier obtained for the first session when initiating the second session. SSL V3 and TLS V1 server session information can be stored in the sysplex session cache, while SSL V2 server session information and all client session information is stored only in the SSL cache for the application process.

This support is implemented through the GSKSRVR started task. GSKSRVR server handles creating, retrieving and deleting SSL session information (SSL handshake).

In order to use the sysplex session cache, each system in the sysplex must be using the same external security manager (for example, z/OS Security Server RACF), and a userid on one system in the sysplex must represent the same user on all other systems in the sysplex. The external security manager must support the RACROUTE REQUEST=EXTRACT,TYPE=ENVRXTR and RACROUTE REQUEST=FASTAUTH functions.

The sysplex session cache must be enabled for each application server that is to use the support. This can be done by defining the GSK_SYSPLEX_SIDCACHE environment variable or by calling the gsk_attribute_set_enum() routine to set the GSK_SYSPLEX_SIDCACHE attribute. The session information for each new SSL V3 or TLS V1 session created by the SSL server will then be stored in the sysplex session cache and can be referenced by other SSL servers in the sysplex.

15.3.8 Serviceability enhancements

For debug tracing, the GSK_TRACE_FILE and GSK_TRACE environment variables must be set prior to the start of the SSL application. It stays in effect until the application is terminated. The trace file produced is unformatted. The **gsktrace** command formats the data into readable a format. GSK_TRACE allows you to tailor the type of data to be trace.

GSK_TRACE_FILE Specifies the name of the trace file and defaults to /tmp/gskssl.%.trc. The gsktrace command is used to format the trace file. The trace file is not used if the GSK_TRACE environment variable is not defined or is set to 0. The current process identifier is included as part of the trace file name when the name contains a percent sign (%). For example, if GSK_TRACE_FILE is set to /tmp/gskssl.%.trc and the current process identifier is 247, then the trace file name will be /tmp/gskssl.247.trc.

GSK_TRACE Specifies a bit mask enabling System SSL trace options. No trace option is enabled if the bit mask is 0, and all trace options are enabled if the bit mask is 0xffff. The bit mask can be specified as a decimal (nnn), octal (0nnnn), or hexadecimal (0xhh) value.

GSKSRVR started task

The GSKSRVR started task is not required to be configured and active unless component trace or sysplex session id caching is requested. When being used for the component tracing function, the started task must be active on the system prior to any System SSL applications being started. The tracing capability of the started task does not have to be activated until tracing is required.

The component trace can be started before the job to be traced is started or while the job is running. The trace will be active for the first instance of the job. For example, if the same job name is used for multiple jobs, only the first job with that name will be traced. Subsequent jobs with the same name will not be traced unless the component trace is stopped and then restarted.

Tracing is turned on using the MVS TRACE command. Trace data can be tailored through the TRACE command's TT parameter specifying the LVL keyword.

Modify GSKSRVR command

The GSKSRVR started task supports several modify operator subcommands. The main subcommands are:

DISPLAY CRYPTO Displays the available encryption algorithms, whether hardware cryptographic support is available, and the maximum encryption key size; see Figure 15-1. The figure shows that DES and 3DES are available and will be used by System SSL.

```
GSK01009I Cryptographic status
Algorithm       Hardware    Level
RC2                No        128-bit
RC4                No        128-bit
DES                Yes       56-bit
3DES               Yes       168-bit
AES                No        256-bit
RSA                No        2048-bit
DSS                No        1024-bit
```

Figure 15-1 DISPLAY CRYPTO

DISPLAY LEVEL Displays the current System SSL service level. Figure 15-2 shows DISPLAY LEVEL response message.

```
GSK01001I System SSL version 3.14, Service level 0Wxxxxxx
```

Figure 15-2 DISPLAY current System SSL level

Archived

16

Parallel Sysplex enhancements

This chapter provides an overview of the Parallel Sysplex enhancements that were introduced in z/OS V1R3 and in z/OS V1R4.

Parallel Sysplex enhancements in z/OS V1R3

In this release many components and subsystems including BCP, USS, and DFSMS functions are enhanced as follows:

- For WLM sysplex enhancements, refer to sections:
  - 7.1, “Removal of WLM compatibility mode” on page 88
  - 7.2, “PAV dynamic alias management for paging devices” on page 99
  - 7.4, “WLM support for sub-capacity pricing” on page 104
  - 7.5, “WLM enqueue management enhancements” on page 105
  - 7.6, “WLM WebSphere performance enhancement” on page 107
- For BCP sysplex enhancements, refer to sections:
  - 16.1.1, “System Logger enhanced logstream attribute support” on page 251
  - 16.1.2, “GRS enhancements” on page 259
- For USS sysplex enhancements, refer to sections:
  - 8.3.2, “Shutting down z/OS UNIX” on page 156
  - 16.1.4, “Sysplex mount table limit monitoring” on page 261
  - 16.1.5, “Sysplex mount/unmount performance improvements” on page 261
- For DFSMS sysplex enhancements, refer to sections:
  - 16.1.6, “DFSMShsm™ common recall queue (CRQ)” on page 262
  - 16.1.7, “Caching of larger than 4 KB CIs in VSAM RLS cache CF structure” on page 264
  - 16.1.8, “VSAM RLS lock structure duplexing enhancement” on page 266
  - 16.1.9, “DFSMS enforced data set separation for high availability” on page 267
  - 16.1.10, “OAM multiple object backup” on page 268

© Copyright IBM Corp. 2003

Parallel Sysplex enhancements in z/OS V1R4

In this release many components and subsystems such as BCP, USS, DFSMS and Communications Server functions are enhanced as follows:

- For WLM sysplex enhancements, refer to sections:
  - 7.7, “WLM batch initiator balancing enhancements” on page 107
  - 7.8, “Performance block application state reporting for enclaves” on page 113
  - , “Response time breakdown RMF reporting enhancements” on page 115
  - 7.9, “WLM msys for Setup enhancement” on page 116
  - 7.10, “WLM support for ESS FICON and I/O priority management” on page 116
- For BCP sysplex enhancements, refer to sections:
  - 5.1.1, “JES3 MAINPROC refresh” on page 50
  - 16.2.1, “XES DB2 Data sharing performance improvement” on page 271
  - 16.2.2, “XES CFRM performance enhancements” on page 271
  - 16.2.3, “RRS multisystem cascaded transaction enhancement” on page 273
  - 16.2.4, “System Logger offload monitor function” on page 274
- For USS sysplex enhancements, refer to sections:
  - 9.1.1, “Automove system list specification” on page 174
  - 9.2, “Byte-range locking in a shared HFS environment” on page 178
  - 9.3, “Shared HFS availability enhancement” on page 179

Hardware, CFCC and other sysplex enhancements

These recent sysplex enhancements do not relate to a specific z/OS release:

- 16.3.1, “Cascaded FICON director switch support” on page 278
- 16.3.2, “CF Request Time Ordering (Sysplex Timer connectivity to CFs)” on page 278
- 16.3.3, “Enhanced Parallel Sysplex support in CFLEVEL 11 and CFLEVEL12” on page 283
- 16.3.4, “zSeries GDPS/PPRC hyperswap function” on page 284

Archived

16.1 Parallel Sysplex enhancements for z/OS V1R3

Let us review the sysplex enhancements introduced in z/OS V1R3.

16.1.1 System Logger enhanced logstream attribute support

Prior to this enhancement, updating or redefining your logstream attributes was a very restrictive (and in most cases, disruptive) process. A number of problem areas have been addressed with this enhancement, including the following:

- Most logstream attribute updates are not allowed if there was any type of logstream connection, irrespective of whether it is active or “failed-persistent”. Attempts to update logstream attributes led to disruption of any workload activity that required the use of that particular logstream.
- System Logger structure definitions defined in CFRM and stored in the LOGR Couple Data Set were difficult to manipulate. Prior to z/OS V1R3, changing such definitions was a very cumbersome process.
- The naming convention for the System Logger logstream offload and staging data sets were limited by the HLQ parameter that only allowed one high level qualifier.

Let us look at how each of these problems are addressed by this enhancement:

- Dynamic logstream attribute update support
- Dynamic System Logger structure definition updates
- Extended high level qualifier support for System Logger

Dynamic logstream attribute update

Logstream offload and logstream staging data sets are single extent VSAM linear data sets (shareoptions “3,3”). We can look at an example of what it took to change a logstream LS_DATACLAS value prior to this enhancement. Let us assume that we needed to increase the size of this value because, over time, we started to experience frequent DASD shifts associated with allocation of new offload data sets, which tended to have a negative impact on performance on your system. From a high level perspective, these were the steps you need to take:

1. Stop all subsystems or applications that are using the logstream.
2. Make sure they successfully disconnect from the logstream.
3. Update the LS_DATACLAS value.
4. Restart or resume all subsystems or applications.
5. Check that the new value is in effect for new offload data set processing.
6. Revert the whole process if you need to change back to the previous setting.

As an enhancement, logstream pending updates are allowed for offload and connection-based logstream attributes in z/OS V1R3 and higher systems. Most logstream attributes can be updated, even those with an existing connection.

Your pending updates will be committed at different points for each logstream attribute, as follows:

- The following are committed only on the subsequent first connection (or last disconnection) to the logstream in the sysplex:
  - Structure-based logstream attributes (**STG_DUPLEX,DUPLEXMODE,LOGGERDUPLEX**)

- DASD-only logstream attribute (**MAXBUFSIZE**)

- The following are committed on the subsequent first connection (or last disconnection) to the logstream in the sysplex, and on the next structure rebuild (for structure-based), which is used for example, when a new staging data set needs to be allocated:
  - (**STG_DATACLAS, STG_MGMTCLAS, STG_STORCLAS, STG_SIZE**)
- The following are committed on the subsequent first connection (or last disconnection) to the logstream in the sysplex, and on the next structure rebuild (for structure-based), which is used when a new staging data set needs to be allocated. In addition it is also committed at next offload data set switch, which is used when a new offload data set is allocated:
  - (**LS_DATACLAS, LS_MGMTCLAS, LS_STORCLAS, LS_SIZE**)
  - (**LOWOFFLOAD, HIGHOFFLOAD, OFFLOADRECALL**) also committed on next offload data set switch, in the case of a DASD-only logstream
- The following is committed on the subsequent first connection (or last disconnection) to the logstream in the sysplex, and on the next offload data set switch, which is used when a new offload data set is allocated:
  - (**RETPD,AUTODELETE**) which was previously only committed on the next offload data set switch

For a list of logstream attributes that cannot be updated dynamically, refer to “Dynamic System Logger structure definition updates” on page 255.

For an overview of when the dynamic updates are committed for the System Logger logstream attributes, refer to Table 16-1.

Table 16-1 System Logger logstream attribute dynamic update commit outline

Logstream attribute	Last disconnect or first connect to logstream in sysplex	Switch to New offload Data Set	CF Structure Rebuild
RETPD	yes	yes	no
AUTODELETE	yes	yes	no
LS_SIZE	yes	yes	yes
LS_DATACLAS	yes	yes	yes
LS_MGMTCLAS	yes	yes	yes
LS_STORCLAS	yes	yes	yes
OFFLOADRECALL	yes	yes (1)	yes
LOWOFFLOAD	yes	yes (1)	yes
HIGHOFFLOAD	yes	yes (1)	yes
STG_SIZE	yes	no	yes
STG_DATCLAS	yes	no	yes
STG_MGMTCLAS	yes	no	yes
STG_STORCLAS	yes	no	yes
STG_DUPLEX (cf)	yes	no	no
DUPLEXMODE (cf)	yes	no	no

Logstream attribute	Last disconnect or first connect to logstream in sysplex	Switch to New offload Data Set	CF Structure Rebuild
LOGGERDUPLEX (cf)	yes	no	no
MAXBUFSIZE (do)	yes	no	n/a
Notes: 1 - These attributes are only committed during switch to new offload data set activity for DASD-only logstreams. These attributes are not committed at this point for CF structure-based logstreams. yes - Indicates the attribute is committed during the activity listed in the column heading. no - Indicates the attribute is not committed during the activity. (cf) - Indicates the attribute is only applicable to CF structure-based logstreams. (do) - Indicates the attribute is only applicable to DASD-only-based logstreams.			

Certain structure attributes, such as the **AVGBUFSIZE** value, cannot be dynamically updated.

However, the dynamic entry-to-element ratio processing introduced back in OS/390 V1R3 helps reduce the need to change this value, since the System Logger will dynamically change this ratio based on actual usage of the structure. The support to allow a logstream to be moved to a new structure, as discussed in “Dynamic System Logger structure definition updates” on page 255, also reduces such needs.

System Logger logstream attributes

For an overview of all System Logger logstream attributes, and whether they can be dynamically updated, refer to Table 16-2 on page 254.

The following is an explanation of the columns in Table 16-2:

- The first column contains the logstream attributes.
- The second column indicates whether the attribute can be specified on an update request.
- The third and fourth columns fall under the general heading “enhanced dynamic updates not enabled”. This means that either z/OS V1R2 or a lower level release is used, or any z/OS release when a LOGR CDS at HBB5520 or HBB6603 format level is used, which in turn indicates formatting option **ITEM(SMDUPLEX)NUMBER(0)**:
- The fifth and sixth columns fall under the general heading “enhanced dynamic updates enabled”. This means that either z/OS V1R3 or a higher release level is used when using a LOGR CDS at HBB7705 format level, which in turn indicates formatting option **ITEM(SMDUPLEX) NUMBER(1)**:

The “no connections” columns (3 and 5) mean there are no current active nor failed connections to the logstream in the sysplex.

The “connections exist” columns (4 and 6) mean there is at least one current active or failed connection to the logstream in the sysplex.

A request to update an attribute may be one of the following:

- “n/a” (not applicable)
- “rejected”
- “committed” update status (immediately)
- “pending” update status (to be committed later)

Table 16-2 System Logger logstream attribute dynamic update capability outline

Logstream attribute	Attribute allowed on update logstream	Enhanced dynamic updates not enabled - no connections exist	Enhanced dynamic updates not enabled - connections exist	Enhanced dynamic updates enabled - no connections exist	Enhanced dynamic updates enabled - connections exist
STREAMNAME	no	n/a	n/a	n/a	n/a
DASDONLY	no	n/a	n/a	n/a	n/a
HLQ	no	n/a	n/a	n/a	n/a
EHLQ	no	n/a	n/a	n/a	n/a
MODEL	no	n/a	n/a	n/a	n/a
LIKE	no	n/a	n/a	n/a	n/a
RMNAME	yes	committed	rejected	committed	rejected
DESCRIPTION	yes	committed	rejected	committed	rejected
STRUCTNAME	yes	committed	rejected	committed	rejected
DIAG	yes	committed	committed	committed	committed
RETPD	yes	committed	pending	committed	pending
AUTODELETE	yes	committed	pending	committed	pending
OFFLOADRECALL	yes	committed	pending	committed	pending
LS_DATACLAS	yes	committed	rejected	committed	pending
LS_MGMTCLAS	yes	committed	rejected	committed	pending
LS_STORCLAS	yes	committed	rejected	committed	pending
LS_SIZE	yes	committed	rejected	committed	pending
STG_DATACLAS	yes	committed	rejected	committed	pending
STG_MGMTCLAS	yes	committed	rejected	committed	pending
STG_STORCLAs	yes	committed	rejected	committed	pending
STG_SIZE	yes	committed	rejected	committed	pending
LOWOFFLOAD	yes	committed	rejected	committed	pending
HIGHOFFLOAD	yes	committed	rejected	committed	pending
STG_DUPLEX	yes	committed	rejected	committed	pending
DUPLEXMODE	yes	committed	rejected	committed	pending
LOGGERDUPLEx	yes	committed	rejected	committed	pending
MAXBUFSIZE	yes	committed	rejected	committed	pending

Archived

Dynamic System Logger structure definition updates

We can look at an example of what it took to change a System Logger CF structure prior to this enhancement. Let us assume that we needed to rebalance logstreams across a set of CF structures because we have experienced suspension of log writes (based on SMF Type 88 Subtype 11 record evidence). Again from a high level perspective, these are the steps you took:

1. Stop or quiesce subsystems or applications.
2. Check that they successfully disconnect from the logstream.
3. When there are no connections to the logstream in your sysplex, delete the logstream from the LOGR CDS.
4. Re-define the logstream in the LOGR CDS.
5. Use a new structure name and ensure it is also defined in the CFRM policy.
6. Activate the CFRM policy.
7. Restart or resume subsystems or applications so they can start to use the new (empty) instance of the logstream using the newly defined CF structure.
8. Use a similar (but reversed) disruptive procedure if the logstream needs to be changed back to use the previous CF structure.

Introduced in z/OS V1R3 is the ability to allow a logstream definition to be updated to use a different System Logger CF structure. You may re-map a logstream to use a different CF System Logger structure without having to first delete the logstream.

Important: This support greatly reduces the need to provide specific System Logger CF structure attributes updates, since the affected logstreams can be re-mapped to another structure with the desired attributes.

Notice that the logstream cannot have any outstanding connections (active or “failed-persistent”) in the sysplex in order for this update to be honored.

Note: The following System Logger logstream attributes can be specified when the structure is defined, but none of them can be updated:

- STRUCTNAME
- LOGSNUM
- AVGBUFSIZE
- MAXBUFSIZE

Extended high level qualifier support for System Logger

Prior to z/OS V1R3, you could specify a high level qualifier for data set names for logstream definitions used for both offload data sets as well as staging data sets. However, your **HLQ** value was limited to 1 to 8 characters. Only one qualifier and no periods was allowed. The default value for the high level qualifier is **IXGLOGR**. These restrictions did not always lend themselves very well to your data set naming convention (note that CICS does not use the high level qualifier HLQ for its logstream offload data set names).

Introduced in z/OS V1R3 is an “extended high level qualifier” for logstream data sets called **EHLQ**. **EHLQ** is very similar to the **HLQ** parameter, but provides added flexibility. Note that **EHLQ** and **HLQ** are mutually exclusive parameters. **EHLQ** syntax requirements are as follows:

- The extended highlevel qualifier must be 33 alphanumeric or national ($, #, or @) characters, padded on the right with blanks if necessary.

- The value can be made up of one or more qualifiers (each 1 to 8 characters) separated by periods, up to the maximum length of 33 characters.
- Each qualifier must contain up to eight alphabetic, national, or numeric characters.
- Lowercase alphabetic characters will be folded to uppercase.
- The first character of each qualifier must be an alphabetic or national character.
- Each qualifier must be separated by a period, which you must count as a character.
- The resulting length of concatenating the significant characters from the **EHLQ** value with the **STREAMNAME** value (including the period delimiter) cannot exceed 35 characters.

Activating enhanced logstream attribute support

To activate the enhanced logstream attribute support you need to do the following:

- Ensure all the systems in your sysplex are at the z/OS V1R3 level. Also refer to "Migration, coexistence and fallback considerations" on page 257.
- Perform LOGR CDS formatting and policy (inventory) updates.
- Format a new level (HBB7705) of the LOGR CDS (including primary, alternate, and spares for each).
- Bring the new LOGR CDSs into your sysplex.
- Define/Update the System Logger logstreams with your preferred options.

How to format LOGR CDS at level HBB7705

SMDUPLEX is an optional input keyword for the LOGR CDS format utility (IXCL1DSU). **SMDUPLEX** indicates your System Logger functional level, such as XES system-managed structure duplexing, as well as enhanced logstream attribute support.

> **Attention:** Once LOGR (and CFRM) CDSs are formatted with the SMDUPLEX keyword, no systems lower than Z/OS V1R2 are able to use those CDSs. Fallback to LOGR (or CFRM) CDSs that do not have this support requires a sysplex-wide IPL.

The specification of **SMDUPLEX** affects the LOGR couple data set format level. The value on the **NUMBER** parameter indicates one of the following:

0 Logger does not support system-managed structure duplexing. Specifying this number results in a LOGR CDS format level of HBB6603.

1 Logger supports system-managed structure duplexing and accepts enhanced logstream attribute updates. Specifying this number results in a LOGR CDS format level of HBB7705.

> **Note:** **SMDUPLEX** can only be specified for z/OS V1R2 and higher release levels.
>
> When **SMDUPLEX** is omitted from the IXCL1DSU format utility, then System Logger handles it the same as if a value of 0, which is the default value, was specified for the **NUMBER** parameter. This specification does cause any additional space in the LOGR CDS to be consumed.

An example of how these input values look in the IXCL1DSU utility is shown in Figure 16-1 on page 257.

```
//INVCDS    JOB                                                          e
//*                                                                      e
//* SAMPLE JCL TO FORMAT THE PRIMARY AND ALTERNATE                       e
//*   COUPLE DATA SETS FOR SYSTEM LOGGER (LOGR)                          e
//*                                                                      e
//* COUPLE DATA SET ALLOCATION RULES:                                    e
//*                                                                      e
//*   1. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS                  e
//*   2. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY       e
//*      MESSAGES                                                        e
//*   3. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY          e
//*      CONTROL STATEMENTS                                              e
//*                                                                      e
//*                                                                      e
//********************************************************************** e
..
//STEP2     EXEC  PGM=IXCL1DSU
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD    *
     DEFINEDS SYSPLEX(XLSDEV)
              DSN(SLC.FDSS12A) VOLSER(HENRI1)
          DATA TYPE(LOGR)
               ITEM NAME(LSR) NUMBER(10)
               ITEM NAME(LSTRR) NUMBER(10)
               ITEM NAME(DSEXTENT) NUMBER(20)
               ITEM NAME(SMDUPLEX) NUMBER(1)
     DEFINEDS SYSPLEX(XLSDEV)
              DSN(SLC.FDSS12B) VOLSER(HENRI2)
DATA TYPE(LOGR)
               ITEM NAME(SMDUPLEX) NUMBER(1)
               ITEM NAME(LSR) NUMBER(10)
               ITEM NAME(LSTRR) NUMBER(10)
               ITEM NAME(DSEXTENT) NUMBER(20)

/*
```

Figure 16-1 IXCLDSU Utility JCL to enable enhanced logstream attributes update capability

Migration, coexistence and fallback considerations

The migration, coexistence and fallback limitation considerations of using the z/OS V1R2 LOGR CDS format level are as follows:

- Only z/OS V1R2 and higher releases can be in the sysplex when the HBB7705 LOGR CDS is activated in your sysplex.
- You will be unable to bring in a lower level LOGR CDS as the alternate once the higher level LOGR CDS is used as the primary.
- If you attempt to IPL a system lower than V1R2 into the sysplex after the primary LOGR CDS format level is HBB7705, System Logger services on that system are unavailable.

Archived

Recommended System Logger updates: APAR OW48553 provides an enhancement to issue an additional diagnostic message (IXG065I) if an up-level LOGR CDS is attempted to be used by a downlevel system.

No explicit toleration or compatibility support is provided in this APAR's PTFs.

The PTFs for APAR OW48553 are recommended but not required to be installed:

- UW78270 - OS/390 Release 8 and Release 9
- UW78271 - OS/390 Release 10 and z/OS V1R1

The required toleration support on z/OS V1R2 is as follows:

- PTF UW82732 for APAR OW50570, which provides consistent commitment operation of logstream pending updates when z/OS V1R3 and V1R2 systems are connected to the same logstream.

Note: APAR OW50570 does not provide support for any new pending update, EHLQ or logstream update to new CF structures for V1R2 systems.

APAR OW50570 PTF is required when using the HBB7705 LOGR CDS format level.

APAR OW50570 PTF is not needed for OS/390 R2.9, OS/390 R2.10 or z/OS V1R1.

Enhanced “pending updates” require HBB7705 formatted LOGR CDS.

New or enhanced System Logger messages and other reporting

New or enhanced reporting and messages are introduced to inform you about new status and error conditions:

- IXCMIAPU DATA TYPE(LOGR) utility:
  - When **REPORT(YES)** or **LIST LOGSTREAM** is requested, the logstream attribute output section now lists new pending updates, as well as EHLQ specifications.
- **DEFINE LOGSTREAM** or **UPDATE LOGSTREAM** error message:
  - IXG009E THE MAXBUFSIZE VALUE IS NOT WITHIN THE VALID RANGE OR IS LESS THAN THE CURRENT VALUE.

More information

For more information about System Logger enhanced logstream attribute support, refer to:

- *z/OS MVS System Messages*, SA22-7640
  - Volume 10 (IXC - IZP) System Logger IXG-messages
- *z/OS MVS Assembler Services Reference,* SA22-7607
  - (IAR - XCT)
- *z/OS MVS Assembler Services Guide*, SA22-7605
  - Using IXGINVNT REQUEST=DEFINE | UPDATE, TYPE=LOGSTREAM
- *z/OS MVS Setting up a Sysplex*, SA22-7625
  - Appendix A (IXCL1DSU): LOGR CDS Versioning - New Format Levels
  - Appendix B (IXCMIAPU): Data Type(LOGR) - Define | Update Logstream options
- *z/OS Planning for Installation*, GA22-7504
- *z/OS MVS Migration*, GA22-7580

16.1.2 GRS enhancements

In this section, we review updates to GRS that are either applicable to z/OS V1R3, or were made available in the z/OS V1R3 timeframe:

- GRS improved throughput
- GRS cross-memory mode operation enhancement
- ATS Star enhancements

For explanations of other recent GRS enhancements (including GRS wildcard support, ENQ/DEQ installation, exit ISGNQXIT replacing the old RNL exit, GRS monitor tool, as well as general RNL list simplification tips and techniques), refer to *z/OS Planning: Global Resource Serialization*, SA22-7600 and the IBM Redbook *z/OS Version 1 Release 2 Implementation*, SG24-6235.

GRS improved throughput

With z/OS V1R3, a new cross-memory services lock has been added to the system for use by GRS functions. The addition of this lock enables improved multiprocessing of GRS ENQ/DEQ and GRS latch services by separating the serialization used by these functions. Previously, a single cross-memory services lock was used by both functions, which could cause unnecessary serialization delays by orthogonal (unrelated) requests. This enhancement is primary aimed at developers.

GRS cross-memory mode operation enhancement

GRS cross-memory mode operation is available on z/OS V1R2 with the new function APAR OW51103 which adds the ability to issue ENQs, DEQs and RESERVEs in cross-memory mode. This function is included in z/OS V1R3 and higher releases.

APAR OW51103 also adds several installation exits for monitoring and modifying ENQ, DEQ, and RESERVE processing (ISGNQXITBATCH, ISGNQXITQUEUED1, and ISGENDQFLQCB). These exits are intended for OEM serialization products. In addition to these exits there is another exit called ISGDGRSRES intended for ENQ exploitation. At present this exit is used by ATS Star and is not intended for general purpose use.

> **Recommendation:** If you want to affect ENQ/DEQ/RESERVE processing, the recommended approach is to use the ISGNQXIT exit. Refer to Appendix B, “Sample GRS exit ISGNQXIT” on page 401 for download instructions on how to obtain a sample exit from the ITSO Web site.

Prior to this enhancement, ENQ/DEQ/RESERVE macros could only be invoked in non-cross-memory mode. ATS Star (OW50900) requires the ability to issue the ENQ and DEQ macros while in cross-memory mode. APAR OW51103 introduces a **`LINKAGE`** parameter to the ENQ/DEQ/RESERVE macros to specify the type of linkage (**`SYSTEM`** or **`SVC`**) to be used for this service.

APAR OW51103, in conjunction with APAR OW50900 mentioned in “ATS Star enhancements”, affects the handling of automatically switchable tape devices in z/OS V1R2 and higher. This is further described in ATS Star enhancements. There are no coexistence concerns for this support. For more information refer to:

- *z/OS Auth Assm Services Reference ALE-DYN*, SA22-7609
- *z/OS Auth Assm Services Reference ENF-IXG*, SA22-7610
- *z/OS Auth Assm Services Reference LLA-SDU*, SA22-7611
- *z/OS Installation Exits*, SA22-7593

ATS Star enhancements

Starting with z/OS V1R2, with new function APARs OW50900 and OW51103, automatic tape switching is managed with GRS rather than through the IEFAUTOS CF structure. Shared tape drives are SYSTEMS in scope, and dedicated tape drives are SYSTEM in scope.

Note: This ATS Star support is included in z/OS V1R3 and higher releases.

This support removes the use of the IEFAUTOS structure. Tape drive information is maintained in the ALLOCAS address space and information is shared via XCF messaging. Tape drive serialization is managed via GRS.

ATS Star is based on the use of the auto switchable (AS) attribute on `VARY` commands and within HCD. It uses assign/unassign techniques to ensure that “foreign” use of tape drives does not cause any tape integrity errors. ATS Star uses GRS enqueues to serialize access to tape drives across a sysplex or Parallel Sysplex, and maintains tape drive information in the ALLOCAS address space. ATS Star works in basic sysplex with GRS Ring as well as in GRS Star mode (which is the recommended option). ATS Star uses XCF services to maintain device state information when allocating shared tape devices.

Both ATS Star and the IEFAUTOS function can coexist in a sysplex composed of z/OS V1R2 and levels of z/OS and OS/390 lower than z/OS V1R2, and will properly maintain the integrity of the allocation of shared tape devices across the mixed sysplex. Systems at a lower level than z/OS V1R2 will continue to use the IEFAUTOS CF structure, and systems at z/OS V1R2 and above will use the ATS Star function. Once all systems sharing the devices are at, or above, the z/OS V1R2 level, the IEFAUTOS structure can be removed from the CFRM policy.

Note: To maximize the performance of the ATS Star function as well as overall GRS performance, we strongly recommend that you use the GRS Star configuration rather than GRS Ring configuration.

Toleration support for the ATS Star function is required for users of the Multi-Image Allocation (MIA) function of Computer Associates Multi-Image (MIM). Contact Computer Associates for the associated support.

Important: The use of IEFAUTOS is defunct with the introduction of ATS Star. Once all sharing systems are at z/OS V1R2 or higher, the IEFAUTOS structure can be removed and ATS Star can be used. Devices in use by IEFAUTOS (pre-ATS Star systems) appear as “assigned to foreign host” (AFH). You may consider to suppress AFH messages during the transition period (IEF292I, IEF294I).

Both IEFAUTOS and ATS Star can share tapes within a Parallel Sysplex. ATS Star is considered the first step toward cross-sysplex tape sharing.

For ATS Star, we recommend that you apply the PTFs for APAR OW54878 to reduce certain delay situations involving reclaim time for tape devices that are AFH. Until you complete the conversion of your systems to z/OS V1R3 or higher, you may wish to dedicate your tape devices to particular systems.

Improved ATS Star diagnostics:

`Display U,AS` shows all ATS devices in your sysplex even if not online to the command issuing system, and not just the ones on current system. For an example refer to Figure 16-2.

`Display GRS,RES=(SYSZATS,*)` shows which jobs have allocated each of the tape drives accross the sysplex.

`Display GRS,[C|ANALYSE]` shows enhanced debug information (QNAME: SYSZATS).

```
IEE343I 22.44.04 UNIT STATUS 594
AUTOSWITCHABLE DEVICES CONNECTED TO SYSTEM FAGEN1
UNIT  TYPE  STATUS     SYSTEM    JOBNAME   ASID  VOLSER   VOLSTATE
05A2  348S                                                 /REMOV
05B0  349S  A          FAGEN3    HOLDTAP2  0066             /REMOV
  FAGEN3
05B1 349S                                                  /REMOV
AUTOSWITCHABLE DEVICES NOT CONNECTED TO SYSTEM FAGEN1
UNIT  TYPE  STATUS     SYSTEM    JOBNAME   ASID  VOLSER   VOLSTATE
                                                           /REMOV
  FAGEN3(05A0,348S)
```

Figure 16-2 System reply to D U,AS command (a.k.a. "show all autoswitchable devices")

For further details, refer to information APAR II13214 (or II13195, if you use MIA).

16.1.3 Sysplex-wide confighfs command scope

The restriction of issuing the **`confighfs`** command only from the system on which the HFS file system is mounted has been removed in z/OS V1R3. The **`confighfs`** command can now be issued from any system within a sysplex. This assumes that the system on which the file system is mounted is also running z/OS V1R3 or later.

For more information refer to 8.7.1, "Shared HFS support for confighfs command" on page 172; you may also refer to *z/OS DFSMS Migration*, GC26-7398.

16.1.4 Sysplex mount table limit monitoring

Prior to z/OS V1R3, it was difficult to monitor the shared HFS mount tables. There was no IBM-architected way to alert you if the table was reaching critical levels unless you coded your own alarm routines. In z/OS V1R3, a console message (BPXI043E) is issued when the mount table for the shared HFS usage reaches a critical level. This alerts you so you can take pro-active actions such as defining a larger alternative CDS and then switching to it. For more information refer to "Mount table limit monitoring" on page 168.

16.1.5 Sysplex mount/unmount performance improvements

z/OS V1R3 provides performance enhancements for mount and unmount USS commands. This will especially benefit sysplexes with high numbers of HFS mounts. For more information refer to "New UNMOUNT option" on page 166.

16.1.6 DFSMShsm™ common recall queue (CRQ)

Prior to z/OS V1R3, DFSMShsm recall requests were only processed by the system (or "host", as it is normally referred to in DFSMShsm jargon) on which they are initiated. Notice that the term "host" refers to an DFSMShsm address space rather than a z/OS system. Starting with DFSMS Release 10, many DFSMShsm hosts may run as many address spaces on one z/OS or OS/390 system.

The DFSMShsm common recall queue (CRQ) introduced in z/OS V1R3 allows all hosts in an HSMplex to place their recall requests onto a single queue. It also enables tape mount optimization and priority optimization. This is because it allows a host that has a tape mounted for recall to process all recall requests requiring that tape, regardless of which host initiated the recall request. In addition, it provides added flexibility for Parallel Sysplex configurations where not all hosts are connected to all devices.

When combined with DFSMShsm's multiple address space support (introduced in OS/390 R10 DFSMS Release 10), the CRQ enhancement enables you to increase the number of concurrent recall tasks above the previous limit of 15 tasks per z/OS image.

The CRQ is a single recall queue that is shared by multiple DFSMShsm hosts. The CRQ enables recall workload balancing across these hosts. This queue is implemented through the use of a Coupling Facility list structure. The CRQ CF structure exploits System-Managed CF Structure Duplexing. Although it is not generally recommended, you may have several HSMplexes in your sysplex, and each HSMplex could have its own CRQ.

> **Restriction:** A DFSMShsm CRQ may only be shared between hosts that are sharing the same HSM CDSs in the same HSMplex. Several HSMplexes and CRQs may exist within your sysplex.

To implement and enable this function, you need to perform the following tasks:

- Update your CFRM policy
- Update your DFSMShsm PARMLIB member
- Update your operational procedures

In the following sections we review each of these actions in turn, but first we need to briefly discuss the software and hardware dependencies for CRQ.

CRQ hardware and software dependencies

DFSMShsm must be in a Parallel Sysplex configuration, since CRQ requires a CF list structure. The minimum CFLEVEL is 8, and the preferred CFLEVEL is 11 or higher. CFLEVEL11 offers an enhancement that DFSMShsm takes advantage of, which is System-Managed CF Structure Duplexing. CFLEVEL 12 and higher provides additional performance benefits for System-Managed CF Structure Duplexing.

All hosts using the CRQ must be at z/OS V1R3 DFSMShsm or higher. Other hosts in the same HSMplex that are not using the CRQ can be at any supported level. Hosts can be converted to use CRQ individually, or all at once. No coexistence APAR is needed. Down-level hosts will be treated as if they were held by the **HOLD COMMONQUEUE** command, which is discussed in "CRQ CFRM policy considerations".

CRQ CFRM policy considerations

Update your active CFRM policy with information about the CRQ CF structure.

CRQ CF structure sizing

You should size the CRQ structure large enough to hold the maximum number of concurrent recall tasks that may occur in your HSMplex. Because of the dynamic nature of DFSMShsm recall activity, there is no exact way to determine what the maximum number of concurrent recalls are.

For this reason, at the time of writing IBM recommends that a structure size (INITSIZE) of about 5 MB be used. A structure of this size is large enough to manage up to approximately 4000 concurrent recall requests. For a start, code a corresponding SIZE value of 10 MB. If you expect to use more than this in your environment, then use the CFSizer which is available at:

http://www.ibm.com/servers/eserver/zseries/cfsizer/crq.html

As input to the CFSizer, you need to enter the *maximum number of concurrent recalls* that you expect and the *average percentage of recalls that require ML2 tapes*. Note that the structure size returned by this tool may vary somewhat from the recommended sizes listed in the *DFSMShsm Implementation and Customization Guide,* SC35-0418. This is due to the fact that CFSizer estimates the size of the structure needed, while the publication lists actual sizes that were used in a particular sample environment.

CRQ CF structure name

The structure name is SYSARC_'basename'_RCL, where basename is the base name specified in SETSYS COMMONQUEUE(RECALL(CONNECT(basename))). Note that basename must be exactly five characters.

CRQ DFSMShsm PARMLIB considerations

Refer to *DFSMShsm Implementation and Customization Guide*, SC35-0418 for information on how to use typical **SETSYS** commands for your system. A set of **SETSYS** commands can become part of the **ARCCMDxx** member pointed to by the DFSMShsm startup procedure.

As an example, you may want to update the DFSMShsm PARMLIB member to include the **SETSYS COMMONQUEUE(RECALL(CONNECT(basename))** command.

CRQ operational and other considerations

The CRQ list structure implements “locks”. XES, on behalf of CRQ, maintains an additional XCF group in relation to this structure. Make sure that your XCF configuration can support this addition. The XCF group name is IXCLOxxx, where xxx is an unpredictable 3-digit hexadecimal number.

> **Attention:** CF failure-independence is strongly recommended for the CRQ structure, especially if System-Managed CF Structure Duplexing is not implemented.

The DFSMShsm CRQ structure provides support for:

- The alter function, including directory-to-element ratio alters
- System-managed rebuilds (and no support for user-managed rebuilds) System-Managed CF Structure Rebuild does not support the REBUILDPERCENT option.
- System-Managed CF Structure Rebuild

The CRQ structure is persistent with non-persistent connections. The structure remains allocated even if all connections have been deleted.

DFSMShsm monitors utilization of the CRQ list structure. If the structure becomes 95% full, DFSMShsm no longer places recall requests onto the CRQ, but routes all new requests to the local queues. Routing of recall requests to the CRQ resumes once the structure drops below 85% full. When the structure reaches maximum capacity, you can increase the size by altering the structure to a larger size or by rebuilding it (and as always, remember to bump up the size in your CFRM policy if you want this size change to become permanent).

A rebuild must be done if the maximum size has already been reached. The maximum size limit specified in the CFRM policy must be increased before the structure is rebuilt. You can use the CF services structure full monitoring feature to monitor the structure utilization of the CRQ.

CRQ processing is very flexible and allows you many configuration options including the following:

- Allow all hosts to be able to both place and process recall requests from the CRQ. This is normally the recommended option because it is the most flexible.
- Specify one or more hosts to be able to both place and process recall requests from the CRQ, and the remainder of the hosts in your HSMplex to be only able to place requests.
- Have some of your systems only process local requests.
- DFSMShsm multiple address space support can provide benefit in a monoplex. This allows other hosts, in addition to just the main host, to use the CRQ.

> **Note:** You use the DFSMShsm `HOLD` command to configure a host to not select recall requests. On hosts that you want to only accept recall requests, issue the `HOLD COMMONQUEUE(RECALL(SELECTION))` command. These hosts will then place recall requests on the CRQ, but not process them.
>
> You may increase the number of recall tasks from 15 to x multiplied by y multiplied by 15. x is the number of DFSMShsm address spaces on the z/OS image, and y is the number of z/OS images in your HSMplex.
>
> If certain hosts are not connected to tape drives, they can be configured to accept all recall requests but only process those requests that do not require ML2 tape. You use the `HOLD RECALL(TAPE)` command for this purpose.
>
> Many DFSMShsm commands are designed to allow the addition of future common queues. At the time of writing, only the common recall queue is available.

You can refer to *z/OS V1R3 Parallel Sysplex Test Report*, SA22-7663, for more information.

16.1.7 Caching of larger than 4 KB CIs in VSAM RLS cache CF structure

VSAM RLS CF caching enhancements in DFSMS z/OS V1R3 allow you to specify the amount of data that is cached in the CF cache structure defined to DFSMS. The effect is that control intervals larger than 4 KB may be cached for enhanced data sharing read performance.

Prior to DFSMS z/OS V1R3, VSAM RLS CF did not cache data that was greater than 4 KB. You may have experienced performance degradation when your CI size was 32 KB, because the corresponding CF operations were asynchronous. DFSMS also experienced extra associated overhead due to page fixing and castin processing.

To activate the RLS CF caching support:

- Ensure all systems in your sysplex are using z/OS V1R3 or higher.
- Specify the keyword **RLS_MaxCfFeatureLevel(A)** in the IGDSMSxx PARMLIB member:
  - Specify **A** to cache VSAM data with control intervals (CIs) larger than 4 KB in the CF cache structure.

    Note: If you do not specify a value, or if you specify **Z** (which is the default value), the system caches only VSAM data sets with control intervals of 4 KB or less in the CF cache.

  - To determine the RLS_MaxCfFeatureLevel for a single system in the sysplex, use the following command: **D SMS,SMSVSAM**
  - To determine the RLS_MaxCfFeatureLevel for the entire sysplex, use the following command: **D SMS,SMSVSAM,ALL**
  - To display a specific CF cache structure, specify the cache structure name using the following command: **D SMS,CFCACHE(cacheName)**
  - To display all the CF cache structures that are defined in the SMS active configuration, use the following command: **D SMS,CFCACHE(*)**

ISMF supports the **RLSCFCACHE** keyword with a new field in the DATACLAS panel related to this enhancement. A new keyword is available for the **SETSMS=XX** command, the **SETSMS** command, and the **D SMS,Options** command. To set the CF controls, specify one of the following values in the **RLSCFCACHE** field in the SMS DATACLAS panel:

- **NONE** indicates that VSAM/RLS caches only the VSAM index entries and does *not* place any VSAM data sets in the CF cache structures.
- **UPDATESONLY** places only the changed (write requests) VSAM data in the CF cache structures.
- **ALL** places all of the VSAM data and index entries in the CF cache structures (default).

Important: For best overall data sharing performance, we recommend using the **ALL** value for the **RLSCACHE**. This ensures fast reads for data contained in the CF cache structure and puts less of a burden on your I/Os.

All your systems must be at z/OS V1R3 or higher levels for this support to be enabled. Once all systems have the function installed, the function can be activated. You must restart the SMSVSAM server in order for the function to be activated. If DFSMS z/OS detects that a system does not have the support installed, messages are generated. RLS_MaxCfFeatureLevel is a sysplex-wide parameter. The first system IPLed in your sysplex determines the value of RLS_MaxCfFeatureLevel.

Install toleration PTF OW49450 on all downlevel systems (OS/390 V2R10 and higher) in your sysplex for this support. At the time of writing, this PTF was in error. You should review the current status in PTF OW50795. RLSCACHE values and SMS DATACLASS names are captured in SMF Type 42 (sub-type 15 and 16) records.

For more information about how to size this structure, refer to:

www.ibm.com/servers/eserver/zseries/cfsizer/vsamrls.html

For more information about this VSAM RLS enhancement, refer to *z/OS DFSMS Migration*, GC26-7398.

16.1.8 VSAM RLS lock structure duplexing enhancement

The VSAM RLS CF lock structure (IGWLOCK00) contains information used to determine cross-system contention on a particular shared VSAM data resource in your Parallel Sysplex. It also contains information about locks that are currently used to control changes to shared resources.

VSAM RLS lock table CF structure duplexing provides the following support:

- A validity check function during the *user-managed rebuild* and lock structure **ALTER** process. This validity-check ensures that the new lock structure has enough free space for locking to proceed.

> **Important:** We recommend that you use the rebuild process to change the size of the VSAM RLS lock structure instead of the ALTER process (since the ALTER does not change the number of lock entries).

- *System-managed duplexing rebuild* function for VSAM RLS lock structures.

Changes and enhancements have been made to the following operator commands dealing with the VSAM RLS CF lock structure in z/OS V1R3:

- **SETXCF START,REBUILD,DUPLEX**

  Issuing this command starts the system-managed duplexing rebuild for the VSAM RLS CF lock structure.

> **Important:** You must specify the **DUPLEX(ALLOWED)**, and should also specify **DUPLEX(ENABLED)** keyword in the CFRM policy before you can use **SETXCF** to start the duplexing function.

- **SETXCF STOP,REBUILD,DUPLEX**

  This command stops the system-managed duplexing rebuild for the VSAM RLS lock structure.

> **Important:** Another way to turn off the duplexing function is to specify the **DUPLEX(DISABLED)** keyword in the CFRM policy.

- **DISPLAY SMS,CFLS**

  This command displays information about the CF lock structure for System-Managed Structure Duplex Rebuild.

- **DISPLAY SMS,SMSVSAM[,ALL]**
- This command displays the status of SMS VSAM servers and the lock structure mode.
- **V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE**

  This command resets the space if validity checking indicates that there is not enough space in a specific lock structure for future locking activity to proceed. This command is rejected if it occurs while duplexing mode is being established.

- **DISPLAY SMS,SEP**

  This command displays the name of the data separation profile.

To use system-managed duplexing, you need the following:

- CFLEVEL 11 or higher depending on your particular hardware; for up-to-date information, refer to:

  http://www.ibm.com/servers/eserver/zseries/pso/cftable.html
- CF-to-CF links
- A duplexed pair of primary and secondary VSAMRLS lock structure instances
- Possible additional CF storage, processor and link capacity

All systems must be z/OS V1R3 or later. This is not a general requirement for system-managed structure duplexing, but it is required for system-managed structure duplexing support for VSAM RLS lock structures.

For more information about how to size the VSAM RLS CF lock structure, refer to:

http://www.ibm.com/servers/server/zseries/cfsizer/vsamrls.html

For more information about this VSAM RLS enhancement refer to *z/OS DFSMS Migration*, GC26-7398.

For more discussion on DFSMStvs, refer to Chapter 3, “Base control program (BCP)” on page 31.

For more information about System-Managed CF Structure Duplexing, refer to “System-Managed CF Structure Duplexing” on page 289.

16.1.9 DFSMS enforced data set separation for high availability

Data set separation allows your storage administrator to designate groups of data sets to be kept separate at the physical control unit (PCU) level. Failure isolation means separate volumes, control units, storage subsystems, and paths to the controllers.

> **Important:** By using this function, you can ensure that if one control unit fails, the sysplex can access the data sets via another control unit.
>
> We highly recommend you use this function to failure-isolate your sysplex CDSs.

An earlier, IBM-supplied prototype of this tool has now been enhanced further and incorporated into z/OS DFSMS V1R3. You need to determine which critical IBM and OEM data sets should be kept separate, and create a data set separation profile for SMS. As an example, you may want to keep the following data set types behind separate control units:

- Software-duplexed data sets such as DB2 logs, CFRM and other Couple Data Sets are candidates for this support. The CDS instances are often referred to as *primary* CDSs and *secondary* CDSs. There are many other data sets (both user and subsystem data sets) that have similar separation requirements.

To control this, a **SEPARATIONGROUP** keyword is introduced when you code your SMS routines. To try this out, go into ISMF, select option **8**, point at the SCDS, then select option **3** and you will a screen similar to the one shown in Figure 16-3 on page 268.

```
HELP-----------------------  DS SEPARATION PROFILE  ----------------------HELP  .
.  COMMAND ===>                                                              .
.                                                                            .
.     Use DS SEPARATION PROFILE field to specify the dataset name that will  .
.     provide SMS with a list of dataset names needing separation.           .
.     Storage Administrators will be able to designate groups of data sets   .
.     where all SMS managed data sets within the group must be kept separate .
.     from other sets within the same group.                                 .
.                                                                            .
.     Possible values:                                                       .
.                                                                            .
.         Any valid PS, PDS or PDSE data set name.                           .
.                                                                            .
.     For example:                                                           .
.                                                                            .
.         DS Separation Profile                  (Data Set Name)             .
.          ==> 'SEPARATN.HENRIK.PROFILE'
.
.                    (or)                                                    .
.          ==> 'SEPARATN.HENRIK.PROFILE.PDS(SMSDS)'
.
.                                                                            .
.     Use ENTER to continue, END to exit Help.                               .
.                                                                            .
.   F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=          F6=    .
```

Figure 16-3 ISMF Data set separation profile

For more information about this high availability enhancement, refer to *z/OS DFSMS Migration*, GC26-7398.

16.1.10 OAM multiple object backup

z/OS V1R3 DFSMSdfp provides enhancements to the Parallel Sysplex support for Object Access Method (OAM). Before we discuss these enhancements, let us take a look at OAM support for Parallel Sysplex.

OAM support for Parallel Sysplex

Each DFSMS OAM instance running in a Parallel Sysplex communicates with all other instances of OAM on same or other systems in the sysplex, using XCF services, provided they belong to the same XCF group. Each instance of OAM in a single Parallel Sysplex, sharing the same database for OAM, must belong to the same XCF group, and the DB2 subsystem must belong to the same DB2 data sharing group. This collection of address spaces in a sysplex is known as an OAMplex. Any OAM object may be retrieved from any z/OS system in a Parallel Sysplex regardless of which OAM stored the object or on which media (DASD, 3995 optical, or tape) the object resided, as long as each instance of OAM is part of the same OAMplex.

OAM restrictions removed in z/OS V1R3

OAM no longer has a 100-object storage group restriction. It is now possible to define more than 100 object storage groups in your configuration. If fewer than 100 object storage groups are needed, views are no longer required to define all 100, as was previously required.

OAM multiple object support is introduced in z/OS V1R3 DFSMSdfp. With OAM multiple object backup support, you can physically separate backup copies of objects based on the object storage group to which the object belongs. You can direct your backup copies of objects to different media types, based on the definitions for the target object backup storage group that will contain the backup copy.

The multiple object backup storage group capability allows you to make a second backup copy of objects. You can direct OAM to create up to two backup copies of objects using current fields in the SMS management class construct and by specifying a first and a second object backup storage group for your object storage groups in the CBROAMxx member of PARMLIB. Additionally, you can direct OAM to write the first and second backup copies on the same removable media type or on different removable media types.

Finally, with this support, OAM improves the overall reliability and usability of the volume recovery utility. OAM will provide a full list of volumes required to accomplish a recovery beyond the previous limit of 70.144 objects that could be recovered during a single invocation of the recovery utility. Additionally, during a volume recovery, OAM provides improved informational messages. It also provides the ability to obtain statistics on the volume recovery.

The z/OS V1R3 release of OAM provides coexistence for lower-level systems in an OAMplex in the following areas: object directory tables, post-migration fallback to a previous release, and XCF messaging. In regard to modified object directory tables, note that the DBRMs that interface with the object directory tables will now select specific columns instead of the full row.

This coexistence is necessary when falling back to a previous release of OAM after migrating to the current release of OAM, or when z/OS V1R3 and one or more systems at a previous level are in the same OAMplex. In regard to XCF messaging in z/OS V1R3, coexistence has been provided for OAM messages sent through XCF across lower-level and current-level system OAMs in an OAMplex.

For more information, including implementation details about this OAM enhancement, refer to *z/OS DFSMS Migration*, GC26-7398.

RLS serialization mechanisms

RLS uses the following serialization mechanisms:

- Locking
- Buffer coherency
- DASD write serialization
- CI/CA split serialization

Prior to this support, batch jobs could read but not update while CICS had the data sets open in RLS mode. RLS did not permit a non-CICS application to open for output a recoverable file in RLS access mode. This was because it was a CICS responsibility to do the logging and two-phase commit coordination.

By adding logging and two-phase commit, as well as backout protocols at the file-system level, TVS allows batch jobs to share recoverable VSAM data sets for both read and update access while CICS is still using them.

TVS sharing possibilities

TVS provides you with the ability to share VSAM data sets for read and write processing between online CICS applications and batch jobs, or between multiple batch jobs in a near-continuous availability environment. Your batch applications may now also interact with multiple resource managers concurrently, including VSAM, IMS, and CICS.

TVS use of ARM, System Logger, and RRS services

TVS may use the Automatic Restart Manager (ARM). TVS services may be restarted on other systems by ARM if system outages occur.

TVS logging services depend on the System Logger (DASD-only or CF logstreams). Backout logs contain “before” images and are not shared between systems in your sysplex. Forward recovery logs may be merged and be processed by CICS/VR or equivalent products.

TVS uses Recoverable Resource Management Services (RRMS) and Recoverable Resource Services (RRS) as sync point managers for two-phase commit processing.

TVS application considerations

The use of TVS has performance implications because all updates to recoverable resources are serialized and logged. This, in turn, translates into cost elements for cross-address space access to servers, record level locking, CF cache accesses and logging. TVS performance is very similar to CICS RLS performance.

You may reduce the performance implications in your environment by using:

- Sequential VSAM processing (causing fewer CI splits and less unused space within the CIs).
- You may also use skip sequential VSAM processing when reading multiple records whose keys are close (not necessarily adjacent).
- Use `GET CR` or `GET CRE` rather than `GET UPD` to avoid taking exclusive locks.
- Review your application read integrity options, specified in JCL, or at the ACB level, or at the RPL level.
  - `NRI` (default): “No Read Integrity” provided, thus no shared locks are used.
  - `CR`: “Consistent Read”, causing queuing for read requests if the record is being updated by another task. After completion of the update request, the shared lock is released and the read performed. By waiting for updating tasks, you ensure you do not read uncommitted data.
  - `CRE`: “Consistent Read Exclusive”. This is similar to Consistent Read, except that the reader holds a shared lock until a sync point is processed and the shared lock is released.

Important: While you may be able to implement TVS in your application environment without issuing explicit sync points, it is not generally recommended. Your applications should be coded in such a way that they “understand” sync point logic. Such understanding involves locking and shortening of units of work, as well as restart considerations.

You may want to consider migrating some of your applications from VSAM to DB2. Such considerations also include cost implications of rewriting your existing application logic, as described.

The VSAM functions NSR chained sequential I/O and LSR deferred write are not supported by TVS.

RLS and TVS information

For more information about RLS, refer to the following publications:

- *CICS and VSAM Record-level Sharing: Planning Guide*, SG24-4765
- *CICS and VSAM Record-level Sharing: Implementation Guide*, SG24-4766

- *CICS and VSAM Record-level Sharing: Recovery Considerations*, SG24-4768
- *DFSMS z/OS DFSMSdfp Storage Administration Reference*, SC26-4920

IBM Systems Journal Volume 36, Number 2, 1997 titled *S/390 Parallel Sysplex Cluster* contains an article about VSAM record-level data sharing and is available on the Internet:

http://www.research.ibm.com/journal/sj/362/strickland.html

16.2 Parallel Sysplex enhancements for z/OS V1R4

In the following section, we review the sysplex enhancements introduced in z/OS V1R4.

16.2.1 XES DB2 Data sharing performance improvement

The XES component of z/OS V1R4 provides new IXLCACHE functions to enhance performance when accessing a cache structure allocated in a CF of CFLEVEL 12 or higher. Three new IXLCACHE macro invocations are provided by XES. Their intent is to improve performance for data sharing systems by batching together high-frequency cache structure operations, and thereby reducing the number of commands sent to the CF.

This enhancement allows for batched group buffer pool (GBP) writes and CF castout data requests, as well as CF cross-invalidate requests in single CF operations. This may reduce data sharing cost in update- or insert-intensive DB2 data sharing environments. IBM intends to exploit these functions in a future release of DB2.

16.2.2 XES CFRM performance enhancements

CFRM CDS I/O performance is key to Parallel Sysplex performance, especially during sysplex recovery including CF structure rebuild activities.

CFRM system termination cleanup enhancements

APAR OW48624 (and related PEs OW50205 and OW50867), which is rolled in to z/OS V1R2 and may be retrofitted back to OS/390 R8, provides CFRM I/O processing enhancements relating to system termination cleanup processing for all types of removal of systems from the sysplex. During such processing activity, surviving OS/390 and z/OS systems may see improvements in performance with this support.

Improvements to performance in the following areas is available for sysplex recovery routines when processing recovery for connectors to CF structures following a system failure:

- Determining the connectors that were active on the failed system and initiating cleanup events for those connectors will be executed by only one system in the sysplex, instead of all systems.
- The algorithm for processing the confirmations from all the other active connectors has been updated to improve its efficiency.
- Critical structures will be given priority by the functions that are processing the system failure events.
- CFRM I/O processing has been reduced for user sync point (IXLUSYNC) event processing.

The goal of APAR OW48624 is to ensure improved Parallel Sysplex performance, continuous availability, and scalability by reducing CFRM I/O contention during connection cleanup phases of sysplex recovery for system failures.

There are no coexistence considerations for this enhancement. Systems with and without this function may be mixed in your sysplex. The more systems that have this function, the better the overall performance improvement you will observe.

There are no externals and no implementation considerations for the CFRM system termination cleanup enhancement.

You may obtain further CFRM I/O contention relief in z/OS V1R4; for more information refer to “CFRM CF structure rebuild performance enhancement”.

CFRM CF structure rebuild performance enhancement

In z/OS V1R4, XES provides CFRM performance enhancements to minimize the amount of I/O to the CFRM CDS for IXCQUERY commands. This is accomplished by first analyzing the IXCQUERY request to reduce the data needed. Following this analysis, a minimum amount of policy data is then retrieved to satisfy the request.

Access to the CFRM CDS is serialized across systems in the sysplex, so other XES services such as IXLCONN, IXLDISC, and IXLREBLD will also benefit from the CF structure rebuild performance enhancement.

CFRM I/O contention relief enables optimized structure rebuild confirmation processing by handling all queued event confirmations in the same function call.

CF structure rebuild tasks perform better as a consequence of the enhanced IXCQUERY request processing. CF structure rebuild processes include:

- User-managed rebuild
- User-managed duplexing rebuild
- System-managed rebuild
- System-managed duplexing rebuild

Reduced CFRM contention enables faster CF structure rebuilds. This will provide benefit especially in rebuild scenarios when many CF structures are rebuilt, such as in the event of CF connectivity failures or CF failures.

There is no implementation activities relative to this function. You can have a sysplex environment with a mix of systems at different z/OS releases. z/OS V1R4 and higher release systems will benefit from this performance enhancement.

Note: Reducing CFRM I/Os is one way to improve sysplex performance for system termination cleanup and rebuild processes as outlined in “CFRM system termination cleanup enhancements” on page 271 and “CFRM CF structure rebuild performance enhancement”.

Another way to increase general sysplex performance is to speed up I/O processing for CFRM CDS I/Os. Significant I/O performance improvements have been observed by using FICON-attached DASD to hold CFRM CDSs.

16.2.3 RRS multisystem cascaded transaction enhancement

RRS multisystem cascaded transaction was provided in z/OS V1R2 to allow RRS-enabled resource managers to participate in transactions that span systems in a sysplex. RRS multisystem cascaded transactions enhancements affect both RRS system management panels and programmable interfaces. This enhancement allow RRS to coordinate a set of units of recovery with separate work contexts across multiple systems in a sysplex under a single commit scope, by providing an architected failure notification method to RRS-enabled resource managers.

At the time of writing, this support is implemented by IMS V8 for its usage of RRS multisystem cascaded transactions to assist in resource manager failure recovery. Figure 16-4 shows a scenario to illustrate the value of this enhancement. Consider the following points:

- APPC/OTMA transactions are load-balanced using MQseries shared message queues in a multisystem sysplex that contains three IMS systems: IMSA, IMSB, and IMSC.
- The front-end IMS(A) creates a coordinator RRS unit of recovery and places the IMS transaction on a MQseries shared queue.
- A backend IMS(C) pulls the IMS transaction from the MQseries queue and creates a subordinate RRS unit of recovery.

Figure 16-4 IMS V8 APPC/OTMA and MQseries Shared Queue use of RRS Multisystem Cascaded Transaction support

IMSC signals to IMSA when the IMS transaction is ready to processed. Once the signal is received by IMSA it will initiate an RRS syncpoint, which in turn will commit or back out the sysplex cascaded unit of recovery. So far so good. (Prior to this enhancement, if the backend IMSB, due to a failure, cannot signal the front-end IMSA, no syncpoint is taken and the transaction does not complete; that is, it is hung.)

Starting with z/OS V1R4, RRS can determine if a failure affects a subordinate unit of recovery, and notify the coordinator unit of recovery about the failure. Once notified, the coordinator unit of recovery can initiate RRS backout for the IMS transaction. In this scenario, a transaction can still complete if IMSB, due to failure, cannot signal the front-end IMSA.

The automated RRS-initiated backout process is possible because RRS is capable of driving a **SUBORDINATE_FAILED** exit in IMSA. The optional **SUBORDINATE_FAILED** exit routine receives control for a sysplex cascade top-level UR when either RRS or any resource manager on a subordinate system fails, or else the subordinate system itself terminates, or the context associated with the subordinate UR abnormally terminates.

This support is provided in z/OS V1R4, and APAR OW50627 provides this support on z/OS V1R2 and z/OS V1R3. The function is only available if all systems in the RRS logging group have the function available.

For more information, refer to *z/OS Programming: Resource Recovery*, SA22-7616.

16.2.4 System Logger offload monitor function

System Logger offload monitor function is delivered with APAR OW51854. APAR OW51854 may be applied to OS/390 R10 and higher releases. It is integrated into z/OS V1R4, and provides a new System Logger monitoring function for offload processing. There are no coexistence issues.

System Logger potential sysplex contention

Only one IXGLOGR address space is allowed to offload a given System Logger logstream to DASD at any point in time. If one System Logger starts an offload for a given logstream, then no other System Logger address space will attempt to offload the same logstream. If that particular offload is delayed for any reason, such as SYSIEFSD Q4 resource contention, then the offload process for the logstream under discussion is *delayed sysplex-wide*. This can lead to other System Logger allocation delays as well. (SYSIEFSD Q4 is used by dynamic allocation (SVC99) as part of its data set serialization. This enqueue is not done by System Logger; it is done in the MVS allocation path.)

With the advent of APAR OW51854, System Logger provides ways of identifying offload inhibitors via messages or monitoring. System Logger monitors offload progress. A monitor function has been added to Logger. This function will notify the installation when offloads are taking too long, and allow the installation to take action by replying to WTOR IXG312E

System Logger periods of inactivity may, for example, be caused by offload data set allocation or DFSMShsm recall activity. Logger is now able to move offload work from a system where it is not progressing, to another system connected to the logstream, to attempt to complete the offload. If there is no other system to move the work to, the offload may be attempted again on the same system, possibly allowing other work to proceed.

In the following sections we discuss how these enhancements are implemented.

System Logger logstream offload activity warning messages

With APAR OW51854, System Logger monitors logstream offload activity on an ongoing basis. If an offload process appears to be hung or is taking too long, System Logger notifies you by issuing new messages, such as IXG310I. In “New or enhanced System Logger messages and other reporting” on page 258, we discuss other accompanying messages.

System Logger data set requests are, at the time of writing, considered to take too long if a time interval of more than 30 seconds is spent for allocation and 60 seconds is spent for DFSMShsm recall activity. If one time interval has elapsed, System Logger will issue warning messages.

A sample IXG310I offload warning message is shown in Figure 16-5.

```
IXG311I SYSTEM LOGGER CURRENT OFFLOAD HAS NOT PROGRESSED
DURING THE PAST 65 SECONDS FOR LOGSTREAM OW51854.STREAM1, STRUCTURE: LIST01
DELETING DSN=IXGLOGR.OW51854.STREAM1.<SEQ#>
```

Figure 16-5 IXG311I warning message - System Logger offload has not progressed

System Logger logstream offload activity WTOR

If a second inactive time interval of offload activity is detected (for a total of 60 seconds for allocation and 120 seconds for DFSMShsm recall activity), an additional warning message IXG311I is issued indicating the offload has not progressed. A WTOR (as shown in Figure 16-6) is issued, prompting operations to reply with one of the following:

Monitor Continue monitoring the offload (and reset timer for another interval).

Ignore Stop monitoring this offload.

Fail Terminate this offload (and attempt to direct the offload to another system if possible. Note that the offloads are owned on a competitive basis by any system that has an active connection to the logstream. The system that was performing an offload and requested to “Fail” (stop) as part of the monitoring response, will also compete to perform subsequent offloads. However, this system will be given a *disadvantage* in the ownership competition, thereby giving the other systems an advantage to gain ownership).

Autofail Terminate this and all future offloads for this logstream on this system while still connected.

Exit Stop all monitoring on this system for the life of the IXGLOGR address space.

An example of a sample IXG312E offload delay message is shown in Figure 16-6.

```
09 IXG312E OFFLOAD DELAYED FOR OW51854.STREAM1, REPLY "IGNORE",
"MONITOR", "FAIL", "AUTOFAIL" OR "EXIT".

IXG311I SYSTEM LOGGER CURRENT OFFLOAD HAS NOT PROGRESSED
DURING THE PAST 61 SECONDS FOR LOGSTREAM OW51854.STREAM2, STRUCTURE: LIST01
ALLOCATING DSN=IXGLOGR.OW51854.STREAM2.<SEQ#>
```

Figure 16-6 IXG311I/IXG312E messages - System Logger offload has not progressed, intervention required

If this message does not receive a reply, and the problem goes away, it will be DOMed.

System Logger messages pertaining to logstream offload monitoring

In addition to the messages shown in Figure 16-5 and Figure 16-6, System Logger provides the following new or enhanced messages:

- New IXG messages:
  - IXG066i - Monitor not active - failed or told to EXIT
  - IXG310i - First interval message
  - IXG311i - Second interval message to precede IXG312e

- IXG312e - WTOR requesting action reply
- IXG313i - Logger AUTOFAILed an Offload
- IXG303i - Message when directed or “mis-directed” offload starts
- IXG304i - Message when directed or “mis-directed” offload completes

- Changed IXG messages:
  - Ixg115a - Updated explanation and operator response
  - Ixg116i - Issued from more than one place now
  - Ixg301i - Additional reason code info

For a discussion on recommended procedures for responding to the IXG messages, refer to *Setting Up a Sysplex*, SA22-7625. For detailed descriptions of IXG messages, refer to *z/OS System Messages*, SA22-7640.

System Logger general recommendations:

- Use the following commands to check for System Logger problems:
  - **D LOGGER[,C|,L]** command
  - **D XCF,STRUCTURE** command
  - **D GRS,C** command
- React to Allocation Messages first such as IEF861I,IEF863I, IEF458D, and so on.
  - Then respond to IXG312
    - **MONITOR** - May show a different DSN if interim progress is being made (DSN Delete, HSM Recall, and so on.)
    - **FAIL** to Fail the offload
    - **AUTOFAIL** to Fail offload and all subsequent offloads

Message will be DOMed if problem goes away

  - Respond to IXG115 last
- Several messages can appear at same time for different logstreams:
  - Usually, only one logstream is having a problem; the rest are waiting for this logstream. The first logstream reported is usually the culprit.
  - Other hints: If a data set name ends with, for example, .A0000001, that usually indicates this logstream is the problem. If a data set name ends with .<SEQ#>, that usually indicates this logstream is waiting for another.

You may experience ABEND1C5-'00070020'x issued and a dump if you reply **FAIL** more than once. This means that for a Structure Full Offload, no progress was made. This is “normal”.

Always stay current with System Logger service.

System Logger-related APARs of interest

The following is a list of recent (at the time of writing) System Logger-related HIPER APARs:

- OW49909- Rebuild stop response not always provided.
- OW50564- Logstream data set directory cleanup when LOGR CDS corrupted.
- OW51713- ABEND1C5-00010001 can be seen when Logger services and offload are running at the same time, causing offloads to fail.
- OW52101- High-Used RBA set to High-Alloc RBA for current offload data set until switched off, then set to actual High-Used RBA.

- OW52110- ABEND1C5 RSN00070021 during Offload when deleted data not taken into account.
- OW52168- Logger orphaning offload data set eligible for deletion.
- OW52589- Pcntl/Ocntl record overlaid in LOGR CDS corrupts logstream, unable to offload successfully.
- OW52796- Dynamic entry/element ratio consistency.
- OW53126- ABEND1C5-0009000B can result if an I/O error occurs on a pack used for staging data sets. This can lead to all DASD-only logstreams being disconnected.
- OW53182- ABEND0C4 when logger ran out of local buffers for duplexing log data.
- OW53235- Logstream writer hung after logger rebuild thread error processing held logstream latch indefinitely.
- OW53349 - Automatically restart IXGLOGR after failure.
- OW53449 - Logstream left in “Disconnect Pending” state.
- OW54389 - ABEND0C4 in IXGC4RBE during structure rebuild
- OW54511 - Logstream offloads not occurring following CF System-Managed Rebuild.
- OW54575 - Logger hangs Structure failure event notification while in CF System-Managed Duplexing Rebuild quiesce phase.
- OW54631 - IXGWRITE rc8, rsn862, but no ENF48 signal is issued to indicate logstream available again.
- OW54923 - IXGLOGR terminates after ABENDA78, rsn18 in IXGS7STG.
- OW55019 - IXGBRWSE rc8, rsnn804, or incorrect data returned.
- OW55055 - ABEND0C4 in IXGWORKT or ABEND0C4 in IXGR2EOT during EOM cleanup due to corrupted LCNTL chain.
- OW55371 - IXGLOGR terminates after ABEND0E0 RC29 in IXGT3LBM.
- OW55916 - Logger might not completely clean up its environment during recovery.

Other System Logger APARs:

- OW51437 - Raise limit of concurrent DASD-only logstreams connected on one system to 1024.
- OW51855 - Dynamic entry/element ratio consistency.
- OW52075 - System Logger may fail a browse attempt when trying to recall data that is no longer migrated; can lead to gaps in log data.
- OW52195 - Browse request ABEND0C9 can occur when DSEXTENT (Dirct) contraction occurs simultaneously.
- OW53822 - Staging data set I/O error leaves logstream in “Disconnect Pending” state.

16.3 Hardware and CFCC sysplex enhancements

We discuss a number of processor and CF hardware enhancements supported by z/OS sysplex functions in the following sections.

16.3.1 Cascaded FICON director switch support

Cascaded connections between FICON switches are supported in z/OS V1R4. This means that a control unit to a FICON channel path connection can run through two or more FICON switches. Therefore, a switch address must be specified for the single FICON switches of the fabric. HCM also supports explicit upgrading from one-byte link addresses to two-byte link addresses, as well as downgrading of two-byte link addresses to one-byte link addresses via a new utility.

16.3.2 CF Request Time Ordering (Sysplex Timer connectivity to CFs)

As processor, CF, and CF link technologies have improved over the years, the time synchronization tolerance between systems in your Parallel Sysplex has become more rigorous. At the same time distances between components of your sysplex such as CFs and processors have potentially increased, especially if a multi-site sysplex has been implemented. Sysplex component distances to accommodate technologies such as GDPS® also contribute to this timing-related sensitivity.

CF Request Time Ordering algorithm

The “CF Request Time Ordering” (also referred to as “Message Time Ordering” in some IBM literature) is introduced in order to ensure that any exchanges of time stamped information between systems in a sysplex involving the CF observe the correct time ordering. CF Request Time Ordering ensures data integrity in the event the Sysplex Timer is not able to synchronize the Time of Day (TOD) clocks to an accuracy that is smaller than the messaging time between systems. The fastest communication mechanism between processors is the communication through the CF. At the time of writing, synchronous CF requests may have a total service time that is less than 10 microseconds. The CF request service time has improved over the years and will continue to do so.

With the advent of CF Request Time Ordering, time stamps are now included in the message-transfer protocol between the z/OS systems and the CFs. This means that CF requests from the processor to the CF, or from the CFs to the processors, are time stamped. The time stamps are compared with the local TOD clock and if the local time is behind the time stamp, the message process is delayed until the local TOD has caught up to the time stamp value. This ensures that any serialization protocol between two processors involving the CF will be delayed by a sufficient degree to guarantee that the local time stamps obtained under this serialization will have the correct ordering.

As a consequence, a request from a processor to a CF or a response from a CF to a processor may be elongated to ensure that the accuracy of the TOD synchronization is maintained. With the current processors and CFs, this function should seldom, if ever, actually have to delay a CF request to preserve ordering. At the time of writing, there is no external IBM performance instrumentation for this function (SMF, RMF, and so on).

All CF requests and CF request types are subjected to time-ordering when the function is required and enabled. It applies to sync and async CF requests, irrespective of whether they are single-entry or list-form. Also observe that an async CF request *reaches* the CF almost as fast as a sync CF request does. The same is true for the first entry in a list-form command; it will get processed almost as soon as a single-entry CF request.

As a consequence, all of these CF requests can update something in the CF equally quickly. As soon as such an update is made, some process on another system in the sysplex can do a read command to observe it. Thus, in order to preserve the appearance of globally synchronized time, the requirements are the same for all these different types of commands. It does not matter that the async CF request generally takes a lot longer to complete, or that subsequent entries in the list-form command take a lot longer before they execute, than might be true with a sync single-entry command.

CF Request Time Ordering requirements for processors and CFs

CF Request Time Ordering may be enabled for z800 (IBM 2066) and z900 (IBM 2064) processors (model 2C1 to 216). Notice that even though CF Request Time Ordering is available on z800 (any model), it will never be required. In addition to z900 model 2C1 to 216 processors, CF Request Time Ordering will be mandatory for follow-on processors.

CF Request Time Ordering hardware requirements:

CFs must be connected to the same set of Sysplex Timers as the other systems in the Parallel Sysplex (as is always the case in a Parallel Sysplex).

CFs must be connected to z/OS systems with CF Request Time Ordering hardware support installed (which at the time of writing includes z800 and z900 2xx model processors).

CFs must be at least CFLEVEL 12.

When a CF is configured as an ICF on a CF Request Time Ordering supported processor, the CF will require connectivity to the same set of Sysplex Timers that the z/OS systems in the Parallel Sysplex are using for time synchronization. If the ICF is on the same server as a z/OS system of its Parallel Sysplex, no additional Sysplex Timer connectivity is required. If an ICF is configured on a z900 2C1 through 216, which does not host *any* z/OS systems in the same Parallel Sysplex, then Sysplex Timer connectivity is required to the processor that has the ICF. Again notice that this requirement only applies to z900 2xx model and faster processors. z800 has the function installed, but it is *not required*.

For an illustration of additional connections required to the Sysplex Timers for or z900 2xx models with ICFs and non-Parallel Sysplex LPARs, refer to Figure 16-7 on page 280. Note that the “non-Parallel Sysplex LPARs” may very well be part of other Parallel Sysplexes. Also note that external CFs of comparable speeds (not shown in the figure) also have a requirement for sysplex connectivity.

Figure 16-7 z900 Turbo Model with ICF and non-Parallel Sysplex LPARs

CF Request Time Ordering requirements for z/OS

CF Request Time Ordering code is available starting with z/OS V1R4 and is rolled back to OS/390 Release 8 via XCF/XES APAR OW53831.

CF Request Time Ordering displays and messages

Many IXL messages are enhanced to show additional information about CF Request Time Ordering:

- Display XCF and CF commands and messages are updated as follows:

  D XCF,Couple Information is added regarding CF Request Time Ordering to indicate whether CF Request Time Ordering is INSTALLED or NOT-INSTALLED, as shown in Figure 16-8 on page 281. The output indicates whether the system on which the z/OS image is running supports CF Request Time Ordering.

```
IXC357I  10.47.26  DISPLAY XCF       FRAME  1     F      E   SYS=D13ID92
SYSTEM D13ID92 DATA
   INTERVAL   OPNOTIFY     MAXMSG    CLEANUP      RETRY   CLASSLEN
        60              60              3000             60              10
956

   SSUM ACTION      SSUM INTERVAL             WEIGHT
         N/A                      N/A                            N/A

   MAX SUPPORTED CFLEVEL: 12

   MAX SUPPORTED SYSTEM-MANAGED PROCESS LEVEL: 12

   CF REQUEST TIME ORDERING FUNCTION: INSTALLED

   CF REQUEST TIME ORDERING FUNCTION: NOT-INSTALLED
```

Figure 16-8 Sample enhanced IXC257I message (DISPLAY XCF,COUPLE)

D CF Information is added regarding CF Request Time Ordering to indicate whether CF Request Time Ordering is ENABLED AND REQUIRED or NOT-REQUIRED AND ENABLED.

Figure 16-9 shows IXL1501 messages when the CF is connected to a Sysplex Timer.

```
IXL150I hh.mm.ss  DISPLAY CF    FRAME LAST  F   E  SYS=SYS01
COUPLING FACILITY 002064.IBM.51.000000067618
                  PARTITION: 0  CPCID: 00
                  CONTROL UNIT ID: FFFE
NAMED TESTCF
COUPLING FACILITY SPACE UTILIZATION
 ALLOCATED SPACE                 DUMP SPACE UTILIZATION
  STRUCTURES:          0 K        STRUCTURE DUMP TABLES:        0 K
  DUMP SPACE:        256 K                  TABLE COUNT:        0
 FREE SPACE:       25856 K       FREE DUMP SPACE:            256 K
 TOTAL SPACE:       26112 K      TOTAL DUMP SPACE:            256 K
                                MAX REQUESTED DUMP SPACE:        0 K
   VOLATILE:          YES         STORAGE INCREMENT SIZE:      256 K
    CFLEVEL:           12
    CFCC RELEASE 12.00, SERVICE LEVEL 02.05
    BUILT ON 01/10/2002 AT 15:11:00

    CF REQUEST TIME ORDERING: REQUIRED AND ENABLED
    CF REQUEST TIME ORDERING: NOT-REQUIRED AND ENABLED

 COUPLING FACILITY SPACE CONFIGURATION
```

Figure 16-9 Sample enhanced IXL150I message (DISPLAY CF): CF Connected to Sysplex Timer

Archived

```
IXL150I  13.23.09  DISPLAY CF        FRAME  1     F      E   SYS=D13ID93
_____________________________________________________________
COUPLING FACILITY SIMDEV.IBM.EN.CF0100000000
                  PARTITION: 0  CPCID: 00
                  CONTROL UNIT ID: 0001
NAMED TESTCF
CF REQUEST TIME ORDERING: REQUIRED AND NOT-ENABLED

REASON: ETR NOT CONNECTED TO COUPLING FACILITY

REASON: REQUEST TIME ORDERING FUNCTION FAILURE

REASON: ETR NETID MISMATCH - CF ETR NETID: OF

NO COUPLING FACILITY SPACE DATA AVAILABLE
```

Figure 16-10 Sample enhanced IXL150I message (DISPLAY CF): CF not connected to Sysplex Timer

Figure 16-10 shows IXL1501 messages when the CF is not connected to a Sysplex Timer.

CF Request Time Ordering is considered *installed* when the z/OS or OS/390 system is running on a processor with CF Request Time Ordering hardware support installed - z800 and z900 2xx or newer processor.

CF Request Time Ordering is *not installed* when the z/OS or OS/390 system is running on an older processor where CF Request Time Ordering hardware support is not installed.

CF Request Time Ordering is *enabled* when *all* the following are true:

- The CF is running on a processor where CF Request Time Ordering hardware support is installed
- The OS/390 or z/OS system is also running on a processor with CF Request Time Ordering hardware support installed
- The OS/390 or z/OS system has CF Request Time Ordering software support applied
- The same Sysplex Timer is attached to both the CF and the z/OS or OS/390 system.

CF Request Time Ordering is *required* and must be *enabled* for connectivity when:

- Both the CF and the OS/390 and the z/OS system are running on processors where the CF requests may be executed fast enough to warrant CF Request Time Ordering to ensure correct time ordering. CF Request Time Ordering is never required if either the sender or the receiver processor is a G5/G6.

Reasons why CF Request Time Ordering is *not-enabled* include the following:

- A Sysplex Timer is not connected to the CEC on which the CF is running
- A connecting z/OS or OS/390 system and the CEC running the CF are not connected to the same Sysplex Timer. The ETR Net ID connected to the CF will be displayed as shown in Figure 16-10 (`OF`).

  Issue a **`D ETR`** command to display the ETR Net ID of the Sysplex Timer connected to the z/OS or OS/390 system.
- CF Request Time Ordering encountered a failure. The Sysplex Timers attached to the CF and z/OS or OS/390 systems may be out of sync.

Note: When CF Request Time Ordering is required but not-enabled, connectivity to the CF will not be allowed.

- Several new console messages contain new information related to CF Request Time Ordering as follows:
  - IXL161I messages are issued at IPL time or whenever the status of the message time ordering function changes, as shown in Figure 16-11.

```
IXL161I CF REQUEST TIME ORDERING: REQUIRED AND ENABLED
             COUPLING FACILITY SIMDEV.IBM.EN.CF0100000000
                                                PARTITION: 0  CPCID: 00

IXL161I CF REQUEST TIME ORDERING: NOT-REQUIRED AND ENABLED
             COUPLING FACILITY SIMDEV.IBM.EN.CF0100000000
                                                PARTITION: 0  CPCID: 00
```

Figure 16-11 IXL161I message: CF Request Time Ordering OK status

  - IXL160I messages indicate that CF Request Time Ordering is required for operations to this CF. All the requirements to enable CF Request Time Ordering are not met.

    This CF will not be allowed to be used by the OS/390 or z/OS system until the error is corrected. This is a highlighted message that will be domed when the error is corrected. An example of the IXL160E message is shown in Figure 16-12.

```
*IXL160E CF REQUEST TIME ORDERING: REQUIRED AND NOT-ENABLED
                 COUPLING FACILITY SIMDEV.IBM.EN.CF0100000000
                                                PARTITION: 0  CPCID: 00
                 REASON: ETR NETID MISMATCH - CF ETR NETID: 0F
                 REASON: ETR NOT CONNECTED TO COUPLING FACILITY
                 REASON: REQUEST TIME ORDERING FUNCTION FAILURE
```

Figure 16-12 IXL160E message: CF Request Time Ordering not-OK status

For more information about enhanced IXL messages, refer to *z/OS MVS System Messages, Volume 10 (IXC-IZP)*, SA22-7640.

16.3.3 Enhanced Parallel Sysplex support in CFLEVEL 11 and CFLEVEL12

CFLEVEL 12 is the zSeries level of CFCC code that supports System-Managed CF Structure Duplexing. CFLEVEL 11 is the G5/G6 level of code that supports duplexing.

All IBM eServer zSeries 900 (2064) and 800 (2066) servers with CFLEVEL 12 or higher provide the following enhancements, supported by z/OS V1R2 or higher:

- 64-bit support within CFCC eliminates the 2 GB “control store” line in the CF. The distinction between “control store” and “non-control store” (also known as data storage) in the CF is eliminated. As a consequence, very large amounts of CF central storage can be used for both CF control and data objects.
- 48 concurrent tasks in CFCC enhances System-Managed CF Structure Duplexing performance. For more information, refer to “System-Managed CF Structure Duplexing” on page 289.

- Support for CF Request Time Ordering providing additional Sysplex Timer connectivity. For more information, refer to 16.3.2, “CF Request Time Ordering (Sysplex Timer connectivity to CFs)” on page 278.
- DB2 performance enhancements. For more information, refer to 16.2.1, “XES DB2 Data sharing performance improvement” on page 271.

Attention: When migrating CF levels, some structure sizes might need to be increased to support new function. For example, when you upgrade from CFLEVEL 9 to CFLEVEL 10, the required size of the structure might increase by up to 768 KB.

At the time of writing, there is no simple rule of thumb that will generalize the increased structure size requirements for CFLEVEL 12. Some structures may see no increase. Some structures may see a substantial 50% (or higher) increase in structure size.

These adjustments can have an impact when the z/OS system allocates structures or copies structures from one CF to another at different CFLEVELs. The CF structure sizer tool can size structures for you, and it takes into account the amount of space needed for the current CFLEVELs.

The tool is available on the Internet at:

`http://www.ibm.com/servers/eserver/zseries/cfsizer/`

For the latest information about CFLEVELs, always refer to:

`http://www.ibm.com/servers/eserver/zseries/pso/cftable.html`

We strongly recommend that you apply APAR OW43778 which provides CFLEVEL toleration support for CF structure size changes that are necessary when allocating structures during rebuild processes. Refer to the APAR text for more information.

16.3.4 zSeries GDPS/PPRC hyperswap function

The GDPS/PPRC hyperswap function is designed to broaden the continuous availability attributes of GDPS/PPRC solutions by extending the Parallel Sysplex *redundancy to disk subsystems*. The hyperswap function provides the ability to transparently swap all primary PPRC disk subsystems with the corresponding secondary PPRC disk subsystems for a *planned* switch (DASD or site) reconfiguration.

Figure 16-13 on page 285 shows a GPDS(PPRC), which is an IBM multiple-site application availability solution. In summary, GDPS (or Geographically Dispersed Parallel Sysplex™) provides near-continuous application and data availability for both planned and unplanned reconfigurations. In case of a disaster, the GPDS solution is designed to ensure data consistency and integrity with no loss of data. The GDPS solution is application-independent and offers an operational single point of control.

Figure 16-13 Geographically Dispersed Parallel Sysplex

GDPS is offered as an IBM service offering and has been generally available since November 1998. For more information, contact your local IBM representative or:

gdps@us.ibm.com

A GDPS white paper is available at:

http://www.ibm.com/servers/eserver/zseries/library/whitepapers/gf225114.html

Hyperswap elimination of DASD single point of failure

The hyperswap function is designed to deliver complete automation, allowing all aspects of a site or DASD switch to be controlled via GDPS from a single point of control. Prior to the advent of hyperswap, DASD may be a single point of failure even when DASDs are mirrored. The DASD single point of failure may be eliminated with hyperswap.

The important ability to re-synchronize incremental disk data changes between primary/secondary PPRC disks, in both directions, is provided as part of the hyperswap function. GDPS exploitation of hyperswap function is expected to be progressively enhanced.

For a schematic overview of the hyperswap function, refer to Figure 16-14 on page 286. The figure illustrates how applications may access the PPRC copy after a hyperswap via the same set of UCBs.

Archived

Figure 16-14 Hyperswap example

Hyperswap is delivered in stages, as follows.

Planned hyperswap

This allows you to transparently (that is, without having to quiesce the applications) switch primary PPRC disk subsystems with the secondary PPRC disk subsystems for a *planned* reconfiguration. Planned hyperswap provides the ability to perform DASD configuration maintenance and planned site maintenance. Large configurations can be supported, as hyperswap is designed to scale to swap large numbers of disk devices within a few minutes.

Unplanned hyperswap

This delivers the ability for GDPS/PPRC to transparently switch to the secondary PPRC disk subsystems in the event of *unplanned* outages of the primary PPRC disk subsystems, without data loss and without requiring an IPL.

zSeries GDPS/PPRC hyperswap value

The overall availability of your Parallel Sysplex may be significantly improved with hyperswap.

The hyperswap function delivers significantly faster primary/secondary PPRC DASD swaps both for planned and unplanned DASD reconfiguration activities. The value of hyperswap may be obtained both in multi-site and single-site environments, as long as DASD is configured to exploit the PPRC function.

The value of hyperswap is depicted in Figure 16-15. The bar chart shows sample elapsed time elements for various site reconfiguration activities including:

- Shut down sysplex
- Reset systems
- CF, DASD, and Tape switch
- Load systems
- Restart sysplex

Prior to hyperswap, the time to accomplish planned or unplanned switches could take between one and two hours. Most of these activities can virtually be eliminated with the advent of the hyperswap function, thus reducing the switch time to minutes, as illustrated in Figure 16-15.

Figure 16-15 Sample GDPS site reconfiguration - where the time is spent

Important: The GDPS and the hyperswap function are designed to provide the ability to perform disk configuration maintenance and planned site maintenance without requiring any applications to be quiesced.

zSeries GDPS/PPRC hyperswap prerequisites

The hyperswap enhancement is an example of integration of IBM eServer and IBM TotalStorage technologies including associated elements of hardware, software and microcode. The following is a list of the prerequisites for hyperswap:

- GDPS:
  - GDPS/PPRC code at the GDPS V2R7 or higher level
- z/OS or OS/390
  - New function (development APAR 4Q2002) is available for OS/390 V2R10, z/OS V1R1 or higher.
  - Initial support is limited to JES2 systems.
  - Parallel Sysplex is highly recommended to implement the GRS Star function

Note: Although strongly recommended, Parallel Sysplex is not a must; the prerequisite is to convert all RESERVEs to global enqueues. This may be done by use of the z/OS V1R2 and higher release RNL masking function (this support is not available in OS/390 V2R10 or z/OS V1R1).

In a Parallel Sysplex, we highly recommend using the GRS Star function for performance reasons.

- DASD:

- Disk subsystems must support PPRC L3 architecture (PPRC extended query).
- ESS PPRC L4 architectural extension provides improved performance for hyperswap function.
- All disk volumes (except volumes that contain CDSs) must be “PPRC’ed” and in duplex mode.
- PPRC must be symmetrically configured (must have a one-to-one correspondence between each primary PPRC SSID and secondary PPRC SSID).
- Hyperswap devices cannot attach to systems outside the Parallel Sysplex.
- Production systems must have sufficient channel bandwidth to both the primary and the secondary PPRC disk subsystems.

Important: Because of bandwidth considerations, it is recommended that you use FICON to disk attachment. Also note that use of FICON reduces cross-site fiber requirements.

For more information about GDPS and hyperswap, contact your local IBM representative or:

gdps@us.ibm.com

GDPS/PPRC hyperswap summary

In summary, the hyperswap function is a significant enhancement to the enterprise-class mission critical zSeries data center environment. With the hyperswap function, the speed of GDPS DASD and site reconfigurations (planned and unplanned) is significantly improved. Full planned outage support for this function is available in the GDPS 2.7 code base, using z/OS and appropriate disk subsystem hardware.

In addition to hyperswap, further enhancements to GDPS 2.7 includes GDPS/XRC peer-to-peer VTS support and a new HMC automation interface through SCLP.

The ultimate objective of the hyperswap function is to provide both planned and unplanned outage multiple site continuous availability. The prerequisite additional functionality will be delivered in future GDPS code releases and ESS microcode enhancements.

Archived

17

System-Managed CF Structure Duplexing

This chapter covers the following topics:

- An introduction to System-Managed CF Structure Duplexing
- A short discussion about how System-Managed CF Structure Duplexing works
- A discussion of the performance considerations for System-Managed CF Structure Duplexing
- A list of the components that must be considered when doing capacity planning in a System-Managed CF Structure Duplexing environment
- Planning for the implementation of System-Managed CF Structure Duplexing
- A step-by-step guide to setting up System-Managed CF Structure Duplexing
- A list of the exploiters of System-Managed CF Structure Duplexing and information about the benefits for that exploiter and the performance considerations
- Operational changes related to using System-Managed CF Structure Duplexing

© Copyright IBM Corp. 2003

17.1 Introduction

System-Managed CF Structure Duplexing is the most significant sysplex-related improvement in z/OS 1.2. It continues along the road of maintaining z/OS as the most versatile and highly available computing platform. Specifically, it delivers the following enhancements:

- It contributes to application availability by decreasing recovery time following a Coupling Facility (CF) or connectivity failure.
- It enables the use of application enablers, such as the CICS Temporary Storage structure and MQSeries shared non-persistent messages, in a high availability environment.
- It contributes to system ease-of-use by providing a consistent way to set up and manage structures. This is especially beneficial for products like JES2, which previously had different ways of managing the structure depending on whether a structure change was planned or unplanned. The superior ease-of-use should lead to improved system availability due to fewer human errors.
- It contributes to further sysplex exploitation by making it easier for products to exploit the benefits of Parallel Sysplex without significant additional programming effort by the product developers.
- It contributes to an improved cost of ownership for sysplex operations by potentially making it possible to exploit data sharing without the cost of standalone Coupling Facilities.

More information about System-Managed CF Structure Duplexing is available in a white paper entitled *System Managed CF Structure Duplexing*, available at:

http://www.ibm.com/servers/eserver/zseries/library/techpapers/gm130103.html

There is also an updated *Coupling Facility Configuration Options* white paper, available at:

http://www.ibm.com/servers/eserver/zseries/library/techpapers/gf225042.html

17.1.1 System-Managed versus User-Managed Duplexing

What is the difference between the new System-Managed CF Structure Duplexing and the structure duplexing that was already available and is used by DB2 for its Group Buffer Pool structures? To understand this, you need to go back and look at the difference between User-Managed Rebuild and System-Managed Rebuild.

The rebuild process that was available prior to OS/390 V2R8, now known as User-Managed Rebuild, required complex programming on the part of the product that was using the CF structure. The entire rebuild process had to be managed by the product. This included tasks such as coordinating activity between all the connectors to stop any accesses to the structure until the rebuild is complete; working with other connectors of the structure to decide who will rebuild which parts of the structure content; recovering from unexpected events during the rebuild process; and handling any errors that may arise during the process. In fact, the *z/OS MVS Sysplex Services Guide* spends 39 pages describing the User-Managed Rebuild process!

By comparison, System-Managed Rebuild, delivered with OS/390 V2R8 and CFLevel 8, takes all this responsibility away from the product. With System-Managed Rebuild support, all the decision making and error-handling is managed by XCF and XES. Rather than the exploiter having to provide code to handle all these situations, the exploiter simply has to

respond to a notification from XES, saying that it understands that the structure is being rebuilt. The next notification it receives is when the rebuild is complete. This process is also described in the *z/OS MVS Sysplex Services Guide.* In this case, the description takes just 13 pages.

Prior to the availability of System-Managed CF Structure Duplexing, the one drawback of System-Managed Rebuild is that it cannot handle rebuild from hard failure situations, such as a broken CF or a loss of connectivity. Because System-Managed CF Structure Duplexing provides the ability to have *two* copies of a structure, the loss of a CF or connectivity to a CF no longer requires that the structure be rebuilt. Instead, the system automatically reverts back to simplex mode for the affected structure, dropping the affected copy of the structure in the process.

Now that you understand the difference between User-Managed and System-Managed Rebuild, we can extend the concept to structure duplexing. User-Managed Duplexing, as used by DB2 for its Group Buffer Pool structures, requires explicit support by the exploiter. While XCF takes responsibility for allocating the second structure instance, DB2 then is responsible for copying the contents from the primary to the secondary structure instance, and subsequently keeping the two copies synchronized. For each update to a GBP structure, DB2 has to do two writes: a synchronous write to the primary structure instance, and an asynchronous write to the secondary instance. Having issued the writes, DB2 then is responsible for ensuring that the writes completed successfully before it continues processing. Just as with User-Managed Rebuild, you can see that User-Managed Duplexing puts a considerable onus on the exploiter.

By comparison, System-Managed CF Structure Duplexing is like System-Managed Rebuild in that all the processing associated with copying initially and keeping the two structure instances synchronized is handled by XES, transparently to the exploiter. Whereas User-Managed Duplexing required the exploiter to issue two write requests every time a structure update was done (as we saw for DB2), System-Managed CF Structure Duplexing automatically looks after sending the update request to both structure instances, and ensuring that the requests completed successfully. The exploiter need not even be aware that its structure has been duplexed.

Any exploiter that supports System-Managed Rebuild automatically gets System-Managed CF Structure Duplexing support for “free”. That is, once the exploiter has added the code in support of System-Managed Rebuild, no additional programming is required to be able to benefit from System-Managed CF Structure Duplexing.

Your next question may be “how does this lower the cost of computing for doing data sharing”? IBM has always recommended that certain structures be kept isolated from the systems they are connected to when you are doing data sharing; an example would be the IRLM lock structure used for IMS or DB2. The reason for this is that if the CF containing the lock structure fails, the lock structure is rebuilt using in-storage information from all of the connected IRLM subsystems. If you have a failure that causes you to lose both the lock structure *and* one of the connected IRLMs, you no longer have all the information required to rebuild the structure. In this case, to recover the lock information, you must restart all the database managers (IMS or DB2) in the data sharing group. This is why it is recommended to keep the lock structure failure isolated from all the connected systems.

Figure 17-1 Use of ICFs vs. standalone CFs

However, if you can duplex the lock structure, it is now acceptable to have the lock structure in the same failure domain as some of the attached subsystems. This makes the use of ICFs (which are less costly than standalone CFs) in a data sharing environment a viable alternative. This is shown in Figure 17-1.

17.2 How System-Managed CF Structure Duplexing works

In order to determine where System-Managed CF Structure Duplexing can be of benefit to you, and to be able to manage it correctly, it is important to understand at least a little about how it works. Therefore, in this section we describe how a duplexed copy of a structure is established, how the two copies are kept in synch, and how duplexing is terminated.

17.2.1 Starting duplexing

The process for initiating a System-Managed Duplex copy of a structure is very similar to how a System-Managed Rebuild starts out. When a request is made to XCF to start the duplexing operation, XCF initiates a duplexing rebuild of the primary structure. This consists of the following steps:

- Notification is sent to all the connectors of the existing structure, informing them that access to the structure is going to be temporarily quiesced.
- Once all the connectors respond to this, access to the structure is quiesced, and a new structure is allocated in one of the other CFs in the preference list for that structure. The classification rules that XES uses to decide which CF to select have been modified as part of the implementation of System-Managed CF Structure Duplexing, and are described in 17.6.7, “CF selection changes” on page 316.
- All of the connectors to the existing structure are connected to the new structure instance.
- All or some of the systems that are connected to the original structure instance take part in copying the required contents from the original structure instance to the new instance.

If this was a normal rebuild, the system would proceed to notify all connectors that the new instance is available and resume access. It would then remove the connections to the original structure instance and delete it. However, because the intent is to keep two copies of the structure, XCF does not remove the old connections and structure instance. Instead, it notifies the connectors that the structure is once again available for use and unquiesces the structure. The next section discusses how it keeps both copies in synch.

17.2.2 Maintaining a duplexed structure

For User-Managed Duplexing, it is the responsibility of the connector to ensure that both copies of the structure are kept in synch. Any changes to the primary structure instance must be reflected in the secondary structure instance by the connector.

However, in System-Managed CF Structure Duplexing, the system takes responsibility for keeping the two copies in synch. In fact, the connector need not even be aware that there are two copies of the structure (although this information is returned to the connector when access to the structure is resumed after setting up the duplex copy).

In User-Managed Duplexing, as implemented by DB2 for its Group Buffer Pools (GBPs), when DB2 wants to make an update to a GBP, it issues an asynchronous write to the secondary structure, then a synchronous write to the primary structure, and then it waits for both writes to complete. Clearly, DB2 is aware that there are two copies of the structure, and it must handle the situation should one of the writes not finish successfully.

In System-Managed CF Structure Duplexing, the connector only issues a single write, exactly as they would for a non-duplexed structure. If the structure is System-Managed Duplexed, XES will then issue the write to both CFs (this is transparent to the connector).

When both CFs receive the request, the two CFs communicate with each other to coordinate the processing of the request. This communication is done using CF-to-CF links that are a new requirement with System-Managed CF Structure Duplexing. The reason for this communication is to ensure atomicity of the command execution. Prior to System-Managed CF Structure Duplexing, the CF architecture guaranteed that once the CF started changing some data in its storage, that data could not be accessed by any other request until the change was complete. This integrity is implemented using locks and latches in the CF.

However, when the data resides in two CFs, and is being operated on by two commands executing independently of each other, some mechanism is required to continue to guarantee that integrity. This is provided by the two CFs exchanging signals before they start executing on the data. From that time until they exchange signals indicating the end of that processing, that data cannot be accessed by any other requests. This additional processing within the CFs obviously has an impact on CF response times for the duplexed structures. This is discussed further in 17.3, “Performance considerations” on page 297.

When the CFs complete processing the request, they both send back the response to the requesting system. When z/OS receives the reply from both the CFs, it returns control to the requestor.

Note that all of this processing only takes place for *update* requests. Read requests will all be sent to the primary CF structure as they are today (structures duplexed using User-Managed Duplexing also do all their reads from the primary structure). And not all writes are necessarily sent to the duplex copy. For example, cache structure directories are only kept in the primary structure, so updates of registrations in cache directories are sent only to the

primary structures. However, all data that is required for recovery purposes will be duplexed. In the case of the directory information, for example, if the primary structure is lost, the CF containing the secondary structure will automatically send a notification to all the connectors of that structure telling them to invalidate any local buffers associated with that structure.

Changes to SYNCH and ASYNCH processing

For customers that have CFs that are remote from some of the connected systems, sending synchronous requests to that CF could have a significant impact on system overhead. The cost of sending signals to a remote CF is roughly 10 microseconds per kilometer, round trip. So if you were to send a synchronous request to a CF that is 20 km away, the requesting CP would spin for an additional 200 microseconds for every request. For a workload that generates large numbers of synchronous requests, this would be a significant impact on the overhead of using that CF.

Note: It is important to understand exactly what is meant by synchronous or asynchronous processing. In both cases, the communication between the operating system and the CF is similar. The operating system gets a subchannel, sends the request to the CF, and holds the subchannel until the CF responds.

The difference is that for *synchronous* requests, XES holds onto the CP and spins, waiting for a response from the CF. For an *asynchronous* request, XES releases the CP so the CP can be used to run other work. XES will then periodically check to see if the CF has responded. The response time that RMF reports for asynchronous requests includes this time while the CP is being used for other processing.

To address this issue, the handling of CF requests by XES has been modified in z/OS 1.2. Previously, the determination of whether a request was synchronous or not was based on a number of things, including:

- The type of request - for example, all lock requests are synchronous.
- The availability of subchannel resources.
- If XES knows that the target CF is a significantly slower processor type (that will deliver long response times), XES will issue the request asynchronously.
- The amount of data in the request. Requests containing large amounts of data are handled asynchronously.
- What was specified by the requestor. When sending a request to a CF, the requestor can specify whether it would like the request to be handled synchronously or asynchronously (although this might be overridden by XES based on one of the above rules).

None of the rules, however, took distance or actual response times into account. In z/OS 1.2, the XES algorithms have been changed. Now, XES maintains a table of actual response times for different request types for every CF and pair of duplexed CFs.

It also maintains a table showing the response time at which it is more efficient to process the request asynchronously rather than synchronously. This calculation is based on the response time for the CF and the speed of the CPC that z/OS is running on. When it is sending a request to a CF, it checks the response time for that type of request and decides whether it is more efficient to issue that request synchronously or asynchronously.

For requests to System-Managed Duplexed structures, the requests to both CFs are issued in the same manner—either both requests are issued synchronously, or both are issued asynchronously. Just as for a simplex structure, this same process is followed for each request, so some requests to a System-Managed Duplexed structure may be handled synchronously and some asynchronously, depending on the response times being delivered by the CF at that time.

To illustrate this, let us look at two examples. In the first example, z/OS is running on a fast CPC, communicating with a slow CF that is a long distance away. When IRLM issues a request to its lock structure, XES knows approximately what the response time will be. Based on this information, XES calculates that the CPC could execute 100K instructions in the time it will be waiting for the response from the CF. It also knows that if it issues the request asynchronously, it will use, for example, 30K instructions.

In this case, if it converts the request to be asynchronous rather than synchronous, z/OS can do 70K instructions for someone else while it is waiting for the CF to respond. So, converting the request to an asynchronous one in this situation will decrease the overhead of using the CF (rather than spending 100K instructions worth of CPC to process the request, only 30K instructions are used).

In another example, z/OS is running on a 9672 G5, communicating with a 2064-100 CF that is right beside the 9672. In this case, you have a relatively slow CPC talking to a very fast CF. Now when IRLM issues the request, the CPC might be able to execute just 5K instructions in the time it takes the CF to respond. If XES were to convert the request to be asynchronous, it would take 30K instructions - 25K *more* than issuing it synchronously. In this case, XES will *not* convert the request because it is more efficient to issue it synchronously. (Note that these are not the actual numbers of instructions; they are used solely to illustrate the logic of this new algorithm.)

While this change may lead to longer response times for some CF requests, it should decrease the data sharing overhead (because the CP isn't spinning, waiting for a response to a long synchronous request), and ultimately allow the system to process more work. The tradeoff is greater throughput in return for poorer response times for some requests.

Inter-CF communication

As stated, the processing in the CFs of all update requests against System-Managed Duplexed structures is bounded by coordinating signals between the CFs containing the two structures.

There is no data sent between the CFs. The links are *only* used for the coordinating signals. However, if the link is unavailable for some reason, the coordination cannot take place, and the integrity of the data in the structure cannot be guaranteed. Therefore, if there is a connectivity failure between the two CFs, any structures in either of the CFs that are System-Managed Duplexed will immediately drop back to simplex mode.

In relation to distance considerations between the CFs, the inter-CF communication can be either synchronous or asynchronous depending on the particular request. Therefore, there may be some additional CF processing cost as the CF spins waiting for an acknowledgement to the signal it sent to the other CF.

17.2.3 Stopping duplexing

Should you wish to stop duplexing a particular structure, the command to do this is the same command that is used to stop duplexing for a User-Managed Duplexing structure; specifically:

```
SETXCF STOP,REBUILD,DUPLEX,STRNM=structure_name,KEEP=NEW|OLD
```

Specifying KEEP=OLD will cause XCF to stop duplexing all update requests for that structure, remove all connections to the *secondary* structure, and delete the secondary structure. Similarly, specifying KEEP=NEW will cause XCF to remove the primary structure and keep the secondary structure.

Should you wish to take a CF out of service, perhaps for scheduled maintenance, you can stop all duplexing relating to structures in that CF (both User-Managed and System-Managed) with a single command:

```
SETXCF STOP,REBUILD,DUPLEX,CFNM=cf_name
```

Any primary structures in the named CF will be deleted and processing will continue in simplex mode, using what was previously the secondary structure in the alternate CF. Similarly, any secondary structures in the named CF will be deleted and processing for those structures will continue in simplex mode using the primary structure in the alternate CF.

17.2.4 Error recovery

Faster recovery from an error is one of the major advantages of System-Managed CF Structure Duplexing. The errors that are most likely to affect a CF structure are a structure failure (which is very rare), a failure of the CF containing the structure, or a loss of connectivity from one or more of the connected systems to the CF containing the structure.

Prior to System-Managed CF Structure Duplexing, different exploiters had different ways of recovering from such failures, for example:

- IRLM would rebuild the structure contents in an alternate CF using information held in the virtual storage of each of the connected IRLM address spaces.
- JES2 would typically be set up to revert to using DASD for its checkpoint information.
- CICS was unable to recover some of its structures (Shared Temporary Storage, CF Data Tables, and Named Counter Server), so all the data in one of those structures at the time of the failure would be lost.

Depending on the structure then, you can see that the loss of a structure can have a varying impact. The best case is that the structure contents are re-constituted from the storage of the connected subsystems. This recovery typically takes seconds to tens of seconds for a single structure. The worst case is that data is lost and must be recreated or reconstructed from a set of logs.

However, if a structure is duplexed, there is a copy kept in each of two CFs (that should be failure-independent of each other). Now, if the CF containing that structure (or connectivity to that CF) is lost, there is a short pause while the structure reverts to simplex mode and the connections to the failed CF are cleaned up, after which processing continues. This processing will typically take a few seconds for a single structure. Depending on which structures are being used, this can have a significant availability benefit.

If there are three CFs in the Parallel Sysplex, and all three CFs are listed on the preference list for a duplexed structure (that has specified DUPLEX(ENABLED)), the system will automatically re-duplex the structure into the third CF, assuming that CF has the required connectivity, both to the systems in the Parallel Sysplex and also to the CF containing the primary instance of the structure. For maximum availability and flexibility, it is highly recommended that all CFs are attached to all systems in the Parallel Sysplex and also to all of the other CFs in the Parallel Sysplex.

Attention: z/OS 1.2 introduced a change in the way re-duplexing is handled. Prior to z/OS 1.2, if you stop duplexing (using the SETXCF STOP,DUPLEX command) for a structure that has DUPLEX(ENABLED), XES will not attempt to put the structure back in duplex mode until there is some sort of event, such as a CF coming online, or a new policy being started.

In z/OS 1.2, XES will automatically try to re-duplex structures approximately every 10 minutes. Therefore if you wish to stop duplexing a structure for an extended time, you should update the policy to make the structure DUPLEX(ALLOWED) or DUPLEX(DISABLED).

In the case of a connectivity failure, where one or more systems lose all connectivity to a CF, recovery typically consists of the surviving systems copying the data from the old structure instance into a new instance in a CF that has full connectivity. In this case the recovery time can take from seconds to minutes, depending on the size of the structures and the speed of the hardware (CFs, CPCs, and links). Once again, having two copies of the structure removes the need to copy any data—all the data needed for continuing processing is available in both CFs. Any affected structures will drop out of duplex mode and the structure instance in the CF that has lost connectivity will be deleted.

This processing takes just seconds. The recovery time is a factor of the number of connectors to the structure, the number of systems, and the performance of the DASD containing the CFRM CDS: unlike normal rebuild processing, it is independent of the size and contents of the structure. As in the case of a CF failure, if there is a third CF with full connectivity (and it is specified on the preference list for the structure and the structure has specified DUPLEX(ENABLED)), the structure will be automatically be re-duplexed into that CF.

In a System-Managed CF Structure Duplexing environment, there is one new component that can fail that did not exist prior to this: the link between the two CFs. For availability reasons, we recommend that there should be two links between each set of CFs (or four links, two Senders and two Receivers, if you are not using peer links). However, if you are unfortunate enough to lose all the links between the two CFs, the impact is that all System-Managed Duplexed structures in either of the affected CFs will drop back to simplex mode. Again, if there is a third CF, and the structure specifies DUPLEX(ENABLED), the structures will be re-duplexed into the third CF, assuming that CF has the required connectivity and is listed on the preference list for the affected structures.

Important: If a structure that supported structure recovery prior to duplexing reverts to simplex mode, it will still have the same recovery capability characteristics that it had before it was duplexed.

However, structures that did not support structure recovery will be exposed to the same restrictions that existed prior to duplexing if they revert to simplex mode. For example, structures that do not support User-Managed Rebuild (CICS Shared Temp Storage, for example) will be exposed until a duplex copy is re-established.

17.3 Performance considerations

There are two aspects of system performance that must be considered when deciding whether the performance implications of duplexing a structure are acceptable in your environment. This first of these is the impact of CF response times on transaction response times or on batch job elapsed times. Generally speaking, the amount of time that is spent communicating with the CF is a very small percentage of the total elapsed time of a job or

transaction. In tests we ran, the elapsed time for the batch jobs only increased by a small percentage, even though there was a significant increase in CF response times. These jobs were unusual in that they did very little processing apart from issuing CF requests, so the impact on them was larger than would normally be expected. Detailed performance information will be made available after IBM has completed performance testing of System-Managed CF Structure Duplexing.

The other aspect is the additional capacity that is used as a result of duplexing the requests. It is said that there is no such thing as a “free lunch”; this is true for System-Managed CF Structure Duplexing as well. Any time additional processing is involved, there has to be some impact, all other things being equal. In this section we first discuss the components that are affected by System-Managed CF Structure Duplexing, and then discuss how this relates to your environment.

17.3.1 CF CPU utilization

When comparing the resources used to process a simplex structure versus a duplexed one, it is obvious that there will be an increase in CPU utilization for the CF containing the secondary structure. After all, that CF now has to process requests for a structure that did not exist in it prior to duplexing. The precise impact will depend not just on the level of write activity to the structure being duplexed, but also on the structure type; for example, with a cache structure, it is possible that some requests will update the primary structure but not be duplexed to the secondary. This is also possible, though less frequently, for lock and list structures.

In addition to the CF containing the secondary structure instance, there will also be an impact on the CF containing the primary structure instance. There is now additional processing required to send the coordinating signals to the CF containing the secondary structure, both before and after processing the actual request. The effect of this processing is twofold:

1. If the CF CPU utilization is already close to or exceeding 50%, the addition of duplexing *may* result in a requirement for additional CF CPU capacity, depending on the level of activity to the duplexed structures. For a discussion of the impact of CF CPU utilization, refer to the IBM Redbook *OS/390 MVS Parallel Sysplex Capacity Planning*, SG24-4680.
2. As CPU utilization in the CF increases, there may be some effect on the response time for *all* structures in that CF because of the increased queueing for access to the CPU.

A rule of thumb, based on pre-GA levels of hardware and software, is that the amount of CF CPU consumed in processing the update requests to the primary structure will increase by a factor of 2 to 2.5.

So, for a structure like the IRLM lock structure, where nearly all requests get duplexed, if enabling System-Managed duplexing of that structure causes CF CPU utilization to increase by 5%, you can estimate that the total CF utilization associated with the IRLM structure is about 10%. The CF containing the secondary structure will have to do the same amount of work as that associated with the update requests to the primary instance, so in this example you would expect the utilization of the CF containing the secondary structure instance to increase by roughly 10%.

Tip: Plan for the additional capacity that may be required. Consider adding more CF CPU capacity, preferably dedicated CPs, with the intent of keeping CF CPU utilization below 50%.

17.3.2 Distance between CPC and CFs

You must remember that the CF request, as issued by the connector, cannot complete until XES gets a response from *both* CFs. If one of the CFs is a lot further from the CPC than the other, the observed CF response time will be gated by the time to communicate with the more distant of the two CFs. If the CF containing the structure today (prior to duplexing) is further from the CPC than the CF that will contain the secondary structure, there should be no impact directly attributable to the distance from the CPC to the CFs. Both requests are issued in parallel, so XES does not have to wait for one CF to receive the request before the request to the other CF is issued.

On the other hand, if the CF that will contain the secondary structure is significantly further away from the CPC than the CF containing the structure today, the requests to the secondary CF will take longer, and may even result in the duplexed requests being converted to asynchronous requests, resulting in significantly longer response times (as seen by the connector) for the duplexed requests.

Tip: As in simplex mode, the best performance is achieved when all the CFs are physically close to the sender CPC.

17.3.3 Effect of synch/asynch conversion

"Changes to SYNCH and ASYNCH processing" on page 294 described the new algorithms in XES to decide whether a CF request should be handled synchronously or asynchronously.

Prior to these changes, CPU overhead would increase proportionally in line with CF response times for synchronous requests. As a result of these changes, CPU overhead will tend to level off as CF response times increase past the point where XES decides to handle the request asynchronously rather than synchronously, as shown in Figure 17-2. This has a positive effect on CPU overhead. However, the response time of the CF requests that are being converted to asynchronous requests will increase significantly. In most cases, the increased CF response time should not have a noticeable impact on online transaction response times.

Figure 17-2 Effect of new sync/asynch algorithm on system overhead

For example, assume that the break-even point between synch and asynch processing for a particular structure is 100 microseconds and that the average asynch response time (as reported in RMF) is 250 microseconds. If the CF response time (as seen at the subchannel) increases from 95 microseconds to 105 microseconds, the requests will start to be processed asynchronously. In the eyes of the requestor, the response time has suddenly increased from 95 to 250 microseconds. Note that the threshold at which requests get converted to asynchronous is a factor of the type of request and the speed of the CPC that z/OS is running on—it is *not* a hard-coded number.

The reason why this is a consideration for System-Managed CF Structure Duplexing is that the response time for System-Managed Duplexed structures will increase compared to simplex structures. Depending on the response time being achieved for the simplex structure, it is possible that enabling duplexing may push the response time over the threshold that causes the request to be handled asynchronously.

Coming back to the preceding example, if you were having 80 microsecond response times on a simplex structure, duplexing that structure could push the response time for requests to that structure over the threshold, resulting in RMF-reported response times for that structure increasing to 250 microseconds. At the same time, however, you might also see a decrease in the CPU overhead associated with those requests. It is worth noting that there are different thresholds for simplex and duplex structures. Because simplex requests have a lower CPU cost, the threshold at which those requests are converted to asynch is lower than the threshold for duplexed requests.

Tip: For System-Managed Duplexed structures, you should monitor the percent of requests for a structure that are being processed asynchronously. If the percentage starts increasing, it is an indicator that the underlying CF response times are increasing. Because of the significant jump in reported CF response times for a structure when a request is handled asynchronously, you should use this increase as a trigger that CF performance should be analyzed. “Capacity planning considerations” on page 304 contains a discussion of the things that can impact CF response times.

Another consideration is if you increase the power of the CPC containing z/OS. If the speed of the z/OS CPC increases, and all other things stay the same, the threshold at which it becomes more efficient to handle the request asynchronously will *decrease*. As a result, you may find that when you upgrade a CPC, some reported CF response times (those that were previously just below the threshold to get converted) will *increase* as more requests get converted from synchronous to asynchronous.

17.3.4 Distance between CFs

A new consideration for System-Managed CF Structure Duplexing is the distance between the CFs. Remember that before an update request against a System-Managed Duplexed structure is carried out, synchronizing signals must be sent between the primary and secondary CFs. These signals are sent independently of each other, so it is possible to get some degree of overlap.

Similarly, before the CFs send their response back to the CPC, synchronizing signals must again be sent between the CFs. Every 1km of distance between the two CFs will add roughly 10 microseconds to the response time of each of those signals. This in turn accrues to the response time of the underlying duplexed CF request as seen by XES.

Figure 17-3 Flow of signals for System-Managed Duplexed structures

If you take the configuration in Figure 17-3 on page 301 as an example, and assume the CFs are 2 km apart, the distance would impact the response time for a duplexed request as follows:

- Steps 2b and 6b (request sent to CF and response back when request is complete): assuming that CF2 is the remote CF and that the simplex structure was previously in CF1, 20 microseconds would be added to the request response time because of the distance.
- Steps 3a and 3b (ready to start signals from each CF): 20 to 40 microseconds, depending on the amount of overlap.
- Steps 5a and 5b (ready to complete signals after the request has been completed): 20 to 40 microseconds, depending on the amount of overlap.

So, if the CFs are 2 km apart, the impact of that *distance* for System-Managed Duplexed structures would be somewhere between 60 and 100 microseconds, compared to two CFs that are within meters of each other. If the simplex structure was in the remote CF prior to duplexing, the 20 microseconds for steps 2b and 6b already exist, so the impact of duplexing would be 40 to 80 microseconds.

Tip: For best performance, duplex between CFs that are physically close to each other and to the systems issuing requests to them.

17.3.5 Link speeds

Just as for simplex structures, the speed of the CF links can have a significant impact on CF response times, especially for large data transfers. However, in addition to the greater bandwidth, there are also benefits to be had from IC or ICB connections due to the reduced latency on those link types. You should *always* use ICs to communicate between z/OS and a CF in the same CPC.

This also applies to the CF-to-CF links—even though the amount of information transferred on the link is small, the reduced latency on ICB links provides a noticeable difference in response times compared to ISC links. (Obviously IC links could not be used for the CF-to-CF links, as the two CFs should be in different CPCs.)

17.3.6 CP speeds

The final component that is involved in processing the System-Managed Duplexed structure requests is the CPC that the operating system is running on. While most of the additional processing takes place in the CFs, there is also additional processing in XES to split the requests across the two CFs and to compare the responses from the CFs after the request has been processed. The impact that this additional processing has on response times is related to:

- The speed of the CPs in the CPC the operating system is running on.
- The structure type. The additional processing in XES to split the signals varies depending on the structure type (lock vs. list vs. cache).
- The percentage of requests that will be duplexed. If the accesses to the structure are update-intensive, there will be a more noticeable increase in XES processing than if the accesses are predominately read-only.

17.3.7 Estimating the impact of System-Managed CF Structure Duplexing

The two most significant variables in determining the impact of System-Managed CF Structure Duplexing are the percentage of requests to a structure that will be duplexed, and the distance between the CPCs and the CFs and between the two CFs.

Unfortunately, it is not possible to accurately determine the percentage of requests that are going to be duplexed until you actually test it in your environment. The reason for this is that, prior to implementing System-Managed CF Structure Duplexing, the number of requests that will be duplexed is not externalized anywhere. RMF only tells you the *total* number of requests; it does not break that down into reads and writes. Even for cache structures, where the number of writes is reported, you do not know how many are directory updates or how many are for unchanged data (as in the case of a store-through cache)—both of these types of writes are not duplexed. Also, the percentage of requests that will be duplexed is workload-dependent, so it is not possible to get accurate numbers in advance.

However, as a general rule of thumb (to be used only until you have actual numbers from your environment), you can assume that 100% of requests to lock structures are duplexed and between 80% and 100% for list structures. The percent for a cache structure is completely workload-dependent, so it is not possible to give a rule of thumb for that structure type—however, in tests conducted in IBM, the amount of requests that were duplexed was generally below 20%.

If you have representative test versions of your applications, you could enable System-Managed CF Structure Duplexing for those structures on a test system. Once System-Managed CF Structure Duplexing is enabled for a structure, you can determine the percentage of requests that are being duplexed by checking the structure statistics for the primary and secondary structure instances in the RMF Coupling Facility Usage Reports as shown in Figure 17-4 on page 303.Figure 17-4 on page 303.

```
                                  C O U P L I N G   F A C I L I T Y   A C T I V I T Y

    z/OS V1R2                SYSPLEX #@$#PLEX            START 09/12/2001-17.05.00           INTERVAL 000
                             RPT VERSION V1R2 RMF        END   09/12/2001-17.10.00           CYCLE 01.000

-----------------------------------------------------------------------------------------------------------
COUPLING FACILITY NAME = FACIL03
TOTAL SAMPLES(AVG) =   299  (MAX) =   300  (MIN) =   298
-----------------------------------------------------------------------------------------------------------
                                              COUPLING  FACILITY  USAGE  SUMMARY
-----------------------------------------------------------------------------------------------------------
STRUCTURE SUMMARY
-----------------------------------------------------------------------------------------------------------

                                                % OF                  % OF      AVG     LST/DIR  DATA
       STRUCTURE                        ALLOC   CF            #      ALL       REQ/    ENTRIES  ELEMENTS
TYPE   NAME              STATUS CHG     SIZE   STORAGE      REQ      REQ       SEC     TOT/CUR  TOT/CUR

LIST   CIC_DFHSHUNT_001  ACTIVE          3M      0.5        0      0.0       0.00     1175     3525
                                                                                          9       64
LIST   I#$#EMHQ          ACTIVE          4M      0.8        0      0.0       0.00     3211     3209
                                                                                          6        5
       I#$#MSGQ          ACTIVE          4M      0.8        0      0.0       0.00     3211     3209
                                                                                          6        5
       IEFAUTOS          ACTIVE          2M      0.3        0      0.0       0.00     2827     2827
                                                                                          0        0
       PSMGAPPL01        ACTIVE  X       5M      1.0     3474     13.5      11.58     1097     6581
                          SEC                                                            995     2000
       PSMGCSQ_ADMIN     ACTIVE         10M      2.0       39      0.2       0.13     9996      20K
                          PRIM                                                            5       80
```

Figure 17-4 RMF Coupling Facility Usage Report

You can then use this information to estimate the number of requests that will be duplexed in the production environment. If you have that information, you can use the following formula to obtain a very rough estimate of what the impact will be. Note that this calculation does not take account of the different performance of various link types nor the impact of synch requests being converted to asynch, and will only give a rough approximation of the expected overall average response times:

```
Multiply the number of reads by the current response time
Multiple the number of duplexed requests by the current response time, and multiply by y
Multiply the number of duplexed requests by 40 microseconds per km between the two CFs
Multiply the number of duplexed requests by 10 microseconds per km between the CPC and
the most distant CF
Add all these numbers and divide by the total number of requests
```

In the calculation above, the “y” in the second line represents the effect of duplexing a request. The value of y depends on the structure type, the contention on the CF links, the type of links used to connect the two CFs, and the speed and utilization of the two CFs. In tests we ran, which were using pre-GA levels of hardware and software, the value of y varied between 2 and 3. When estimating the effect of the distance between the CFs, we used a value of 40 microseconds per km between the CFs; if one CF is close to the connected CPC and the other is remote, it is very unlikely that there will be any overlap when issuing the CF-to-CF signals.

For the most accurate projections prior to implementing System-Managed CF Structure Duplexing, work with your IBM representative to run the Parallel Sysplex Quick Sizer. Support for System-Managed CF Structure Duplexing and the synch/asynch conversion algorithm in z/OS 1.2 has been added to Version 5.6 of this IBM Internal tool.

17.3.8 Comparison to alternatives

After reading all this you may be wondering if System-Managed CF Structure Duplexing will provide adequate performance for your applications. It is true there will be an increase in CF response times associated with the use of System-Managed CF Structure Duplexing, however, this must be viewed in light of the alternatives. You also have to put the change in

response time into perspective: if you currently spend 500 microseconds during a.5 second CICS transaction talking to the CF, doubling the CF response times because of duplexing would increase the CICS transaction response time by 0.1%,which is hardly a significant amount.

There are also situations where System-Managed CF Structure Duplexing may result in improved response times. Taking the System Logger as an example: if you have a requirement to have two non-volatile copies of the data in a log stream, prior to System-Managed CF Structure Duplexing, you had no choice but to duplex the log stream data to a staging data set on DASD. If you could use System-Managed CF Structure Duplexing to create a second, CF-based, non-volatile copy of the data, you might increase the response time of the CF component by tens or even hundreds of microseconds, however, the elimination of the I/So to the staging data sets might save you thousands of microseconds on every associated Logger request.

It is not possible to give a single answer about whether or not you should use System-Managed CF Structure Duplexing that will apply to all situations. Response time is only one aspect to be considered. And depending on the exploiter and the alternatives, the overall response time impact may or may not turn out to be an issue. The only way to know for sure is to actually test it in your own environment, with your hardware mix, distance considerations, and application profiles.

17.4 Capacity planning considerations

Because there are more components involved in processing a duplexed request, there are more opportunities for poor performance in one component to have a larger impact on CF response times, and therefore overall performance. The components involved in processing a duplexed CF request are as follows:

- Operating system CPC
- CF CPC
- CF links connecting the operating system to the CFs
- CF links connecting the CFs to each other

17.4.1 Operating system CPC

Enabling System-Managed CF Structure Duplexing for a structure will result in additional cycles being used by XES. Whether this additional processing is charged to XES or to the connector depends on whether the request is handled synchronously or asynchronously. Time spent processing synchronous requests is all charged to the connector. For asynchronous requests, some of the CPU time is charged to the XCFAS address space, and the remainder is charged to the connector.

17.4.2 CF CPC capacity

As a rule of thumb, enabling System-Managed CF Structure Duplexing for a list or lock structure will roughly double the CF CPU required to process update requests to that structure. Unfortunately, there is no easy way of identifying the amount of CF CPU associated with the processing for each structure.

When you enable System-Managed CF Structure Duplexing for a structure, the increase in CPU utilization in both CFs should help you size the capacity requirement for that structure.

A lack of CF CPC capacity may show up as one or more of the following:

- High CF CPU utilization in the RMF reports.
- Increased CF response times
- Increased response time for the CF-to-CF requests, as shown in the RMF CF-to-CF Activity report contained in Figure 17-5. An increase in these response times is a possible indicator of increased CPU contention in either the CF that issued the request, the target CF, or possibly both. The report shown in Figure 17-5 is for a CF called FACIL03, and reports the response time for requests sent to a CF called FACIL04. If these response times start to increase, it is possible that one or both CFs are suffering CPU contention.

```
-----------------------------------------------------------------------------------------------------------
                                                CF TO CF ACTIVITY
-----------------------------------------------------------------------------------------------------------
          # REQ                            ----------- REQUESTS -----------   -------------- DELAYED REQ
 PEER     TOTAL       -- CF LINKS --                #   -SERVICE TIME(MIC)-          #     % OF  ------
 CF       AVG/SEC     TYPE       USE                REQ     AVG    STD_DEV          REQ    REQ   /DEL

 FACIL04   8914       CBR         1       SYNC    8914     7.5       0.0     SYNC     0    0.0    0.0
           29.7
```

Figure 17-5 RMF CF-to-CF Activity report

In general, IBM recommends that CFs are not run at utilizations higher than 70 to 80%. Allowing for the situation where one CF might have to handle the entire workload, this translates into 35 to 40% each, assuming you have two CFs. CFs with more than one (dedicated) CP can be run at higher utilizations and still provide acceptable response times.

However, if a large percentage of your CF workload is related to duplexed structures, you should be aware that System-Managed Duplexed requests are especially sensitive to high CF CPU utilizations. This is because these requests can get undisputed several times during their execution as signals are sent back and forth between the CFs. So, as CPU utilization increases, these requests may have to wait longer each time they are ready to be dispatched again. Therefore, you should attempt to keep CF CPU utilization on the lower side of the guidance of 35 to 40% busy for CFs that process a lot of System-Managed Duplexed requests.

Another consideration is if the two CFs containing the duplexed structures are not the same technology. For example, if CF1 is a 9672-R06, and CF2 is a 2064-100, the speed with which CF2 can process the System-Managed Duplexed requests will be gated by the speed with which CF1 can process its corresponding requests. If you have this type of configuration, you can offset the impact to some extent by placing more of the CF workload in the faster CF, reducing the utilization in the slower CF and therefore improving its chances of processing the duplexed requests in a timely manner.

From a recovery perspective, things are a bit different than for an environment where System-Managed CF Structure Duplexing is not being used at all. In a traditional environment, the guidance is that the combined workload of the two CFs should not exceed approximately 80% of the capacity of the slower of the two CFs, to ensure that CF can handle the entire workload should the faster CF fail.

However, if you are using System-Managed CF Structure Duplexing, there are some additional considerations. The first is that the workload associated with the secondary structures will disappear if one of the CFs fails (because those structures will revert to simplex mode). Also, the amount of CPU required to process System-Managed Duplexed requests is higher than that required to process a simple request. So, the amount of CPU that is being

consumed processing requests for the primary structures will decrease. This is shown diagrammatically in Figure 17-6. In this case, you can see that the total CF CPU utilization of the two CFs was 109% of a single CF prior to the failure, but only 69% of a single CF after the failure.

Figure 17-6 Impact of failure on CF utilization when using System-Managed CF Structure Duplexing

So, in an environment when System-Managed CF Structure Duplexing is being used intensively, the decision about how much CF CPU capacity is required will be driven more by the need to provide acceptable performance in normal operation, and less by making sure that each CF has enough CPU capacity to handle a failure situation.

Another aspect of CF CPC capacity planning is the amount of storage in the CFs. Unlike virtual storage operating systems, which can benefit from the installation of additional processor storage, CFs are very simple—the CF needs as much processor storage as is required to hold all the structures that have been defined to reside in that CF. Providing more storage than this amount does not have any impact (positive or negative) on performance.

However, to cater for the situation where one CF fails, we recommend that each CF should be configured with sufficient storage to hold *all* the structures defined in the Parallel Sysplex. This means that in normal operation there will be an amount of unused storage in each CF—this unused storage is referred to as “white space”. Prior to duplexing, the white space in each CF should be equal to the amount of allocated storage in the other CF.

When you start using duplexing, either user-managed or System-Managed, the second copy of the duplexed structures will reside in some of this white space. This means that you should not need to purchase additional storage in order to use duplexing. However, it also means that it is no longer valid to add up all the allocated storage in a CF and expect to have that amount of white space in the other CF. This is shown diagrammatically in Figure 17-7 on page 307.

In this example, the total allocated storage in CF01 is 640 MB, and the total allocated storage in CF02 is 592 MB. If all the structures were in simplex mode, this would indicate that each CF should have 640+592 MB or 1232 MB of storage to cater for a failure of either CF. However because a number of the structures are duplexed, the storage for the secondary structure will not be required after a failure, so, in fact, each CF only really needs a total of 688 MB to be able to accommodate all allocated structures.

Figure 17-7 Use of CF storage in a duplexing environment

Now, we don't recommend configuring CFs with only the exact amount of storage required for the current structures. Over time, structures will increase in size as the workload grows. More importantly, new CFLevels often increase the storage requirements for structures, sometimes significantly, so you should always configure a generous amount of additional storage in your CFs.

Also, before migrating a CF to a new CFLevel, you should always use the CFSizer tool to check the storage requirements for your structures on the new CFLevel. The CFSizer is available on the Web at:

`http://www.ibm.com/servers/eserver/zseries/cfsizer/`

17.4.3 CF link capacity and subchannels

Earlier, we discussed the difference between synch and asynch processing, and pointed out that in both cases, the subchannel is held from the time the request is sent to the CF until XES is able to determine whether the request was successful. If a structure is in simplex mode, processing 100 requests a second with a 60 microsecond response time, the utilization of the subchannel will be roughly 6000 microseconds per second (about 0.6%). If that structure is then duplexed and the response time doubles to 120 microseconds, you would now be driving *two* subchannels (one to each CF) at 1.2% *each.*

If duplexing causes the response time of the structure to exceed the threshold at which XES decides to issue it asynchronously, the response time could increase to 300 microseconds (for example). Now, both subchannels are tied up for 300 microseconds for each request, increasing the utilization to two subchannels running at 3%, compared to the original situation where one subchannel was running at .6%. You can see from this that implementing System-Managed CF Structure Duplexing can have a sudden and significant impact on your subchannel utilization.

On traditional (non-peer) CF links with only two subchannels per CF link, this additional utilization could have an adverse impact on other CF users. If the z/OS LPAR has two non-peer links to the CF, it is possible that the four subchannels[1] will all be busy when z/OS tries to send another request to the CF.

[1] Non-peer mode CF links have two subchannels per link. Peer mode CF links have seven subchannels per link.

Over-utilized subchannels may show up as large subchannel busy counts in the RMF Subchannel Activity report. A rule of thumb is that the number of subchannel busy occurrences should be less than 10% of the total number of CF requests.

You also need to consider the CF-to-CF links. While it is possible to share a peer or CF Sender link so that the z/OS LPARs and a CF LPAR in the same CPC can all use the same physical link to communicate to a CF in another processor, you should be aware of the impact this may have. As the use of duplexing increases, the number of CF-to-CF messages increases exponentially, and may start causing contention on the CF links, resulting in more subchannel busy conditions. Also, because there are four messages sent back and forth between the CFs for each duplexed request, the performance of the CF-to-CF links is critical, even more so than the performance of the CPC-to-CF links. The increased contention resulting from sharing the links between the z/OS and CF LPARs could potentially increase that response time.

If you are going to use System-Managed CF Structure Duplexing for structures with large numbers of requests, you should strongly consider the newer peer CF links. Peer CF links may only be used between zSeries CPCs (they are not supported on 9672 CPCs), however, they provide higher speed and *seven* subchannels per link, compared to only two subchannels for non-peer links.

17.5 Planning for System-Managed CF Structure Duplexing

In this section, we describe the hardware and software requirements to provide an environment that supports System-Managed CF Structure Duplexing. We also provide a list of the required PTFs or release levels for the products that will be able to use System-Managed CF Structure Duplexing.

17.5.1 Hardware prerequisites

If you are considering implementing System-Managed CF Structure Duplexing, the first thing to check is that you have sufficient capacity to handle this new requirement. The system components that will be impacted by System-Managed CF Structure Duplexing are discussed in 17.3, “Performance considerations” on page 297. You should analyze each of these to ensure there are no bottlenecks before you start.

When looking at a Parallel Sysplex, there are two types of processors that must be considered: the processors that the operating systems will run on, and the processors that the CFs will reside in. It is also possible that these will actually be the same processors (if you are using ICFs).

To exist in a sysplex where System-Managed CF Structure Duplexing is being used, an operating system must be at the z/OS 1.2 level or higher. Therefore, all the processors within the sysplex *must* be capable of supporting those operating system levels. At the time of writing, the IBM processors that support z/OS 1.2 in a sysplex are the 9672 G5 and G6 range of processors, and the zSeries z800 and z900 ranges of processors. If you wish to use System-Managed CF Structure Duplexing in an LPAR on one of these processors, you should ensure you are using the latest driver and microcode levels to be sure of picking up the required support

While the MultiPrise 3000 range of processors support z/OS 1.2, they do not support ETR or CF links, and therefore cannot take part in a multi-system sysplex. Additionally, there are no plans at this time to make a CFLevel higher than CFLevel 9 available on those processors, so it would not be possible to use one of these processors as a test environment for System-Managed CF Structure Duplexing.

In order for a CF to support System-Managed CF Structure Duplexing, it must be running in a processor that supports CFLevel 11 or higher. At the time that System-Managed CF Structure Duplexing becomes available, this support will be available on the 9672 G5 and G6 range of processors, and the zSeries z800 and z900 processors (CFLevel 12 is the System-Managed CF Structure Duplexing-enabling CFLevel on these processors)[2].

Additionally, any CF LPAR taking part in System-Managed CF Structure Duplexing should have *dedicated* CPs. At a minimum, if they are using shared CPs, they must be running with Dynamic CF Dispatching *disabled*, and the LPAR must have a high weight compared to any LPARs it is sharing the CP with. The reason for this is because duplexed operations could time out in the time it would take for a CF to be re-dispatched on a shared CP, resulting in the System-Managed Duplexed structures dropping out of duplex mode.

A new requirement specific to System-Managed CF Structure Duplexing is the need for links *between* the CFs. These are the links required for cross-CF synchronization. If a CF is not linked to another CF, you cannot establish System-Managed CF Structure Duplexing for any structures in that CF, even if the CF is running the correct CFLevel. For availability, it is recommended to provide two links (two sender and two receiver, or two peer links) between each set of CFs. It is not envisioned that those links will be very busy (there is no data transferred over those links, only command signals), however, you should have two to cater for a link failure situation.

There are two options for providing the CF-to-CF links. You can have dedicated links between the two CFs. This will provide the highest performance. The other option is that you can share a CF Sender link between one CF LPAR and the operating system LPARs in the same CPC that is connected to the same target CF LPAR. This saves the cost of dedicated links, but will not provide the same level of performance due to contention on the links. You may decide to use this configuration until you decide to what extent you will be using System-Managed CF Structure Duplexing.

If you have more than two CFs, it is not strictly necessary to connect all the CFs to each other; however, a configuration of not-fully connected CFs could impact recovery and result in a configuration that is significantly more complex to manage and operate.

17.5.2 Software prerequisites

The software requirements for System-Managed CF Structure Duplexing consist of support in the operating system and enabling support in the connectors.

In order to define the CF-to-CF connectors, you must have HCD APAR OW45976 applied on the system that you will use to create the IOCDS for the CPC containing the CF LPARs. Without this APAR, you will not be allowed to define CF Sender links on a CF LPAR.

System-Managed CF Structure Duplexing is provided with z/OS 1.2 and later, plus an enabling APAR (OW41617). This provides the capability to format a CFRM with the new SMDUPLEX keyword, indicating that System-Managed CF Structure Duplexing can be used. It also provides the System-Managed CF Structure Duplexing code.

[2] CFLevel 11 is not available on the zSeries processors. In this chapter, any time we refer to CFLevel 11, we mean CFLevel 11 if on a 9672 G5/G6, or CFLevel 12 or later if on a z800 or z900.

Before you can use System-Managed CF Structure Duplexing, *every* system that is going to access the CFRM CDS *must* be running z/OS 1.2 or higher. If a system at a level lower than z/OS 1.2 tries to use a CFRM CDS that has been formatted with the SMDUPLEX keyword, it will be rejected. While it can stay in the basic sysplex, it will not be able to use the CFRM CDS and therefore will be unable to use any of the CFs.

In addition to support within the operating system, the connector must also support System-Managed processing. This is indicated by the connector specifying ALLOWAUTO=YES on the IXLCONN macro. Prior to z/OS 1.2, the following structures provided support for System-Managed processing:

- JES2 (since OS/390 R8)
- WLM Multisystem Enclaves (since OS/390 R9)
- WLM Intelligent Resource Director (since z/OS 1.1)
- MQSeries 5.2

In addition, the following structures have announced support for System-Managed processing (and therefore System-Managed CF Structure Duplexing) prior to, or coincident with, the availability of System-Managed CF Structure Duplexing:

- CICS Shared Temporary Storage structure[3]
- CICS CF Data Tables structure[3]
- CICS Named Counter Server structure[3]
- DB2 Version 7 Shared Communications Area (SCA)
- DB2 IRLM lock structure
- IMS IRLM lock structure
- IMS V7 Common Queue Server
- IMS V7 VSO structures
- IMS V8 Resource structure
- VTAM Generic Resources
- VTAM Multi Node Persistent Sessions
- MVS System Logger (and, by extension, all the products that use the System Logger)
- BatchPipes®
- DFSMS VSAM Record Level Sharing lock structure
- DFSMShsm Common Recall Queue
- z/OS 1.4 TCP/IP SysplexPorts structure
- z/OS 1.4 TCP/IP Sysplex-Wide Security Associations structure

The specific releases or PTFs required to enable this support are listed in Table 17-1. For the latest information on required service, refer to the PSP bucket Upgrade CFDUPLEXING, Subset

Table 17-1 System-Managed CF Structure Duplexing enabling product levels

Structure/Product	Required release or APAR
CICS CF Data Tables	CICS TS 2.2 plus z/OS APAR OW39892
CICS Named Counter Server	CICS TS 2.2 plus z/OS APAR OW39892

[3] Requires CICS TS 2.2, planned to be available in 2002

Structure/Product	Required release or APAR
CICS Shared Temporary Storage	CICS TS 2.2 plus z/OS APAR OW39892
DB2 SCA	DB2 V7.1 base
DB2 IRLM	PQ52341 and PQ48996
IMS IRLM	PQ45407 and PQ48823
IMS Shared Message Queue	IMS V7.1 plus APAR PQ47642
IMS VSO	IMS V7.1 plus APAR PQ50661
IMS Resource	IMS V8.1 base
JES2 checkpoint	z/OS 1.2 base
MQ Series	MQSeries V5.2
MVS System Logger	z/OS 1.2 base
VTAM GR	z/OS 1.2 base
VTAM MNPS	z/OS 1.2 base
WLM Multisystem Enclaves	z/OS 1.2 base
WLM IRD	z/OS 1.2 base
BatchPipes	APAR PQ49953
VSAM/RLS lock structure	z/OS 1.3 base
DFSMShsm Common Recall Queue	z/OS 1.3 base
TCP/IP EZBDVIPA	z/OS 1.4 base
TCP/IP EZBEPORT	z/OS 1.4 base

Any remaining structures that do not support System-Managed CF Structure Duplexing when it is made generally available do not require the functionality provided by it. For example, XCF has no requirement for failure isolation, and can recover regardless of whether there is a failure of a CF and one or more connected operating systems. Most of the Resource Sharing exploiters are in this category.

17.6 Setting up System-Managed CF Structure Duplexing

If you decide to use System-Managed CF Structure Duplexing, the first step is to install the prerequisite hardware and software, as described in 17.5, “Planning for System-Managed CF Structure Duplexing” on page 308.

In this section, we describe a step-by-step process to get you to the point where at least one of your structures is being duplexed by System-Managed CF Structure Duplexing.

17.6.1 Hardware changes

In 17.5, “Planning for System-Managed CF Structure Duplexing” on page 308, we describe the hardware levels required to use System-Managed CF Structure Duplexing.

The changes that must be made in relation to the hardware are as follows:

- A new IOCDS containing the definitions for the new CF-to-CF links must be activated. On a z900 at the GA2 or later level and on a z800, it is possible to add CF receiver links non-disruptively. However, on a 9672 or GA1-level z900, this change requires a Power-on Reset.
- If you decide to provide dedicated CF-to-CF links (as opposed to using MIF (formerly EMIF) to share the CF Sender links with the Operating System LPARs), you may need to add CF link adaptors to the processors containing your CFs. If you are not using peer links (only available on z900 and z800 processors), you must have *at least* two links between each pair of CFs—one sender and one receiver. Two senders and two receivers are recommended for highest availability.
- Upgrade the processor microcode to the level that provides CFLevel 11 (CFLevel 12 on a z800 or z900). This is a disruptive change in that the affected processor must have a Power-on Reset performed.
- The cables connecting the CFs to each other must be installed.
- If necessary, add capacity to support System-Managed CF Structure Duplexing. This may be required in the links to the CFs, the CF CPU, and possibly the operating system CPC.

As the upgrade of the processor Driver level and the addition of CF receiver link definitions (on a 9672) are both disruptive, it would be wise to implement both of these changes at the same time. The definitions of the CF links in HCD can be added in advance of the actual hardware being installed, so you can avoid a second outage by activating the IOCDS containing the definitions in advance. The CF link hardware may be subsequently installed non-disruptively.

The other hardware change you must implement is to link the CFs together. Prior to System-Managed CF Structure Duplexing, CFs were only linked to operating system LPs, and in fact HCD would not allow you to define a link that connected two CF LPs. In order to use System-Managed CF Structure Duplexing, however, you must now connect the CF LPs. To do this, you have to install the CF link adaptors and the connecting cables.

If you wish, you can actually physically connect the two CF LPARs prior to upgrading to CFLevel 11. This has no impact—the CF will not use the new links until CFLevel 11 is installed.

Verifying CF-to-CF connectivity

In order to be able to use System-Managed CF Structure Duplexing on a structure, there must be a functioning connection between at least two of the CFs specified in the PREFLIST for that structure. After you have physically connected the two CFs, the **D CF** MVS command should show which links are used to connect each CF to the other CF(s).

An example of the output from this command is shown in Example 17-8 on page 313. In this example, you can see that the CF called FACIL05 is connected to another CF called FACIL06 using CF links F0 and F2 (which, in this case, are peer links).

```
 D CF
IXL150I  19.34.28  DISPLAY CF 968
COUPLING FACILITY 002066.IBM.02.000000011CE3
                  PARTITION: 4  CPCID: 00
                  CONTROL UNIT ID: FFF0
NAMED FACIL05
COUPLING FACILITY SPACE UTILIZATION
 ALLOCATED SPACE                 DUMP SPACE UTILIZATION
  STRUCTURES:      38912 K        STRUCTURE DUMP TABLES:        0 K
  DUMP SPACE:       2048 K                  TABLE COUNT:        0
 FREE SPACE:      420096 K       FREE DUMP SPACE:           2048 K
TOTAL SPACE:      461056 K      TOTAL DUMP SPACE:           2048 K
                               MAX REQUESTED DUMP SPACE:        0 K
   VOLATILE:         YES         STORAGE INCREMENT SIZE:      256 K
    CFLEVEL:          12
    CFCC RELEASE 12.00, SERVICE LEVEL 04.15
    BUILT ON 09/06/2002 AT 16:03:00

....
REMOTELY CONNECTED COUPLING FACILITIES
        CFNAME          COUPLING FACILITY
        --------        -------------------------
        FACIL06         002066.IBM.02.000000011CE3
                        PARTITION: 5  CPCID: 00

                        CHPIDS ON FACIL05 CONNECTED TO REMOTE FACILITY
                        RECEIVER:   CHPID    TYPE
                                    F0       ICP
                                    F2       ICP

                        SENDER:     CHPID    TYPE
                                    F0       ICP
                                    F2       ICP
```

Figure 17-8 Enhanced D CF command

If the links do not show the status that you expect, it is possible that they have been mis-cabled. To determine whether this is the case, follow this procedure:

1. From the HMC, go into Single Object Operations for the CPC containing one of the CFs.
2. From Single Object Operations, double-click the CPC object.
3. You will be presented with an icon representing the CPC. Click the right half of that icon, using the right mouse button. You will be presented with a list of items you can work with; select **CHPIDs**.
4. You will be presented with icons representing all CHPIDs defined on the CPC. Using the left mouse button, select the CHPID that you are having a problem with. On the right portion of the screen, move to the “CHPID Operations” work area and double-click the **Channel Problem Determination** icon.
5. You may be presented with a list of LPARs that are in the access list for the CHPID you selected. In this case, select the LPAR containing the CF you are working with and click **OK**.
6. You will be presented with a window offering a choice of six possible actions; select **Analyze Channel Information** and click **OK**.
7. You will be presented with a window full of information. In the bottom right quarter of the window, you will see information about what is at the “other” end of the CHPID. The

Type/Model field contains the type and model of whatever is at the other end (hopefully some sort of CPC), and the Tag field contains the CHPID number of the slot the cable is plugged in to.

Using a combination of the enhanced D CF command and the HMC, you should be able to debug any CF Link cabling problems.

17.6.2 HCD

While System-Managed CF Structure Duplexing is only supported when operating under z/OS 1.2 or higher, it is possible to use a lower-level system to set up the HCD definitions in advance. To do this, you must install the PTF for APAR OW45976 on HCD. The CFs are connected to each other using the familiar “Connect CF Channel Paths” function in HCD.

17.6.3 Operating system

No changes are required to the operating system in order to be able to use System-Managed CF Structure Duplexing (with the exception of the enabling APAR). However, all the systems in the sysplex must be migrated to z/OS 1.2 or later before you start implementing this new capability. Once you upgrade the CFRM and LOGR couple data sets for duplexing support, those data sets are no longer usable by systems at a lower level than z/OS 1.2.

17.6.4 RMF

The z/OS 1.2 level of RMF provides specific new support for System-Managed CF Structure Duplexing. In order to gather the detailed information you need to analyze the performance of System-Managed CF Structure Duplexing, you should ensure that the CFDETAIL keyword is specified in the ERBRMFxx member used to control RMF Monitor III.

Various RMF Post Processor reports have been enhanced in support of System-Managed CF Structure Duplexing:

- The Coupling Facility Usage Report now indicates whether a structure is the Primary or Secondary instance of a duplexed structure, and reports each instance separately.
- The Coupling Facility Structure Activity report also indicates whether each structure is the Primary or Secondary instance of a duplexed structure. New fields have been added that report the Peer Wait (PR WT) and Peer Completion (PR CMP) times.

  Peer Wait is the time between when XES successfully obtains the first subchannel and when it obtains the second one. There will always be some value in this field as this is a serial process. The “% of REQ” field will always be at least 100%. If it is greater than 100, it is an indication that some requests had to be redriven because the link was busy.

  Peer Completion is the time between when the first CF responded to XES and when the second CF responds. The Peer Completion percentage for the primary and secondary structures will always add to 100%. This field can help you identify if one of the CFs consistently responds faster than the other.

  The CHANGED field in this report does not report on the number of Synch requests that have been converted to Asynch by the new Synch/Asynch algorithms. The meaning of this field is the same as it was prior to z/OS 1.2; that is, it is the number of Synch requests that were converted because no subchannel was available when XES attempted to start the request. Synch requests that have been converted to Asynch as a result of the new algorithm are reported as Asynch requests.

- There are no changes to the Subchannel Activity report specifically relating to System-Managed CF Structure Duplexing. This report continues to report on the subchannel usage by z/OS. There is no corresponding report for the new CF-to-CF links.
- Finally, there is a new CF-to-CF report, as shown in Figure 17-9. This report indicates the name of the attached CF, the number and rate of requests sent from this CF to the attached one, the number and type of CF links, and the number of requests and the service time for those requests. There is also reporting on delays, without being specific about the cause of the delay. However, the only type of delay that will show up here is a delay related to contention on the CF link used to connect the CFs.

```
-----------------------------------------------------------------------------------------------------------
                                                  CF TO CF ACTIVITY
-----------------------------------------------------------------------------------------------------------
          # REQ                          ----------- REQUESTS -----------  -------------- DELAYED REQ
 PEER     TOTAL       -- CF LINKS --                #   -SERVICE TIME(MIC)-          #     % OF  ------
 CF       AVG/SEC     TYPE       USE                REQ     AVG    STD_DEV           REQ    REQ   /DEL

 FACIL04   8914       CBR         1        SYNC    8914     7.5       0.0     SYNC     0    0.0    0.0
           29.7
```

Figure 17-9 RMF CF-to-CF report

17.6.5 CFRM changes

Whether you will use System-Managed CF Structure Duplexing for a structure or not is determined by the contents of the CFRM couple data set.

The first thing you must do is allocate new CFRM couple data sets (Primary, Alternate, and Spare), using the ITEM(SMDUPLEX) NUMBER(1) keyword on the input to the IXCL1DSU program. You then use the SETXCF COUPLE commands to migrate the new couple data sets into production. This can be done non-disruptively; however, once you are using a Primary CFRM CDS containing the SMDUPLEX keyword, it is not possible to non-disruptively transition back to a CDS without that keyword. Moving back to a CFRM CDS that was not formatted with this keyword requires a sysplex-wide IPL; however, there is no reason this should be necessary assuming the proper planning was done. Therefore you should make sure that you have an alternate and a spare CFRM CDS defined that contains this keyword.

Once the new CFRM CDS is in use, you can start defining structures to use System-Managed CF Structure Duplexing. You use the DUPLEX keyword on the structure definition to control whether you want Duplexing enabled or not.The DUPLEX keyword is used for both System-Managed Duplexing and User-Managed Duplexing. DUPLEX(ENABLED) will cause XES to automatically duplex the structure any time it is allocated, while DUPLEX(ALLOWED) will allow the structure to be duplexed, but the duplexing must be started using the **SETXCF START,REBUILD,DUPLEX,STRNM=structure_name** command.

17.6.6 LOGR CDS changes

If you wish to use System-Managed CF Structure Duplexing for any of the Logger structures, you must also create new LOGR CDSs, also containing the SMDUPLEX keyword. Once again, you should create Primary, Alternate, and Spare LOGR CDSs all with this keyword. You can use the SETXCF COUPLE command to non-disruptively transition the new data sets into production; however, you cannot migrate back to a non-SMDUPLEX LOGR CDS without a sysplex-wide IPL.

Once you have migrated the new format LOGR CDS into production, you can then change the definition of a Logger structure in the CFRM policy to specify DUPLEX(ENABLED) or DUPLEX(ALLOWED). If you specify this keyword before you migrate to the new format LOGR CDS, you will receive failure messages every 10 minutes or so as XES tries to duplex the structure, but fails because of an incorrect-format LOGR CDS.

In addition to the new format LOGR CDSs, there is also a new optional keyword (LOGGERDUPLEX) that can be used when defining the log streams in a structure that will be duplexed with System-Managed CF Structure Duplexing. Note that the determinant of whether a log stream is System-Managed Duplexed or not is the structure that it resides in—structures are duplexed, not individual log streams.

LOGGERDUPLEX(UNCOND) indicates that Logger will provide its own specific duplexing of the log data regardless of any other duplexing (such as System-Managed CF Structure Duplexing) that may be occurring.

LOGGERDUPLEX(COND) indicates that Logger will provide its own specific duplexing of the log data *unless* the log stream is in an alternative duplexing configuration that provides an equivalent or better recoverability of the log data. For example, Logger will not provide its own duplexing of the log data in the following configuration:

- When the log stream is in a non-volatile CF structure that is System-Managed Duplexed, and,
- There is no single point of failure between the two structure instances, and,
- There is a failure-independent connection between the connecting system and composite structure view.

17.6.7 CF selection changes

The rules that XES uses when deciding which CF it should use when allocating a CF structure are described in *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617. However, to summarize, the rules prior to the availability of System-Managed CF Structure Duplexing were as follows:

1. Does the CF in question have connectivity to the system trying to allocate the structure?
2. Does the CF in question have a CFLevel equal to or greater than the requested CFLevel or, if the connector has specified ALLOWAUTO=YES, does it have a CFLevel equal to or greater than CFLevel=8?
3. If the structure is being allocated for use by user-managed rebuild, is the CF failure-independent from the CF containing the old structure? The system will give preference to failure-independent CFs when allocating the new structure during User-Managed Duplexing.
4. Does the CF in question have space available that is greater than or equal to the requested structure size?
5. Does the CF in question meet the volatility requirement requested by the connector?
6. Does the CF in question contain a structure that is in this structure's exclusion list?

For a structure that is going to be duplexed using System-Managed CF Structure Duplexing there are some additional considerations:

- The secondary structure instance must be allocated in a different CF than the primary instance.
- XES prefers a CF that is failure-isolated from the CF where the primary is allocated (that is, on a different CPC). If none of the CFs in the preference list meets this requirement,

XES may use a CF that is not failure-isolated from the one containing the primary structure instance.

- The CF that is to contain the secondary structure must have CF-to-CF link connectivity with the CF where the primary structure instance is allocated.
- Both CFs must be CFLevel 11 or higher in order to do System-Managed CF Structure Duplexing.

17.7 Exploiters of System-Managed CF Structure Duplexing

In this section we discuss the benefits and considerations for each of the CF exploiters that support System-Managed CF Structure Duplexing.

17.7.1 CICS structures

Prior to CICS Transaction Server V2.2, it was not possible to rebuild any of the CICS structures (Shared Temporary Storage, CICS CF Data Tables, or CICS Named Counter Server), either for planned or unplanned changes. CICS did provide the ability to save the contents of a structure to a data set, and then reload it back into a structure later on; however, this process was disruptive—the data could not be accessed from the start of the offload until a successful completion of the reload.

Benefits

CICS TS V2.2 added ALLOWAUTO support to the three structure types. This change adds support for both System-Managed Rebuild and System-Managed CF Structure Duplexing. As a result, you can now make a planned change to a structure without impacting operations using that structure. In addition, the loss of a CF no longer means that the data in the structure is lost. This is a significant advance, meaning that it is now possible to use these structures as part of a continuously available application.

How to set it up

No changes are required to CICS in order to utilize System-Managed CF Structure Duplexing support. The ALLOWAUTO parameter is issued automatically when it allocates its structures—you do not have to do anything to tell CICS that you wish to use this support. The only changes required are to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword to the CFRM policy statements for the CICS structures. Obviously, there must be at least two CFs specified on the preference list of all the structures.

Operational considerations

If you use these structures today, you will need to update the operational procedures relating to the management of those structures. First and foremost, it is no longer the case that the loss of a CF will result in the loss of the data in these structures. If your operational procedures include processes for addressing this situation, they must be changed. Once System-Managed CF Structure Duplexing is implemented for these structures, no manual recovery is required, should there be a CF failure.

The other aspect to consider is the process of offloading and reloading data into the structures. If you do this today, it is more than likely because you want to move the related structure without losing the data in the structure. Once CICS TS V2.2 is implemented on all connected CICS regions, the structure can be moved simply by using the SETXCF REBUILD command—there is no longer a need for a disruption to CICS while this takes place.

17.7.2 DB2 SCA

The DB2 SCA structure contains information about the bootstrap data sets and any errors relating to the DB2 databases. Each DB2 instance maintains a subset of the information, and the total information is kept in the DB2 SCA structure in the CF.

If the CF fails, DB2 is able to recover by allocating a new structure and populating it with information from each of the attached DB2s. And if a DB2 fails, the information that was held in that DB2 is still available in the SCA structure in the CF. So, DB2 is able to continue processing and automatically recover from a failure that affects either the CF *or* one or more DB2 instances. Prior to System-Managed CF Structure Duplexing however, if there was a double failure that impacted both the SCA structure in the CF *and* one or more connected DB2 instances, all the DB2s in the data sharing group would abend, and a group restart was required.

Benefits

The availability of System-Managed CF Structure Duplexing means that DB2 is no longer impacted by a double failure of this type. Because there are now two copies of the SCA structure, even if there is a failure that impacts one CF and one or more connected DB2s, the remaining DB2s in the group are able to continue processing, using the data that is still available in the surviving CF.

As a result, it is now acceptable to place the SCA structure in a CF that is not failure-isolated from the connected DB2s. This can make it feasible to do DB2 data sharing using just ICFs, resulting in lower overall costs than if external CFs were required.

In addition, should there be a CF failure or a CF connectivity failure, the recovery of the SCA structure is faster for a duplexed structure. Rather than having to allocate and populate a new structure, the connections to the failed structure just need to be cleaned up and the remaining structure continues to be accessible, although in simplex mode.

How to set it up

No changes to DB2 are required when you decide to duplex the SCA structure. As stated in Table 17-1 on page 310, DB2 V7 is required in order to use System-Managed CF Structure Duplexing with the SCA structure. This is because this is the first release that supports System-Managed Rebuild for that structure, and System-Managed Rebuild support is a prerequisite for System-Managed CF Structure Duplexing. All the systems that are connected to the structure must be running DB2 V7 or later. If there are any connected DB2s that are at a lower level, it will not be possible to duplex the structure.

To duplex the SCA structure, you must update the structure definition in the CFRM policy to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword. You also have to make sure that there are at least two CFs specified on the preference list. Once you activate the new policy, the structure will be eligible for duplexing.

Operations considerations

There are no specific operational considerations if you duplex the SCA structure.

17.7.3 IRLM

IRLM is used by both DB2 and IMS to control the integrity of record-level sharing across multiple subsystems. The considerations relating to the IRLM structure are the same for DB2 and IMS.

The IRLM lock structure contains serialization information about databases that are being shared between the DB2s/IMSs in the data sharing group. Each connected IRLM instance keeps a copy of all the locks held by that DB2/IMS, and the complete picture about the locks held by all the connected DB2/IMS is kept in the IRLM lock structure in the CF.

If the CF fails, IRLM is able to recover by allocating a new structure and populating it with information from each of the attached IRLMs. And if an IRLM fails, the information that was held in that IRLM is still available in the IRLM structure in the CF. So, DB2/IMS is able to continue processing and automatically recover from a failure that affects either the CF *or* one or more DB2/IMS instances. Prior to System-Managed CF Structure Duplexing, however, if there was a double failure that impacted both the IRLM structure in the CF *and* one or more connected IRLM subsystems, all the DB2s/IMSs in the data sharing group would abend to protect the integrity of their data, and a group restart was required.

Benefits

The availability of System-Managed CF Structure Duplexing means that DB2/IMS is no longer impacted by a double failure of this type. Because there are now two copies of the lock structure, even if there is a failure that impacts one CF and one or more connected IRLMs, the remaining DB2s/IMSs in the group are able to continue processing, using the data that is still available in the surviving CF.

As a result, it is now acceptable to place the lock structure in a CF that is not failure-isolated from the connected IRLMs. This can make it feasible to do DB2/IMS data sharing using just ICFs, resulting in lower overall costs than if external CFs were required.

In addition, should there be a CF failure or a CF connectivity failure, the recovery of the lock structure is faster for a duplexed structure. Rather than having to allocate and populate a new structure, the connections to the failed structure just need to be cleaned up and the remaining structure continues to be accessible, although in simplex mode.

How to set it up

No changes to DB2/IMS or IRLM are required when you decide to duplex the lock structure. As stated in Table 17-1 on page 310, some fixes are required on IRLM to enable System-Managed Rebuild support for the lock structure and System-Managed CF Structure Duplexing. All the IRLMs that are connected to the structure must have these fixes applied.

To duplex the lock structure, you must update the structure definition in the CFRM policy to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword. You also have to make sure that there are at least two CFs specified on the preference list. Once you activate the new policy, the structure will be eligible for duplexing.

Operations considerations

There are no specific operational considerations if you duplex the IRLM structure.

17.7.4 IMS Shared Message Queue Structure

The IMS Shared Message Queue (SMQ) implementation uses from 1 to 4 CF structures to hold message queues that are shared among multiple IMS subsystems. The MSGQ structure is mandatory if Shared Queues is in use. The MSGQ Overflow structure is optional but recommended. A minimal EMHQ structure is required if Fast Path is available in the system even when EMH messages are not implemented—that is, if DEDBs are implemented. The optional EMHQ Overflow structure is recommended if EMH messages are processed. As users input IMS transactions, the resulting messages are placed in a shared message queue in the CF. Independent of which IMS placed a message on the queue, any of the Message

Processing Regions (or IFPs for EMH messages) in the IMS queue sharing group can potentially take a message off that queue for processing. When the transaction is complete, the resulting message is placed back on the queue for subsequent transmission back to the user.

In order to be able to recover the contents of the SMQ structures should there be a CF or CF connectivity failure, IMS requires the user to take periodic “structure checkpoints” to DASD of the structures. Activity to the structure is quiesced during this time, which may last for several seconds, depending on the size of the structure. In the event that an SMQ structure is lost, the most recent structure checkpoint data set (SRDS) is used to restore the structure as of the time of the structure checkpoint, then the logs of changes to the structure are applied (using information from System Logger).

The duration of the recovery depends on how much log data needs to be re-applied. Taking frequent checkpoints minimizes recovery time, but results in more frequent interruptions while the checkpoints are taken. On the other hand, minimizing the interruptions by taking less frequent checkpoints results in longer recovery times. While the recovery is proceeding, all message activity in the IMS group is quiesced.

Benefits

System-Managed CF Structure Duplexing would reduce the need (or desire) to take frequent structure checkpoints. If the cost of duplexing the structure is acceptable in your environment, System-Managed CF Structure Duplexing can allow you to reduce the frequency with which you take structure checkpoints, without exposing you to long recovery times should there be a failure.

How to set it up

No definition or parameter changes to IMS or CQS are required when you decide to duplex the SMQ structure(s). As stated in Table 17-1 on page 310, IMS V7 with some fixes is required to enable System-Managed CF Structure Duplexing support for the SMQ structures. All the IMSs that are connected to the structures must be at the required version and service level.

Because the Shared Message Queue function uses the System Logger to store log records, if you are going to use System-Managed CF Structure Duplexing for the SMQ structures, you should also use System-Managed CF Structure Duplexing for the related Logger structures.

To duplex the SMQ structures, you must update the structure definition in the CFRM policy to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword. You also have to make sure that there are at least two CFs specified on the preference list. Once you activate the new policy, the structure will be eligible for duplexing.

If an objective is to reduce the frequency of structure checkpoints, all four SMQ structures should be duplexed. In addition, you should set up procedures to reset the checkpoint frequency to the current value should any of the structures fall out of duplex mode.

Operations considerations

There are no specific operational considerations if you duplex the SMQ structures.

17.7.5 IMS Shared Fast Path Database

The IMS VSO structures are used to share IMS Virtual Storage-Only, Fast Path Databases between multiple IMS subsystems. IMS uses the “Write-Into” model. As database records get updated, the data is asynchronously written into the structure, but not to DASD. Subsequently, the updated data is hardened to DASD when IMS takes a system checkpoint. This introduces

the possibility that, should a structure or CF fail, the data on DASD cannot be assumed to be valid and a database recovery is needed. To address this issue, IMS V6 introduced IMS-managed duplexing of DEDB/VSO structures on an area-by-area basis. If one structure should become unavailable, the other IMS-managed duplexed structure is still available for reads and updates in simplex mode. To get back to duplex mode, an IMS /VUNLOAD command is issued, followed by a /STA Area command. IMS does not stop data sharing during this process. The /VUNLOAD command hardens the data to DASD and the /STA Area command puts the data back into the two structures. All data is available during this time.

Benefits

Because IMS-managed duplexing of the VSO structures is already available, is highly efficient, and provides similar availability characteristics as System-Managed CF Structure Duplexing, it is not expected that many people will convert to using System-Managed CF Structure Duplexing for these structures.

The only real benefit of using System-Managed CF Structure Duplexing over IMS-managed duplexing is that no operator intervention is required to re-duplex the structure following a failure (assuming the structure is defined with DUPLEX(ENABLED)).

How to set it up

If you do decide to implement System-Managed CF Structure Duplexing for these structures, you should disable the IMS-managed duplexing function. To do this, you must change your INIT.DBDS statements to say NOCFSTR2 rather than CFSTR2(structurename). This will stop IMS from doing IMS-managed duplexing of the associated structure.

You then need to change the definition of the VSO structures in the CFRM policy to specify DUPLEX(ENABLED)—if you specify DUPLEX(ALLOWED), you will require manual intervention to re-duplex the structure, and therefore are no better off than if you had continued using IMS-managed duplexing. Again, make sure that at least two CFs are contained in the preference list for all the System-Managed duplexed structures.

Operations considerations

If you decide to use System-Managed CF Structure Duplexing for these structures, you should update your documentation to remove the IMS /VUNLOAD and /STA AREA commands when you want to re-duplex; assuming you have specified DUPLEX(ENABLED), no operator commands are required to re-duplex when a second CF becomes available. No other changes are required.

17.7.6 JES2

OS/390 V2R8 introduced JES2 support for System-Managed Rebuild. This provided the ability to do a planned move of the JES2 checkpoint structure from one CF to another, using XCF rather than JES2 commands. However, because System-Managed Rebuild does not support rebuild from error situations, JES2 commands still had to be used to recover the structure following a CF or CF connectivity failure.

Benefits

Because JES2 already supported System-Managed Rebuild prior to z/OS 1.2, JES2 gains the ability to duplex its checkpoint structure for “free”.

This means that all management of the JES2 checkpoint structure, either for a planned or unplanned change, can now be handled using XCF commands. This brings JES2 into line with most other CF exploiters, and removes the need for a separate set of operator's procedures to recover the JES2 checkpoint following a CF-related failure. The result is simplified operations and higher availability.

The improved availability is a result of:

- Removing the stall in JES2 processing that takes place when the JES2 reconfiguration dialog is running. Because the structure is duplexed, a failure simply results in the structure reverting to simplex mode. Without duplexing, a failure would result in the structure being forwarded by JES2 to another location - either DASD or another CF.
- Simpler operations. Because the JES2 structure can now be managed in the same way as all other structures, there is less risk of an operator mistake causing a problem.

How to set it up

Whether the JES2 structure is duplexed or not is transparent to JES2. Because System-Managed CF Structure Duplexing is handled by XES, no changes are required to JES2. The definition of the JES2 structure in the CFRM policy must be updated to specify DUPLEX(ENABLED) or DUPLEX(ALLOWED), and you must ensure that there are at least two CFs specified on the Preference list.

If you presently specify another CF structure on the NEWCKPTx statement, you may decide to change that to refer to a DASD checkpoint instead. The reason is that if there is a CF failure and the structure is duplexed, it will simply revert to simplex mode using the other CF. Therefore, it is unlikely that there would ever be a situation where you would want to forward the structure to another CF structure rather than using XES to simply rebuild it. If you *do* want to forward the structure for some reason, it is more likely that you would do so to move it out of the CF completely and on to DASD.

Operations considerations

When the CF structure is duplexed, the operations procedures should be updated to remove any mention of the JES2 reconfiguration dialog in relation to recovering from a CF failure. If there is a failure of a CF containing one of the JES2 structure instances, recovery will be automatic, removing any need for operators to get involved.

There may still be times when the operators will need the JES2 reconfiguration dialog, but not in relation to a CF failure.

17.7.7 MQSeries

MQSeries for OS/390 V5.2 delivered the ability to place non-persistent messages in a structure in the CF. As long as any member of the Queue Sharing Group was available, those messages could be accessed. This provided improved availability for non-persistent messages compared to previous levels of MQSeries, where the messages would be lost if the MQSeries address space abended.

However, MQSeries for OS/390 V5.2 only provides support for System-Managed Rebuild. This means that (prior to System-Managed CF Structure Duplexing) if the CF containing the non-persistent messages were to fail, the structure could not be rebuilt and all the messages in that structure would be lost.

Benefits

System-Managed CF Structure Duplexing provides the ability to ensure that MQSeries non-persistent messages are *not* lost if there is a CF failure. If one of the CFs containing a non-persistent message structure is lost, that structure system will simply revert back to simplex mode, using the structure in the unaffected CF, and MQSeries will continue processing.

How to set it up

The support for System-Managed CF Structure Duplexing is completely transparent to MQSeries. No changes to MQSeries definitions are required, nor are there any PTFs required for MQSeries. The only change required is to add the DUPLEX(ALLOWED) or DUPLEX(ENABLED) keyword to the structure definition in the CFRM policy, as shown in Example 17-1.

Example 17-1 Sample CFRM statements for MQSeries structure

```
STRUCTURE NAME(PSMGCSQ_ADMIN)
          SIZE(20480)
          INITSIZE(10240)
          MINSIZE(6144)
          PREFLIST(FACIL03,FACIL04)
          FULLTHRESHOLD(85)
          DUPLEX(ENABLED)
```

When you do this, you must ensure that you duplex *all* the MQSeries structures. Specifically, make sure that you duplex the ADMIN structure as well as the ones that will actually contain the message queues.

Operations considerations

The ability to duplex the MQSeries structures should provide better availability for the non-persistent MQSeries messages and make operations significantly simpler, should there be a failure that impacts the shared queues.

If you lose one system, MQSeries on the remaining systems can continue processing, using the shared queue. If you lose connectivity from one system to a CF containing an MQSeries structure, the structure will automatically revert to simplex mode, and MQSeries will continue operating, masking the failure.

Similarly, if you lose one of the CFs, the MQSeries structures will revert to simplex mode and all the connected MQSeries address spaces will continue operating.

Note that in both cases, when the structure reverts to simplex mode you are once again exposed should there be a subsequent failure. Therefore, it is important to get back into duplex mode as soon as possible.

17.7.8 MVS System Logger

The System Logger is a z/OS component that provides a high performance logging capability, with the option of merging log records from multiple systems in the sysplex.

Because of the importance of log records for data integrity, System Logger has built-in functions to ensure that there is always at least one copy of the data in the log streams. If the CF and the CPC containing the Logger user are failure-isolated, you can lose either the CF or the CPC and still have a copy of the data. However, if the CF and CPC are not failure-isolated, or if the CF is not non-volatile, Logger must keep a copy of the data on DASD—if it did not do this, a single failure could potentially lose both copies of the log records.

The System Logger is used by a number of z/OS subsystems and other products:

- OPERLOG, for a single sysplex-wide syslog
- LOGREC, for a single sysplex-wide repository of LOGREC data
- CICS, for its DFHLOG and DFHSHUNT files
- IMS, for logging of messages in the Shared Message Queue
- RRS, for its multiple logs

Not all of these users require data persistence, however, for those that do, System-Managed CF Structure Duplexing can provide significant benefits.

Benefits

If the log records for a particular exploiter are currently being duplexed to DASD, System-Managed CF Structure Duplexing can provide significant performance benefits. Consider that a typical CF response time (even allowing for duplexing) is around 100 microseconds, whereas a typical DASD response time is 1500 microseconds or more. If the implementation of System-Managed CF Structure Duplexing removes the need to duplex the data to DASD, this can improve the response of the Logger requests by a factor of 10 or more.

How to set it up

When using System Logger, you will typically have many log streams in a single Logger structure. Whether a log stream is duplexed or not depends on which structure the log stream resides in. You cannot specify at an individual log stream level whether it is to be duplexed or not—if the structure the log stream resides in is duplexed, then the log stream will automatically be duplexed. If the structure is not duplexed, then the log stream will not use System-Managed CF Structure Duplexing.

If you wish to use System-Managed CF Structure Duplexing with a Logger structure, you must first reformat the Logger Couple Data Set. This is described in “LOGR CDS changes” on page 315.

In addition, there is a new keyword that can be specified at the log stream level. This keyword is LOGGERDUPLEX. Specifying a value of COND on this keyword indicates that the log stream should not be duplexed to DASD if the log stream resides in a duplexed structure *and* the two structure instances reside in CFs that are failure isolated from each other. Specifying a value of UNCOND indicates that you still want the log stream duplexed to DASD, regardless of the duplexing status of the Logger structure.

Therefore, if you wish to use System-Managed CF Structure Duplexing for a given log stream, you must ensure it is in a Logger structure that has specified DUPLEX(ENABLED) or DUPLEX(ALLOWED) in the active CFRM policy. No changes are required to the Logger exploiter (beyond any supporting service that may be indicated in Table 17-1 on page 310).

Operations considerations

There are no specific operations considerations for Logger when using System-Managed CF Structure Duplexing, beyond an understanding of how System-Managed CF Structure Duplexing works and how it automatically recovers from a CF or CF connectivity failure.

17.7.9 VTAM Generic Resources

The VTAM Generic Resources function provides the ability to assign the same alias to a number of separate instances of an application, and allow users to specify the alias name when logging on to the application. VTAM uses a structure in the CF to maintain information about each alias, including the actual APPLIDs that are associated with each alias, and information about sessions that are associated with each of those aliases. When a user issues a LOGON APPLID(name), VTAM uses the information in the CF structure to help decide which specific application instance the logon request should be routed to.

If there is a failure affecting the structure, VTAM temporarily pauses setting up new sessions until a new structure has been allocated and repopulated by all the connected VTAMs.

Benefits

Depending on the number of applications you have that are using VTAM Generic Resources and the number of sessions with those applications, it may take an amount of time to recreate the VTAM Generic Resources structure. If you duplex the structure using System-Managed CF Structure Duplexing, there is no need to go through this processing—all the required information is already available in the duplex copy. Instead, the lost structure just needs to be cleaned up, and processing can then continue, resulting in a reduced impact to VTAM session handling.

How to set it up

The support for System-Managed CF Structure Duplexing is transparent to VTAM. As long as you are using the level of VTAM delivered with z/OS 1.2, no changes to VTAM definitions are required, nor are there any PTFs required for VTAM. The only change required is to add the DUPLEX(ALLOWED) or DUPLEX(ENABLED) keyword to the structure definition in the CFRM policy.

Operations considerations

There are no specific operational considerations if you duplex the VTAM GR structure.

17.7.10 VTAM Multi Node Persistent Sessions

The VTAM persistent session function allows sessions to survive application failures, since VTAM retains certain vital session data when the application fails and restores it when the application is restarted. VTAM Multi Node Persistent Session (MNPS) support extends this capability by allowing the application to be restored on a system other than the original one. It does this by saving the session data in a structure in the CF, thereby making that data available to every system in the sysplex.

If there is a failure that affects both the MNPS structure and one or more of the connected systems, it will not be possible to recover the session information for those systems. Given that one of the reasons for using MNPS is to protect your application from a system outage, you should therefore not place the MNPS structure in a CF that is not failure-isolated from the connected systems.

Benefits

Using System-Managed CF Structure Duplexing for the VTAM MNPS structures can bring two substantial benefits:

- Faster recovery from a CF failure situation
- It is now possible to use MNPS without having a failure-isolated CF

The volume of data in an MNPS structure can be significant. If there is a failure affecting the MNPS structure, recreating the structure contents from all the connected VTAMs can take considerable time. By duplexing the structure, there is no need to go through this recreation process—all the required data is already available in the other structure instance. As a result, recovery time may be as brief as a few seconds.

The other benefit is the removal of the requirement for a failure-isolated CF. This can make MNPS a more affordable option for applications that have a persistent session requirement.

How to set it up

The support for System-Managed CF Structure Duplexing is transparent to VTAM. As long as you are using the level of VTAM delivered with z/OS 1.2, no changes to VTAM definitions are required, nor are there any PTFs required for VTAM. The only change required is to add the DUPLEX(ALLOWED) or DUPLEX(ENABLED) keyword to the structure definition in the CFRM policy.

Remember that if you have more than one MNPS structure, you should duplex all of them if you are going to duplex any.

Operations considerations

There are no specific operational considerations if you duplex the VTAM MNPS structures.

17.7.11 WLM Multisystem Enclaves

MVS/ESA SP 5.2.0 introduced the ability to create an enclave on a system and to schedule SRBs into it. OS/390 V2R9 added the ability to extend the scope of an enclave to include SRBs and TCBs running on *multiple* MVS images in a Parallel Sysplex. Some work managers split large transactions across multiple systems in a Parallel Sysplex, improving the transaction's overall response time. These work managers can use multisystem enclaves to provide consistent management and reporting for these types of transactions. WLM uses a CF structure to maintain information about each component of the multisystem enclaves.

If there is a failure affecting the WLM Multisystem Enclaves structure, WLM allocates a new empty structure. Any information about existing enclaves that was in the structure before the failure is discarded, and WLM begins building up information again (starting at the time of the new structure allocation) for any enclaves that start after that time.

At the time of writing, the only function that exploits WLM Multisystem Enclaves support is the Intelligent Data Miner product.

Benefits

If the Multisystem Enclaves structure is duplexed using System-Managed CF Structure Duplexing, information in the structure will be still be available should there be a failure affecting one of the structure instances. As a result, WLM functions, such as the ability to switch periods across enclaves on multiple systems, continue to work for existing enclaves.

How to set it up

No changes are required to WLM in order to tell it that the Multisystem Enclaves structure is to be duplexed. The only change required is to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword to the definition of the structure in the CFRM policy. Once the new policy is activated, the structure is immediately available for duplexing.

Operations considerations

There are no specific operational considerations if you duplex the WLM Multisystem Enclaves structure.

17.7.12 WLM Intelligent Resource Director LPAR Cluster structure

The Intelligent Resource Director (IRD) function in z/OS uses a CF structure to maintain detailed information about resource utilization in the systems in the LPAR Cluster. Specifically, the LPAR CPU Management and Dynamic Channel-Path Management functions in each system store information in the structure about how the related resources are being used by that image. Should another image wish to make a change (for example, to add another channel path to a control unit), information in the LPAR Cluster structure is used to evaluate the impact of that change on all the systems in the LPAR Cluster. This usage information is updated every 10 seconds, and approximately the last two minutes' worth of data are kept in the structure.

If there is a failure affecting the LPAR Cluster structure, a new, empty structure is allocated. The contents of the old structure are not recreated; however, WLM starts creating a new set of information in the new structure. It takes about two minutes for this new structure to be fully populated. In the intervening period, WLM does not make any IRD-related changes.

Benefits

If System-Managed CF Structure Duplexing is used for the LPAR Cluster structure, information in the structure is not lost. This means that there is no interruption to WLM management of IRD-related resources.

How to set it up

There are no changes that need to be made to WLM to tell it that the LPAR Cluster structure is to be duplexed. The only change required is to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword to the definition of the structure in the CFRM policy. Once the new policy is activated, the structure is immediately available for duplexing.

Operations considerations

There are no specific operational considerations if you duplex the WLM LPAR Cluster structure.

17.7.13 VSAM/RLS Lock structure

VSAM Record Level Sharing (RLS) uses a lock structure in the CF to enable CICS sharing of data in VSAM files across multiple systems in a Parallel Sysplex. The SMSVSAM address space on each system maintains a copy of the locks that are held by that system, and the lock structure contains the complete set of locks held by all systems in the sysplex.

If there is a failure affecting one or more systems in the sysplex, VSAM RLS can continue, because information about the locks held by those systems is still available in the lock structure. Similarly, if there is a failure affecting the lock structure, the contents of that structure can be recreated using information in each of the SMSVSAM address spaces. However, if there is a failure that affects both the lock structure and one or more of the connected systems, you enter what is known as a "lost locks" condition, and processing cannot continue until those locks are recovered. For this reason, it is required that the CF containing the lock structure should be failure-isolated from the systems using that structure.

Benefits

z/OS 1.3 delivers the ability to use System-Managed CF Structure Duplexing with the VSAM/RLS lock structure. This removes the requirement for the CF to be failure-isolated from the connected systems (as long as the two CFs containing the duplexed structures are failure-isolated from each other). This may permit VSAM/RLS to be exploited without necessarily having to have standalone CFs.

In addition, should there be a failure affecting the lock structure, recovery is faster for duplexed structure because no data movement is required. This can enable improved service levels across unexpected hardware events.

How to set it up

You do not have to do anything to tell SMSVSAM that its lock structure is to be duplexed. The only change required is to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword to the definition of the structure in the CFRM policy. Once the new policy is activated, the structure is immediately available for duplexing.

Operations considerations

There are no specific operations considerations for VSAM/RLS when using System-Managed CF Structure Duplexing for its lock structure, beyond an understanding of how System-Managed CF Structure Duplexing works and how it automatically recovers from a CF or CF connectivity failure.

17.7.14 DFSMShsm Common Recall Queue

z/OS V1R3 delivers the ability to store DFSMShsm RECALL requests in a structure in the CF. This has a number of benefits over the traditional mechanism whereby each HSM address space only processed its own RECALL requests:

- If one system in the sysplex has many RECALL requests queued, and the other systems are not busy, it is now possible for the RECALL requests to be processed by all systems in parallel, providing much faster recall of the data sets.
- It is possible, using an HSM exit, to prioritize RECALL requests. By placing the RECALL requests in a common queue, the high priority requests can be acted on by *all* systems in the sysplex, before any of the lower priority requests get actioned. This can provide significantly better service for these high priority requests.
- If a user on one system issues a RECALL request for a data set that is on a HSM Level 2 tape that is currently being used by another system, that other system will process the request while the tape is still mounted. Previously, the first system would need to wait for the other system to demount the tape, then issue a mount request to get the same tape mounted up again—this could result in a significant delay.
- Not all systems need to have a tape unit attached. If a system issues a RECALL request for a tape-resident data set, but does not have any tape drives attached to it, the request can be processed by one of the other systems.
- Prior to this new function, the RECALL requests were only kept in the virtual storage of one HSM address space. If that address space is stopped before all RECALL requests have been processed, the un-processed requests would be lost. By forwarding all requests to a common recall queue in the CF, the requests will still be available even if *all* the HSM address spaces in the sysplex were shut down.

Benefits

While HSM Common Recall Queue offers improved persistence for the HSM RECALL requests (assuming that the availability of the CF structure is better than the availability of a single HSM address space), it is possible to provide even better persistence by duplexing the Common Recall structure. If you do this, not only can you lose all the HSM address spaces, you could also lose one of the CFs and *still* have the RECALL requests available for when HSM is restarted.

How to set it up

You do not have to do anything to tell HSM that its Common Recall Queue structure is to be duplexed. The only change required is to add the DUPLEX(ENABLED) or DUPLEX(ALLOWED) keyword to the definition of the structure in the CFRM policy. Once the new policy is activated, the structure is immediately available for duplexing.

Operations considerations

Operators should understand that it is now possible to shut down an HSM address space without losing any un-processed RECALL requests. However, this is a characteristic of the new Common Recall Queue support, and is not specific to whether the structure has been duplexed or not. From an operations point of view, there are no specific changes should you decide to use System-Managed CF Structure Duplexing with this structure.

17.8 Operations procedures

There may be some changes to your operations procedures after you implement System-Managed CF Structure Duplexing. However, the operational considerations for System-Managed CF Structure Duplexing are exactly the same as those for User-Managed duplexing. So, if you are already using DB2 duplexed Group Buffer Pools, there should be no significant changes, except for the fact that there may be more duplexed structures now than there were previously.

17.8.1 Recovery

System-Managed CF Structure Duplexing is designed to improve the resiliency of a sysplex to errors. As such, any errors affecting duplexed structures should have less impact than was the case previously.

When recovering from a CF or CF connectivity outage, there are some extra considerations. If a CF is to contain System-Managed Duplexed structures, the CF must be directly connected to at least one other CF. Therefore, when you recover the CF, you must ensure that the CF-to-CF connectivity is restored as well. You can check the status of these links using the **D CF** command. If the CF-to-CF link has not come back online, you can use the HMC to issue commands to the CF to bring those links online; however, in most cases, those links should come online automatically when the CF is recovered.

When the CF comes back online, any structures that are defined as DUPLEX(ENABLED) will automatically re-duplex into the newly-available CF. Any structures that are defined as DUPLEX(ALLOWED) must be re-duplexed manually. Normally you would issue the following command when the CF is once again available:

```
SETXF START,RB,POPCF=newly_recovered_CF
```

This command will cause all structures that are normally resident in the named CF to move back into that CF. Once the command completes, you should issue the following command for each structure that has to be re-duplexed:

```
SETXCF START,RB,DUPLEX,STRNM=structurename
```

At the end of this process, both CFs should contain the structures that they normally contain, and any structure that was duplexed will once again be duplexed.

17.8.2 Shutting down a CF

If you wish to shut down a CF containing a duplexed structure (either User-Managed or System-Managed), there is a small change compared to stopping a CF that does not contain any duplexed structures.

Normally, when you want to stop a CF, you would issue the following command:

```
SETXCF START,RB,CFNM=cf_to_be_stopped,LOC=OTHER
```

This will cause the system to rebuild all those structures that support Rebuild from the named CF into one of the other CFs contained in their Preference Lists. However, because a duplexed structure is actually in the middle of a Rebuild, you cannot issue another Rebuild command against it. Therefore, to remove a duplexed structure from a CF, you would issue the following command:

```
SETXCF STOP,RB,DUPLEX,CFNM=cf_to_be_stopped
```

This command will cause any duplexed structures to be removed from the named CF. When the command completes, any structures that had a duplexed instance in that CF will once again be in simplex mode, with those structures located in whichever CF the other duplexed instance was in previously.

Once you have stopped all duplexed structures in the target CF, you can then proceed with the command to move all the remaining structures out of that CF, exactly as you did prior to duplexing.

17.8.3 Changes affecting duplexed structures

Some changes to a structure definition (for example, to the INITSIZE) cause a structure to go into POLICY CHANGE PENDING status when the new policy is started. For a non-duplexed structure, that status is cleared by doing a rebuild-in-place of the structure. However, you cannot issue a Rebuild command against a structure that is currently duplexed. Therefore, if you have duplexed structures with a POLICY CHANGE PENDING status, you must stop the duplexing for that structure (using the SETXCF STOP,RB,DUPLEX command). Doing this will clear the POLICY CHANGE PENDING status, and you simply re-duplex the structure again using the SETXCF START,RB,DUPLEX command. Before you can duplex a structure, you must clear the POLICY CHANGE PENDING status by rebuilding the structure.

18

z/OS V1 DFSMS Transactional VSAM Services (DFSMStvs)

In this chapter we describe the new offering of the z/OS V1 DFSMS Transactional VSAM Services (DFSMStvs), which enables the running of batch VSAM processing concurrently with online VSAM transactions. Batch and transactions can be run against the same VSAM data and concurrent updates are supported.

This chapter describes the following:

- DFSMStvs overview
- Why DFSMStvs
- The batch window problem
- Recoverable data sets
- Non-recoverable data sets
- Transactional recovery
- VSAM RLS
- Parallel Sysplex CICS VSAM RLS
- Objective of DFSMStvs
- Accessing a data set with DFSMStvs
- Using DFSMStvs
- SYS1.PARMLIB changes
- Application considerations

© Copyright IBM Corp. 2003

18.1 DFSMStvs overview

Users can improve the accessibility, availability, performance, and productivity of their VSAM storage assets with DFSMStvs offerings.

The first new offering is DFSMStvs, which enables the running of batch VSAM processing concurrently with online VSAM transactions. Batch and transactions can be run against the same VSAM data, and concurrent updates are supported.

DFSMStvs features include:

- Concurrent shared update of VSAM recoverable files across CICS transactions and batch applications
- Logging and backout functions
- Data sharing across CICS/ESA applications, batch applications, and local or distributed object-oriented (OO) applications

Important: For implementation considerations, see “DFSMStvs” on page 332.

18.2 DFSMStvs

The majority of customers using CICS have batch windows ranging from one to several hours in duration. The programs running during the batch window consist of both in-house applications and vendor-written applications. These customers would like to be able to execute their batch applications concurrently with their online applications in order to reduce or eliminate the batch window.

While VSAM record-level sharing was a starting point for this goal, it does not fully enable the necessary sharing.

18.3 Why DFSMStvs

There are a number of reasons why CICS must sometimes be taken down. A few of the major reasons are:

- To back up a recoverable VSAM data set
- To allow batch update of recoverable VSAM data sets
- To perform a reorg of a recoverable KSDS

Originally, it was necessary to quiesce CICS access to a data set in order to back it up. VSAM record level sharing (RLS) is addressed by providing “backup while open” support, which allows the data set to be backed up while it is still in use.

VSAM RLS does not allow applications update access to recoverable VSAM data sets while CICS is using them. If batch applications need to update recoverable VSAM data sets, then CICS access to them using RLS must be quiesced. This item is addressed by DFSMStvs.

Note: The online reorganization of a KSDS is not addressed by DFSMStvs.

18.4 The batch window problem

In today's environment, the batch window problem is a period of time in which CICS access to recoverable data sets is quiesced in order to allow batch jobs to run against those data sets. This process typically involves taking a backup of the data set to provide a point-in-time recovery mechanism in case the batch job fails. The batch job is then run. If it is successful, then another backup is taken, if one is needed, and CICS access to the data set is unquiesced. If the batch job is unsuccessful, the earlier backup is restored, and CICS continues using that.

18.5 Recoverable data sets

VSAM Record Level Sharing introduced a VSAM data set attribute called LOG. With this attribute a data set can be defined as recoverable or non-recoverable. A data set whose LOG parameter is undefined or NONE is considered non-recoverable. A data set whose LOG parameter is UNDO or ALL is considered recoverable.

Recoverable data sets have the following characteristics:

- An accessing application must be recognized as a commit protocol application if recovery function is required (CICS or DFSMStvs).
- Transactional backout and forward recovery capability (if needed).
- LOG(UNDO|ALL) attribute.

18.6 Non-recoverable data sets

Non-recoverable data sets have the following characteristics:

- Applicable to any application that can tolerate multiple updaters (including CICS).
- No assumption on application recovery environment.
- LOG(NONE) or undefined attribute.

The concept is to maintain a log of changed records for a recoverable data set and use the log to provide atomic commit or backout of a unit of work's (also known as “transaction” or “unit of recovery”) changes to the data set. For VSAM RLS, CICS maintains logs of its changes to recoverable data sets, and VSAM RLS inhibits batch jobs from updating recoverable data sets in RLS mode. Batch readers may concurrently read a recoverable data set, but they cannot update it. Batch updaters and CICS File Control may concurrently share non-recoverable data sets, since these data sets do not require logging.

DFSMStvs is the second stage of the DFSMS recoverable data set strategy. It provides transactional recovery within VSAM, rather than deferring this capability to callers of VSAM, by providing both logging and two-phase commit and backout protocols, in addition to the locking functions already provided by VSAM RLS.

Archived

18.7 Transactional recovery

The transaction program execution model provides data sharing of recoverable resources. During the life of a transaction, its changes to recoverable resources are *not* seen by other transactions. The transaction may request that its changes be rolled back (backed out). If the transaction fails, its changes are backed out. This capability is called *transactional recovery*. It is provided by the resource managers.

Applications that are designed to the transaction model are able to easily share the recoverable resources. The resource managers provide the sharing isolation and recovery when a transaction fails or when the execution environment fails.

IMSDB and DB2 are resource managers that provide transactional recovery for their databases. CICS File Control provides transactional recovery for VSAM recoverable files. Now, DFSMStvs provides transactional recovery for VSAM recoverable files, as well.

The following is a simple example that illustrates transactional recovery. The application wants to make a change to two different data items. A field in one data item is decremented from 200 to 100. A field in the other data item is increased from 700 to 800. Transactional recovery ensures that either both changes are made, or neither change is made. When the application requests Commit, both changes are made atomically.

If the application makes these changes to non-recoverable data and the application or the application environment fails, one or both of the changes may be lost.

The use of recoverable data ensures all data and other recoverable resources that are within the same commit scope are always commit-consistent. A *commit scope* is the set of recoverable resource managers that participate in a commit.

18.8 VSAM RLS

VSAM RLS is an alternative to traditional VSAM record management. It enhances cross-system data sharing by enabling VSAM data sets to be shared at the record level, rather than at the CI or CA level. It does this using a storage hierarchy made up of the storage on the system, a Coupling Facility (CF) cache, and DASD and its cache.

VSAM RLS uses the same interfaces as traditional VSAM and the same formats of data sets. KSDSs, RRDSs, VRRDSs, and ESDSs are supported; linear data sets are not. Note that for a data set to be eligible for RLS access, it must be SMS-managed. This is because there is information in the storage class that indicates how the data set is to be managed within the storage hierarchy, and it is the association of a storage class with a data set that makes the data set SMS-managed.

RLS introduced the concept of Recoverable File to VSAM. This attribute is specified via Access Methods Services (AMS) DEFINE or ALTER. The parameter and options are:

LOG(NONE) This declares the file non-recoverable.

LOG(UNDO) This declares the file recoverable.

LOG(ALL) This specifies that the file is recoverable and also requests forward recovery logging (redo) of changes.

The attribute only applies when the file is accessed in RLS or DFSMStvs mode. When the file is accessed in NSR/LSR mode, VSAM ignores the attribute.

The recoverable attribute means that when the file is accessed in RLS or DFSMStvs, transactional recovery is provided. With RLS, the recovery is only provided when the access is through CICS File Control, so RLS does *not* permit a batch (non-CICS) job to OPEN a recoverable file for OUTPUT.

RLS does provide read access to a recoverable file by batch jobs. The batch job may request RLS read locking to avoid seeing uncommitted changes made by sharing CICS applications. The batch job may declare, via a JCL parameter, that it wants to access the file through RLS.

The parameter and its options are:

RLS=NRI This specifies RLS access without read locking.

RLS=CR This specifies RLS access with read locking.

RLS also supports non-recoverable files. It provides record locking and file integrity across concurrently executing CICS and batch applications; however, transactional recovery is *not* provided because VSAM RLS does not do the following:

- Undo logging
- Two-phase commit/backout support

Most batch jobs that modify VSAM files are *not* designed to share data and *cannot* use this form of data sharing.

18.9 Parallel Sysplex CICS VSAM RLS

Figure 18-1 illustrates how CICS Application Owning Regions (AORs) directly access VSAM RLS.

Figure 18-1 Parallel Sysplex CICS VSAM RLS

CICS AORs do not have to function-ship VSAM requests to a CICS File Owning Region (FOR) as they used to do prior to RLS. Doing this enhances availability because it no longer relies on a CICS FOR, thus avoiding a single point of failure (if the FOR becomes unavailable, all access to the file is lost).

The Coupling Facility is used to provide the base multisystem functions required for sharing data between members in a sysplex.

Figure 18-2 on page 336 shows that VSAM RLS is structured as a server running in its own address spaces with associated data spaces. Users request VSAM RLS services from their own address spaces. These requests, made using the standard VSAM interfaces, are passed to the SMSVSAM server. The server keeps lock information within the address space and out in the Coupling Facility-based lock structure. It also caches user data in cache structures within the Coupling Facility, as well as keeping copies of buffers in the SMSVSAM data space.

The Coupling Facility cache structure provides a mechanism to maintain buffer pool coherence between the buffer pools created in the data spaces. This coherence is maintained using the cross-invalidate facility provided by the Coupling Facility.

Figure 18-2 VSAM record-level sharing

VSAM uses the store-through cache (CF cache structure) technique to write changed records. In this way, changed records are written both to DASD and the CF cache structures.

VSAM RLS is an access mode specified in the ACB.

18.10 Objective of DFSMStvs

The objective of DFSMStvs is to provide transactional recovery directly within VSAM. It is an extension to VSAM RLS. It allows any job or application that is designed to permit read/write sharing of VSAM recoverable files.

DFSMStvs is a follow-on capability based on VSAM RLS, which provides a sysplex-wide server for sharing VSAM files. It provides Coupling Facility (CF)-based locking and data caching with local buffer cross-invalidate. RLS supports CICS as a transaction manager. This provides sysplex data sharing of VSAM recoverable files when accessed through CICS. CICS provides the necessary unit-of-work management, undo/redo logging, and commit/backout functions. VSAM provides the underlying sysplex-scope locking and data access integrity.

DFSMStvs adds logging and commit/backout support to VSAM RLS. DFSMStvs requires/supports the z/OS Recoverable Resource Management Services (RRMS) component as the commit or sync point manager.

DFSMStvs provides a level of data sharing with built-in transactional recovery for VSAM recoverable files that is comparable to the data sharing and transactional recovery support for databases provided by DB2 and IMSDB.

The transaction program execution model provides data sharing of recoverable resources. During the life of a transaction, its changes to recoverable resources are *not* seen by other transactions. The transaction may request that its changes be rolled back (backed out). If the transaction fails, its changes are backed out. This capability is called *transactional recovery*. It is provided by the resource managers.

IMSDB and DB2 are resource managers that provide transactional recovery for their databases. CICS File Control provides transactional recovery for VSAM recoverable files. Now, DFSMStvs provides transactional recovery for VSAM recoverable files as well (see Figure 18-3).

Figure 18-3 DFSMStvs

Applications that are designed to the transaction model are able to easily share the recoverable resources. The resource managers provide the sharing isolation and recovery when a transaction fails or when the execution environment fails.

IMSDB and DB2 are resource managers that provide transactional recovery for their databases. CICS File Control provides transactional recovery for VSAM recoverable files. Now, DFSMStvs provides transactional recovery for VSAM recoverable files, as well.

With VSAM RLS, batch jobs could share non-recoverable files for read and update while CICS is using them. Assuming the share options were correctly defined, they could also share recoverable files, as long as they only want to read them.

With DFSMStvs added to the picture and built on top of VSAM RLS, full sharing of recoverable files becomes possible. Batch jobs can now update the recoverable files without first quiescing CICS's access to them.

18.11 Accessing a data set with DFSMStvs

For the most part, DFSMStvs only supports those data sets that are defined as recoverable. That is, the log attribute for the data set is either UNDO (backout logging only) or ALL (backout and forward recovery logging). When a batch job opens a recoverable data set for update, the open is done in DFSMStvs mode. This allows DFSMStvs to provide the necessary transactional recovery for the data set.

Note that data sets opened for input with the CRE option specified are also open in DFSMStvs mode. This is because the CRE (consistent read explicit, also known as repeatable read) locks are sync point duration locks. Without DFSMStvs support, VSAM RLS would know nothing about sync points, and the locks would never get released.

In either case - read or update access - the application is responsible for defining the sync points by invoking the RRS commit or backout function. It is not possible for Transactional VSAM to define the sync points because it knows nothing about what the application is actually doing. If it tried to imply a sync point periodically, any of the following could happen:

- DFSMStvs could insert the sync point between paired operations (that is, a GET UPD and its paired PUT UPD or ERASE).
- DFSMStvs could insert the sync point in the middle of two pieces of work that were meant to be atomic (for example, between subtracting 100 dollars from a checking account and adding it to a savings account).
- DFSMStvs could decide to commit a piece of work that the application would have realized should have been backed out (or vice versa).

18.12 Using DFSMStvs

DFSMStvs permits a batch job to OPEN a recoverable file for OUTPUT. However, most existing batch jobs that modify VSAM files are not designed to permit data sharing. Each job assumes the file is not being changed by another, concurrently executing program.

DFSMStvs provides the necessary transactional recovery to enable data sharing. Batch jobs that are designed to use the transactional programming model may use DFSMStvs to read/write shared VSAM recoverable files.

Figure 18-4 Using DFSMStvs

Table 18-1 shows an example of the usage of interfaces to DFSMStvs. The table shows the application interfaces available; notice that the only change to the already existing VSAM application interface is the repeatable read option on GET.

Table 18-1 VSAM application interface

Application	RLS and DFSMStvs
GET UPD	Obtain Lock, Log Undo
PUT UPD	Log Redo
GET repeatable read	Obtain Lock
PUT Add	Obtain Lock, Log Undo/Redo
GET UPD	Obtain Lock, Log Undo
PUT UPD	Log Redo
Call SRRCMIT	Commit Changes, Release Locks

Call SRRCMIT invokes the RRS component of RRMS to commit the changes made by the application. RRS interfaces with DFSMStvs to commit the VSAM file changes and release the corresponding VSAM locks.

18.12.1 DFSMStvs logging

DFSMStvs logging uses the z/OS System Logger. The DFSMStvs logger is a reuse of the design and much of the code of the CICS logger. Forward recovery logstreams for VSAM recoverable files are shared across CICS and DFSMStvs. The forward recovery logstream is specified as an attribute of the file. It is specified via AMS.

Figure 18-5 DFSMStvs logging

DFSMStvs logstreams

Each DFSMStvs instance has a primary system log (undo log) and a secondary system log (shunt log). Both are implemented as MVS system logger logstreams. These logs are intended for use only for recovery purposes (for example, during backout or restart). They are not meant to be used for any other purpose. The primary system logstream holds data for most normal in-flight units of recovery (URs). The secondary system logstream holds information for URs that cannot be completed, normally due to backout failures, or that have been determined to be long-running.

System logstream names are generally qualified names, of which the high level qualifier is the DFSMStvs instance name. DFSMStvs requires the logstream names to be IGWTVnnn.IGWLOG.SYSLOG and IGWTVnnn.IGWSHUNT.SHUNLOG, respectively.

DFSMStvs instance names must be unique throughout the sysplex. Each DFSMStvs instance supports the system log for its exclusive use. All other logs are kept separate from the system log. Their stream names are checked to ensure that they are different from that of the system log.

You must define the logstreams for the DFSMStvs primary and secondary system logs to the MVS system logger before starting DFSMStvs.

Forward recovery logs are kept separate from system logs. DFSMStvs obtains the logstream name of a VSAM forward recovery log from the ICF catalog entry for the data set. Note that DFSMStvs only *writes* to forward recovery logs; it is the responsibility of products that provide forward recovery capability to read them.

If the use of a log-of-logs is requested via the IGDSMSxx member of SYS1.PARMLIB, DFSMStvs writes one. It contains copies of the start of run records, the tie-up records and file close records for recoverable data sets, and logstream exception information. This provides data set recovery products such as CICSVR with the information required to control forward recovery. If you use both DFSMStvs and CICS, it is recommended that you use the same log-of-logs for both.

If you do not want DFSMStvs to write a log-of-logs, omit the LOG_OF_LOGS parameter from the IGDSMSxx member of SYS1.PARMLIB. If you use DFSMStvs in a sysplex environment, and run with a log-of-logs, then the log-of-logs should be a single logstream shared by all DFSMStvs instances that are used to access the same set of recoverable data sets.

Define log structures

You must define the Coupling Facility structures needed for all the logstreams that are used by your DFSMStvs instances. You define structures in the MVS system logger LOGR policy in the system logger couple data sets using the DEFINE STRUCTURE specification of the ICXMIAPU utility.

Define logstreams

The logstreams are defined using the DEFINE LOGSTREAM specification of the ICXMIAPU utility.

We strongly recommend that you specify DIAG(YES) for the DFSMStvs system logs. This parameter indicates that special Logger diagnostic activity is allowed for this logstream. This is especially useful when certain Logger errors occur, such as X'804', that indicate that some data may have been lost.

RACF definitions for DFSMStvs logstreams

Installations must define authorization to system logger resources in order for DFSMStvsto be able to access, read, and write its logstreams (undo, shunt, and forward recovery logstreams). This applies to both Coupling Facility and DASD-only logstreams. It is recommended that the RLS server address space (SMSVSAM) be assigned privileged and/or trusted RACF status.

Log deletion

DFSMStvs does not provide any mechanism for the automatic deletion of records from forward recovery logstreams. It is the installation responsibility to manage the size of these logstreams. If the installation wants to keep them from becoming excessively large and needs long-term data retention, then it may want to copy the data from logstream storage into alternative archive storage.

18.12.2 Context and Unit of Recovery (UR)

In order to have transactional recovery, we must have a means of uniquely identifying transactions (units of recovery). For DFSMStvs, z/OS provides the interfaces through which the UR identity is communicated.

z/OS provides a Context and UR structure associated with a TCB. The structure includes a UR identifier, and it tracks the resource managers that can participate in the commit or rollback of the UR.

The UR consists of the set of changes that are to be made or not made as an atomic unit. Therefore, a UR represents an application program's changes to resources since the last commit or backout, or, for the first UR, since the beginning of the application. Each UR is associated with a context. The life of a context is typically a series of application program URs.

Context management

z/OS provides the context and UR management under which DFSMStvs participates as a recoverable resource manager. When a TCB is dispatched, it has a context. For the standard case where the work is not managed by a transaction monitor or work manager, z/OS provides a default context known as the *native context*. A native context is the automatically occurring context of the application program and protected resources associated with a work request. A native context is always associated with a single application task and always exists.

A transaction/work manager may use z/OS Context Services to create privately-managed contexts. The resource/work manager owns any privately-managed contexts it creates, and can switch them from one task to another. A privately-managed context is usually used be a work manager that is a resource manager (such as IMS) that accepts and manages work, such as transactions, from outside the system.

Every task in the system has an associated context, so there is always a context for a given task. When a task is created, context services provides the original (native) context for the task. Resource managers can create privately-managed contexts and associate them with a specific task. The privately-managed context then becomes the current context. The native context still exists, but it is not current. If the resource manager later disassociates the privately-managed context from the task, the native context would again become current.

DFSMStvs does not create or switch context. It does, however, allow others to do so. When DFSMStvs receives control, it works under whatever context is current. Since a UR always remains paired with its context, it is important that the correct context be in control when any DFSMStvs work is done.

18.13 SYS1.PARMLIB changes

Figure 18-6 shows the IGDSMSxx PARMLIB member. It contains the parameters used by DFSMStvs, which are highlighted.

```
SMS  ACDS(acds)                                    COMMDS(commds)
     INTERVAL(nnn|15)                              DINTERVAL(nnn|150)
     REVERIFY(YES|NO)                              ACSDEFAULTS(YES|NO)
     SYSTEMS(8|32)                                 TRACE(OFF|ON)
     SIZE(nnnnnK|M)                                TYPE(ALL|ERROR)
     JOBNAME(jobname|*)                            ASID(asid|*)
     SELECT(event,event....)                       DESELECT(event,event....)
     DSNTYPE(LIBRARY|PDS)
     RLSINIT(NO|YES)                               RLS_MAX_POOL_SIZE(nnn|100)
     SMF_TIME(NO|YES)                              CF_TIME(nnn|3600)
     BMFTIME(nnn|3600)                             CACHETIME(nnn|3600)
     DEADLOCK_DETECTION(iii|15,kkk|4)              RLSTMOUT(nnn|0)
     SYSNAME(sys1,sys2....)                        TVSNAME(nnn1,nnn2....)
     TV_START_TYPE(WARM|COLD,WARM|COLD...)         AKP(nnn|1000,nnn|1000)
     LOG_OF_LOGS(logstream)                        QTIMEOUT(nnn|300)
     MAXLOCKS(max|0,incr|0)
```

Figure 18-6 IGDSMSxx PARMLIB member

18.13.1 DFSMStvs-related parameters

Some DFSMStvs parameters apply only to the system on which they are found. Others apply across the sysplex. Regardless of which type a parameter may be, values are not remembered across IPLs. Therefore, your IGDSMSxx member of SYS1.PARMLIB must always specify a complete set of the parameters that you wish DFSMStvs to use.

RLSTMOUT It specifies the maximum time in seconds that a VSAM RLS or DFSMStvs request is to wait for a required lock before the request is assumed to be in deadlock and aborted with VSAM return code 8 and reason code 22(X'16'). RLSTMOUT is specified as a value in seconds in the range of 0 to 9999. The default is 0. A value of 0 means that the VSAM RLS or DFSMStvs request has no timeout value; the request waits for as long as necessary to obtain the required lock.

SYSNAME It specifies the names of the systems to which the preceding or subsequent DFSMStvs instance names apply. Up to 32 system names may be specified. The system names must be specified in the same order as the DFSMStvs instance names. SMS examines the system names specified and compares them to the system name in the CVT. When a match is found, it stores the value of the TVSNAME parameter in the matching position as the DFSMStvs instance name for the system. The combination of SYSNAME and TVSNAME should be used when the PARMLIB member is shared between systems.

TVSNAME It specifies the identifiers that uniquely identify instances of DFSMStvs running in the sysplex. Up to 32 identifiers may be specified. The identifiers must be unique within the sysplex. They must be a numeric value from 0 to 255, which DFSMStvs uses as the last byte of its instance name (although it is displayed as three bytes).

TV_START_TYPE It specifies the type of start DFSMStvs is to perform. Up to 32 TV_START_TYPE values may be specified. TV_START_TYPE values must be specified in the same order as DFSMStvs instance names. If WARM is specified, then DFSMStvs reads its undo log and processes the information it finds in accordance with the information RRS has about any outstanding URs. If COLD is specified, then DFSMStvs deletes any information remaining in the undo log and starts as if the log were empty. COLD should be used when the DFSMStvs undo log has been damaged. The default is WARM.

AKP It specifies the activity key point trigger value, which is the number of logging operations between the taking of key points. Up to 32 activity key point (AKP) values may be specified. AKP values must be specified in the same order as DFSMStvs instance names. Valid values are 200 to 65535. The default is 1000.

LOG_OF_LOGS It specifies the logstream to be used as the log of logs. This log contains copies of the tie-up records written to forward recovery logs and is used by forward recovery products. If it is not specified, then no log of logs is used. The default is to use no log of logs.

QTIMEOUT It specifies the quiesce exit timeout value in seconds. Only one quiesce timeout value may be specified and it applies to all systems in the sysplex. The first instance of DFSMStvs brought up within the sysplex determines the value. Subsequent DFSMStvs instances use the value established by the first system, regardless of what may be specified in their members of SYS1.PARMLIB. The quiesce timeout value specifies the amount of time the DFSMStvs quiesce exits allow

to elapse before concluding that a quiesce cannot be completed successfully. Valid values are 60 to 3600. The default is 300.

MAXLOCKS It specifies two values: The maximum number of unique lock requests that a single unit of recovery may make before warning messages are issued, and an increment value. Once the maximum number of unique lock requests is reached, the warning messages are issued every time the number of unique lock requests exceeding the maximum increases by a multiple of the increment.

SYS1.PARMLIB parameter rules

The following rules apply to the DFSMStvs parameters:

- If any DFSMStvs parameter is specified but TVSNAME is not, it is treated as a syntax error.
- TVSNAME may be specified without SYSNAME only when there is only a single TVSNAME specified, for example TVSNAME(1).
- If more than one TVSNAME value is specified and SYSNAME is not, it is treated as a syntax error.
- All DFSMStvs parameters that take multiple values must either accept defaults for all values, or specify the same number of values. That is, AKP specifying two values and TV_START_TYPE specifying three values is invalid.
- If DFSMStvs parameters are found in the IGDSMSxx member of SYS1.PARMLIB by a system that is not listed in SYSNAME, then DFSMStvs is not started on that system (the DFSMStvs parameter is ignored).
- Syntax errors such as TVSNAME(1,2,3) SYSNAME.(SYS1,SYS2,SYS3) TV_START_TYPE(COLD,WARME,COLD) are flagged only on the system to which they apply (SYS2, in this case). Note, however, that errors that the parser cannot handle, such as (COLD,WARMEST,COLD), may be treated as an error on all systems. In this case, the parser disallows the specification because the value specified exceeds the maximum length the parser was told to accept.
- Syntax errors where there is a mismatch in the number of values specified, such as TVSNAME(1,2) SYSNAME.(SYS1,SYS2,SYS3), are flagged on all systems, because DFSMStvs is unable to determine to which systems the TVSNAMEs were intended to apply.
- If there was a TVSNAME specified on the system, but an IGDSMSxx member of SYS1.PARMLIB is subsequently activated that does not specify TVSNAME, the TVSNAME is cleared. If DFSMStvs comes down (or the SMSVSAM server recycles), DFSMStvs is not restarted.

18.14 Application considerations

DFSMStvs provides isolation of each UR. The application may use repeatable read to inhibit changes to records that it reads. Changed records of recoverable files are locked with exclusive locks. All of the UR's locks are released at commit or backout. The UR's changes are then visible to other URs.

Notice that DFSMStvs is providing transactional recovery, and hence is supporting a transactional environment. In order for a batch job to participate in this data sharing, it must be designed as a series of transactions. At end-of-transaction, the job's DFSMStvs record locks are released. Issuing frequent sync points (commits or backouts) avoids timeout of other jobs/ transactions that may be waiting for the locks. It also avoids writing large amounts of data to the log that then has to be offloaded from the Coupling Facility. Overuse of the log results in a more frequent need to offload and can create a performance bottleneck.

The application must rely on RLS and DFSMStvs locking of file records. The locks ensure that the application sees consistent information. When reading, three levels of integrity are available:

- NRI - no read integrity, or dirty read
- CR - consistent read, which ensures that the application does not read a record that is in the process of being updated
- CRE - consistent read explicit, which ensures that the record remains unchanged for the duration of the UR.

Although the application may use repeatable read to inhibit changes to records that it reads, it is recommended that this not be used more than absolutely necessary, since it locks out updaters.

Notice that an application that uses CRE reads *must* issue commits or backouts in order to release the locks, even if it has made no updates. If it does not, it continues to hold the locks, potentially locking out other applications.

Applications need to be designed to avoid deadlocks. A deadlock occurs if each of two URs needs exclusive control of some resource (a record) that the other already holds (for example, UR A holds record 1 and wants record 2, while UR B holds record 2 and wants record 1).

In order to avoid this type of deadlock, it is important that applications that are intended to run in a shared environment access resources in a predictable order. If, for example, URs A and B in our example always access resources in ascending order, the deadlock would not occur.

Here are some rules for avoiding deadlocks:

- All applications that update multiple resources should do so in the same order.
- If a data set has an alternate index, beware of mixing URs that perform updates via the base key with URs that perform updates via an alternate key. URs that perform updates via the base key can deadlock with URs that perform updates via an alternate key because the key that is locked is always the base key.
- An application that issues a GET UPD should follow it with a PUT UPD or ERASE and complete work on behalf of the UR as quickly as possible. It should then invoke RRS to commit the UR, allowing the locks held on behalf of the UR to be released.
- Commits should be issued by applications that use CRE to browse data sets. Although the locks for GET CRE requests are shared locks, they are held until sync point processing. This also applies to POINT CRE. In addition, POINT CR causes a shared lock to be obtained that is not released until positioning on the RPL is changed or the UR reaches a sync point.

Archived

A

msys for Operations implementation checklist

This appendix contains a step-by-step implementation checklist as outlined in Chapter 4, “msys for Operations enhancements” on page 43:

- “Step 1: Create VSAM and non-VSAM data sets” on page 348
- “Step 2: Copy additional PROCs into PROCLIB data set” on page 354
- “Step 3: Data sets for LNKLST and LPALST” on page 356
- “Step 4: Add a PPT entry” on page 358
- “Step 5: Update MPFLST” on page 359
- “Step 6: Define application major nodes for VTAM” on page 361
- “Step 7: Make determined security definition changes” on page 362
- “Step 8: Alter msys for Operations NVSS style sheet” on page 371
- “Step 9: Enable msys for Operations functions” on page 373
- “Step 10: Build the VTAM logon mode table AMODETAB” on page 383
- “Step 11: REXX environment table entries” on page 384
- “Step 12: Perform hardware customization on SEs” on page 387

© Copyright IBM Corp. 2003

Step 1: Create VSAM and non-VSAM data sets

Start by creating the shared and system-unique VSAM and non-VSAM data sets. The shared DSIPARM data set will contain customized, installation-unique msys for Operations settings. Sharing this data set will minimize setup activities. The sample job to perform this activity is member INGALLC0 in ING.SINGSAMP. Modify this job so that a set of unique data sets are created for each participating system, as shown in Example A-1.

Note: The second job step copies a seed record into the IPLDATA cluster after it is created. This record is in stream under ddname LOWKEY, and the first 20 bytes must be hexadecimal zeros and not spaces or blanks (X'40's) as they would appear:

```
                                10Y
00000000000000000000FFE4444444444444444444444444444444444444444444444
00000000000000000000108000000000000000000000000000000000000000000000
```

Example: A-1 Modified sample job to create msys for Operations data sets

```
//RONNMSOA JOB (034D000,TS),NORTHRUP,CLASS=C,MSGCLASS=T,
//             NOTIFY=RONN
//MSOALOC PROC Q1=MSOPS,              ** DSN HIGH LEVEL QUALIFIER
//             DOMAIN=,               ** MSYS/OPS DOMAIN NAME
//             SER=TOTSTJ,            ** WHERE TO ALLOCATE DATA SET
//             DSN=,                  ** Low LVL DSN QUALIFIER
//             PS=,                   ** PRIMARY PDS SPACE
//             RECFM=FB,              ** RECORDFORMAT
//             LRECL=80,              ** RECORDLENGTH
//             BLKSIZE=6160,          ** BLOCKSIZE
//             DIR=                   ** DIRECTORY BLOCKS FOR PDS SPACE
//STEP1   EXEC PGM=IEFBR14
//DSNMODEL DD  DSN=&Q1..&DOMAIN..&DSN,
//             DISP=(NEW,CATLG,CATLG),
//             SPACE=(CYL,(&PS,1,&DIR)),
//             DCB=(RECFM=&RECFM,LRECL=&LRECL,BLKSIZE=&BLKSIZE),
//             UNIT=SYSDA,VOL=SER=&SER
//        PEND
//DSIPARM EXEC MSOALOC,DSN=DSIPARM,PS=5,DIR=30,DOMAIN=SHARED,BLKSIZE=3920
//DSILST1 EXEC MSOALOC,DOMAIN=MSO47,DSN=DSILIST,PS=1,DIR=10
//DSILST2 EXEC MSOALOC,DOMAIN=MSO54,DSN=DSILIST,PS=1,DIR=10
//DSILST3 EXEC MSOALOC,DOMAIN=MSO55,DSN=DSILIST,PS=1,DIR=10
//*
//STEP2   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DATA,DLM='||'
 /* ************************************************************** */
 /* Define Shared VSAM Cluster For Saving IPL Related Information   */
 /* ************************************************************** */
 SET MAXCC=0
 DEF CLUSTER(NAME(MSOPS.SHARED.IPLDATA)                    -
        INDEXED                                            -
        NOREUSE                                            -
        IMBED                                              -
        KEYS(20 0)                                         -
        RECSZ(90 4089)                                     -
        CISZ(4096)                                         -
        SHR(4 3)                                           -
        REC(10000 1000))                                   -
        VOL(TOTSTJ)                                        -
      DATA(NAME(MSOPS.SHARED.IPLDATA.DATA))                -
```

```
        INDEX(NAME(MSOPS.SHARED.IPLDATA.INDEX))
 IF LASTCC = 0 THEN                                        -
   DO
   REPRO IFILE(LOWKEY)                                     -
           ODS(MSOPS.SHARED.IPLDATA)
   END
||
//LOWKEY   DD  DATA,DLM='||'
                    10Y
||
//*                  +-- Y/N SAVE PARMLIB DATA W/ OR W/O COMMENTS
//*                ++--- NN  NUMBER OF IPL RECORDS PER SYSTEM
//*
//STEP3   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DATA,DLM='||'
 /* ************************************************************** */
 /* Define System SC47 Unique VSAM Clusters - msys/Ops Domain MSO47 */
 /* ************************************************************** */
 DEF CLUSTER(NAME(MSOPS.MSO47.DSILOGP)     -
          INDEXED                          -
          KEYS (4,8)                       -
          RECSZ(125,404)                   -
          FSPC(0,0)                        -
          REUSE                            -
          SHR(2)                           -
          CYLINDERS(1)                     -
          VOL(TOTSTJ))                     -
        DATA                               -
         (CISZ(4096))                      -
        INDEX                              -
         (CISZ(1024)                       -
          IMBED)
 DEF CLUSTER(NAME(MSOPS.MSO47.DSILOGS)     -
          INDEXED                          -
          KEYS (4,8)                       -
          RECSZ(125,404)                   -
          FSPC(0,0)                        -
          REUSE                            -
          SHR(2)                           -
          CYLINDERS(1)                     -
          VOL(TOTSTJ))                     -
        DATA                               -
         (CISZ(4096))                      -
        INDEX                              -
         (CISZ(1024)                       -
          IMBED)
 DEF CLUSTER(NAME(MSOPS.MSO47.DSITRCP)     -
          INDEXED                          -
          KEYS (4,8)                       -
          RECSZ(114,146)                   -
          FSPC(0,0)                        -
          REUSE                            -
          SHR(2)                           -
          CYLINDERS(1)                     -
          VOL(TOTSTJ))                     -
        DATA                               -
         (CISZ(16384))                     -
        INDEX                              -
         (CISZ(512)                        -
```

```
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.DSITRCS)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(114,146)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(16384))                     -
       INDEX                              -
        (CISZ(512)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.DSISVRT)     -
         INDEXED                          -
         SHR(2)                           -
         VOL(TOTSTJ)                      -
         CYLINDERS(2 1)                   -
         KEYS(54 0)                       -
         RECSZ(64 0512)                   -
         FSPC(5 5)                        -
         REUSE)                           -
        DATA                              -
        (CISZ(8192))                      -
       INDEX                              -
        (CISZ(4096)                       -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO47.STATS)       -
         VOL(TOTSTJ)                      -
         KEYS(20 0)                       -
         RECSZ(252 252)                   -
         FSPC(0 0)                        -
         SHR(2)                           -
         CISZ(512)                        -
         INDEXED                          -
         REUSE                            -
         IMBED)                           -
       DATA                               -
        (NAME(MSOPS.MSO47.STATS.DATA)     -
         CYL(2 0))                        -
       INDEX                              -
        (NAME(MSOPS.MSO47.STATS.INDEX)    -
         TRK(2 0))
/* ************************************************************** */
/* Define System SC54 Unique VSAM Clusters - msys/Ops Domain MSO54 */
/* ************************************************************** */
DEF CLUSTER(NAME(MSOPS.MSO54.DSILOGP)     -
         INDEXED                          -
         KEYS (4,8)                       -
         RECSZ(125,404)                   -
         FSPC(0,0)                        -
         REUSE                            -
         SHR(2)                           -
         CYLINDERS(1)                     -
         VOL(TOTSTJ))                     -
       DATA                               -
        (CISZ(4096))                      -
       INDEX                              -
```

```
        (CISZ(1024)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSILOGS)      -
         INDEXED                           -
         KEYS (4,8)                        -
         RECSZ(125,404)                    -
         FSPC(0,0)                         -
         REUSE                             -
         SHR(2)                            -
         CYLINDERS(1)                      -
         VOL(TOTSTJ))                      -
       DATA                                -
        (CISZ(4096))                       -
       INDEX                               -
        (CISZ(1024)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSITRCP)      -
         INDEXED                           -
         KEYS (4,8)                        -
         RECSZ(114,146)                    -
         FSPC(0,0)                         -
         REUSE                             -
         SHR(2)                            -
         CYLINDERS(1)                      -
         VOL(TOTSTJ))                      -
       DATA                                -
        (CISZ(16384))                      -
       INDEX                               -
        (CISZ(512)                         -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSITRCS)      -
         INDEXED                           -
         KEYS (4,8)                        -
         RECSZ(114,146)                    -
         FSPC(0,0)                         -
         REUSE                             -
         SHR(2)                            -
         CYLINDERS(1)                      -
         VOL(TOTSTJ))                      -
       DATA                                -
        (CISZ(16384))                      -
       INDEX                               -
        (CISZ(512)                         -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.DSISVRT)      -
         INDEXED                           -
         SHR(2)                            -
         VOL(TOTSTJ)                       -
         CYLINDERS(2 1)                    -
         KEYS(54 0)                        -
         RECSZ(64 0512)                    -
         FSPC(5 5)                         -
         REUSE)                            -
       DATA                                -
        (CISZ(8192))                       -
       INDEX                               -
        (CISZ(4096)                        -
         IMBED)
DEF CLUSTER(NAME(MSOPS.MSO54.STATS)        -
         VOL(TOTSTJ)                       -
```

```
          KEYS(20 0)                        -
          RECSZ(252 252)                    -
          FSPC(0 0)                         -
          SHR(2)                            -
          CISZ(512)                         -
          INDEXED                           -
          REUSE                             -
          IMBED)                            -
        DATA                                -
         (NAME(MSOPS.MSO54.STATS.DATA)      -
          CYL(2 0))                         -
        INDEX                               -
         (NAME(MSOPS.MSO54.STATS.INDEX)     -
          TRK(2 0))
/* ************************************************************** */
/* Define System SC55 Unique VSAM Clusters - msys/Ops Domain MSO55 */
/* ************************************************************** */
DEF CLUSTER(NAME(MSOPS.MSO55.DSILOGP)       -
          INDEXED                           -
          KEYS (4,8)                        -
          RECSZ(125,404)                    -
          FSPC(0,0)                         -
          REUSE                             -
          SHR(2)                            -
          CYLINDERS(1)                      -
          VOL(TOTSTJ))                      -
        DATA                                -
         (CISZ(4096))                       -
        INDEX                               -
         (CISZ(1024)                        -
          IMBED)
DEF CLUSTER(NAME(MSOPS.MSO55.DSILOGS)       -
          INDEXED                           -
          KEYS (4,8)                        -
          RECSZ(125,404)                    -
          FSPC(0,0)                         -
          REUSE                             -
          SHR(2)                            -
          CYLINDERS(1)                      -
          VOL(TOTSTJ))                      -
        DATA                                -
         (CISZ(4096))                       -
        INDEX                               -
         (CISZ(1024)                        -
          IMBED)
DEF CLUSTER(NAME(MSOPS.MSO55.DSITRCP)       -
          INDEXED                           -
          KEYS (4,8)                        -
          RECSZ(114,146)                    -
          FSPC(0,0)                         -
          REUSE                             -
          SHR(2)                            -
          CYLINDERS(1)                      -
          VOL(TOTSTJ))                      -
        DATA                                -
         (CISZ(16384))                      -
        INDEX                               -
         (CISZ(512)                         -
          IMBED)
DEF CLUSTER(NAME(MSOPS.MSO55.DSITRCS)       -
```

```
          INDEXED                          -
          KEYS (4,8)                       -
          RECSZ(114,146)                   -
          FSPC(0,0)                        -
          REUSE                            -
          SHR(2)                           -
          CYLINDERS(1)                     -
          VOL(TOTSTJ))                     -
        DATA                               -
         (CISZ(16384))                     -
        INDEX                              -
         (CISZ(512)                        -
          IMBED)
 DEF CLUSTER(NAME(MSOPS.MSO55.DSISVRT)     -
          INDEXED                          -
          SHR(2)                           -
          VOL(TOTSTJ)                      -
          CYLINDERS(2 1)                   -
          KEYS(54 0)                       -
          RECSZ(64 0512)                   -
          FSPC(5 5)                        -
          REUSE)                           -
        DATA                               -
         (CISZ(8192))                      -
        INDEX                              -
         (CISZ(4096)                       -
          IMBED)
 DEF CLUSTER(NAME(MSOPS.MSO55.STATS)       -
          VOL(TOTSTJ)                      -
          KEYS(20 0)                       -
          RECSZ(252 252)                   -
          FSPC(0 0)                        -
          SHR(2)                           -
          CISZ(512)                        -
          INDEXED                          -
          REUSE                            -
          IMBED)                           -
        DATA                               -
         (NAME(MSOPS.MSO55.STATS.DATA)     -
          CYL(2 0))                        -
        INDEX                              -
         (NAME(MSOPS.MSO55.STATS.INDEX)    -
          TRK(2 0))
||
//
```

Archived

Step 2: Copy additional PROCs into PROCLIB data set

Five additional PROCs need to be copied into a PROCLIB data set, usually SYS1.PROCLIB, pointed to by the PROC00 DD definition statement in the JES2 procedure.

The sample members are INGNVAP0, HSAPIPLC, INGPHOM, INGPIPLC, and INGPIXCU; they can be found in ING.SINGSAMP.

The first procedure starts msys for Operations and requires installation-specific customization, as highlighted in Example A-2. It may be renamed to anything you choose.

Example: A-2 msys for Operations startup procedure

```
//MSOPS   PROC DOMAIN=MSO&SYSCLONE., ** MSYS/OPS DOMAIN NAME
//             SQ1='NETVIEW',        ** NVSS DSN HLQ
//             SQ3='ING',            ** MSAS DSN HLQ
//             VQ1=MSOPS             ** VSAM DSN HLQ
//MSOPS   EXEC PGM=DSIMNT,TIME=1440,REGION=64M,DPRTY=(13,13),
//             PARM=(24K,200,'&DOMAIN','','','','')
//SYSPRINT DD  SYSOUT=*
//STEPLIB  DD  DSN=&SQ3..SINGMOD1,DISP=SHR
//         DD  DSN=&SQ1..CNMLINK,DISP=SHR
//DSICLD   DD  DSN=&SQ3..SINGNREX,DISP=SHR
//         DD  DSN=&SQ1..CNMCLST,DISP=SHR
//         DD  DSN=&SQ1..CNMSAMP,DISP=SHR
//DSIOPEN  DD  DSN=&SQ1..SDSIOPEN,DISP=SHR
//DSIPARM  DD  DSN=MSOPS.SHARED.DSIPARM,DISP=SHR
//         DD  DSN=&SQ3..SINGNPRM,DISP=SHR
//         DD  DSN=&SQ1..DSIPARM,DISP=SHR
//DSILIST  DD  DSN=MSOPS.&DOMAIN..DSILIST,DISP=SHR
//DSIVTAM  DD  DSN=ESA.SYS1.VTAMLST,DISP=SHR
//DSIPRF   DD  DSN=&SQ3..SINGNPRF,DISP=SHR
//         DD  DSN=&SQ1..DSIPRF,DISP=SHR
//DSIMSG   DD  DSN=&SQ3..SINGNMSG,DISP=SHR
//         DD  DSN=&SQ1..SDSIMSG1,DISP=SHR
//BNJPNL1  DD  DSN=&SQ1..BNJPNL1,DISP=SHR
//BNJPNL2  DD  DSN=&SQ1..BNJPNL2,DISP=SHR
//CNMPNL1  DD  DSN=&SQ3..SINGNPNL,DISP=SHR
//         DD  DSN=&SQ1..CNMPNL1,DISP=SHR
//         DD  DSN=&SQ1..SEKGPNL1,DISP=SHR
//DSILOGP  DD  DSN=&VQ1..&DOMAIN..DSILOGP,
//             DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSILOGS  DD  DSN=&VQ1..&DOMAIN..DSILOGS,
//             DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSITRCP  DD  DSN=&VQ1..&DOMAIN..DSITRCP,
//             DISP=SHR,AMP=AMORG
//DSITRCS  DD  DSN=&VQ1..&DOMAIN..DSITRCS,
//             DISP=SHR,AMP=AMORG
//DSISVRT  DD  DSN=&VQ1..&DOMAIN..DSISVRT,
//             DISP=SHR,AMP=AMORG
//AOFSTAT  DD  DSN=&VQ1..&DOMAIN..STATS,
//             DISP=SHR,AMP=AMORG
//HSAIPL   DD  DSN=&VQ1..SHARED.IPLDATA,
//             DISP=SHR,AMP=AMORG
```

The second procedure should be run immediately after an IPL and causes information related to the IPL to be stored for later retrieval and comparison. Ensure that the correct high level qualifiers (HLQs) are used, and add the entry **COM='S HSAPIPLC'** to COMMNDxx to automatically start this procedure following an IPL.

```
//HSAPIPLC PROC HLQ1=ING,HLQ2=MSOPS
//COLLECT  EXEC PGM=HSAPSIPL,REGION=2M
//STEPLIB  DD  DISP=SHR,DSN=&HLQ1..SINGMOD1
//HSAIPL   DD  DISP=SHR,DSN=&HLQ2..IPLDATA
```

Figure A-1 Procedure to store IPL information

The other procedures require no changes. They are used dynamically for internal processes and should be copied as is. If not copied to SYS1.PROCLIB, ensure that the Proclib data set is concatenated to the IEFPDSI data definition statement in SYS1.PARMLIB(MSTRJCL).

Archived

Step 3: Data sets for LNKLST and LPALST

From the list of data sets shown in Figure A-2, the first three need to be authorized. In addition, the SINGMOD2 data set must be added to the LNKLST concatenation, and the SINGMOD3 data must be set to the LPALST concatenation.

```
ING.SINGMOD1
ING.SINGMOD2    <<< Add to LNKLST Concatenation
NETVIEW.V1R4M0.CNMLINK
ING.SINGMOD3    <<< Add to LPALST Concatenation
```

Figure A-2 Important msys for Operations data sets

Figure A-3 shows how to accomplish this. Statements need to be added to the active PROGxx and LPALSTxx members of SYS1.PARMLIB, which would take effect at the next IPL. To make the changes permanent, add the following statements to the LNKLST and APF sections of the active PROGxx member. Add the single statement to the active LPALSTxx member.

Note: You must be careful to match the LNKLST set name (here, LNKLST00) to the one actually in use, and change the volume parameter to match the volser on which these data sets are allocated.

```
PROGxx LNKLST Change:

LNKLST ADD NAME(LNKLST00) DSN(ING.SINGMOD2) VOLUME(TOTSTJ)

PROGxx APF Changes:

APF ADD DSNAME(ING.SINGMOD1)           VOLUME(TOTSTJ)
APF ADD DSNAME(ING.SINGMOD2)           VOLUME(TOTSTJ)
APF ADD DSNAME(NETVIEW.V1R4M0.CNMLINK) VOLUME(TOTSTJ)

LPALSTxx Change:

ING.SINGMOD3(TOTSTJ)
```

Figure A-3 Permanent APF, LNKLST, and LPALST changes

To make the changes dynamically, the most straightforward approach is to create a new member (PROGMO) based on the example member, INGPROG0 located in ING.SINGSAMP. The following statements, as illustrated in Figure A-4 on page 357, show what is needed. Choose an appropriate LNKLST set name and change the volume parameter to match the volser on which these data sets are allocated. Be aware that dynamic changes to LPA requires the data set to be cataloged.

```
LNKLST DEFINE NAME(LNKLSTMO) COPYFROM(CURRENT)
LNKLST ADD NAME(LNKLSTMO) DSNAME(ING.SINGMOD2) VOLUME(TOTSTJ) ATTOP
LNKLST ACTIVATE NAME(LNKLSTMO)
APF ADD DSNAME(ING.SINGMOD1)                   VOLUME(TOTSTJ)
APF ADD DSNAME(ING.SINGMOD2)                   VOLUME(TOTSTJ)
APF ADD DSNAME(NETVIEW.CNMLINK)                VOLUME(TOTSTJ)
LPA ADD MODNAME(HSAPHARI) DSNAME(ING.SINGMOD3)
LPA ADD MODNAME(HSAPHARR) DSNAME(ING.SINGMOD3)
LPA ADD MODNAME(HSAPSTAR) DSNAME(ING.SINGMOD3)
LPA ADD MODNAME(HSAPSTPC) DSNAME(ING.SINGMOD3)
```

Figure A-4 Dynamic APF, LNKLST, and LPALST changes

Once PROGMO has been set up, issue the command **SET PROG=MO**. This command results in the merging of these changes with current settings.

Note: Dynamic changes only remain effective until the next IPL.

Archived

Step 4: Add a PPT entry

A Program Properties Table (PPT) entry needs to be added to the active SCHEDxx member of SYS1.PARMLIB, as shown in Figure A-5. The statements to perform this activity can be found in the sample INGSCHE0 in ING.SINGSAMP.

```
/*                NVSS DSIMNT                                        */
PPT PGMNAME(DSIMNT)                  /* PROGRAM NAME NETVIEW         */
    KEY(8)                           /* PROTECTION KEY               */
    NOSWAP                           /* NON-SWAPPABLE                */
```

Figure A-5 Sample statements to add PPT entry

Archived

Step 5: Update MPFLST

The active Message Processing Facility List, MPFLSTxx in SYS1.PARMLIB, needs to be updated. If no form of automation is in use, then the statements to perform this activity can be found in sample member INGEMPF in ING.SINGSAMP. The only statements required are shown in Figure A-6.

```
.NO_ENTRY,SUP(NO),RETAIN(YES),AUTO(YES)
.DEFAULT,SUP(YES),RETAIN(YES),AUTO(NO)
```

Figure A-6 Sample statements to update MPFLSTxx

However, if the installation's current processing is reliant on existing MPFLSTxx settings which may be incompatible with the above, then the specific statements shown in Example A-3 need to be worked into the active member instead.

Example: A-3 Specific statements to update MPFLSTxx

```
AOF603D,SUP(NO),RETAIN(NO),AUTO(YES)
AOF*,SUP(NO),RETAIN(NO),AUTO(NO)
IXC247D,SUP(NO),RETAIN(NO),AUTO(YES)
IXC263I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC500I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC501I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC559I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC560I,SUP(NO),RETAIN(NO),AUTO(YES)
IXC*,SUP(NO),RETAIN(NO),AUTO(NO)
IXG257I,SUP(NO),RETAIN(NO),AUTO(YES)
IXG261E,SUP(NO),RETAIN(NO),AUTO(YES)
IXG*,SUP(NO),RETAIN(NO),AUTO(NO)
IEA404A,SUP(NO),RETAIN(NO),AUTO(YES)
IEA405E,SUP(NO),RETAIN(NO),AUTO(YES)
IEA406I,SUP(NO),RETAIN(NO),AUTO(YES)
IEA231A,SUP(NO),RETAIN(NO),AUTO(YES)
IEA230E,SUP(NO),RETAIN(NO),AUTO(YES)
IEA232I,SUP(NO),RETAIN(NO),AUTO(YES)
IEA*,SUP(NO),RETAIN(NO),AUTO(NO)
IEE043I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE037D,SUP(NO),RETAIN(NO),AUTO(YES)
IEE041I,SUP(NO),RETAIN(NO),AUTO(YES)
EE533E,SUP(NO),RETAIN(NO),AUTO(YES)
IEE769E,SUP(NO),RETAIN(NO),AUTO(YES)
IEE889I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE400I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE600I,SUP(NO),RETAIN(NO),AUTO(YES)
IEE*,SUP(NO),RETAIN(NO),AUTO(NO)
```

VTAM requirements

msys for Operations requires VTAM to be operational on every participating system in the sysplex. Fortunately, Cross-System Coupling Facility (XCF) services are most often used for VTAM-to-VTAM communication within a sysplex, thus virtually eliminating the need for any VTAM definitions.

Essentially, the first VTAM to start creates an XCF group called ISTXCF. This forms the basis of communication with other sysplex VTAMs. As subsequent VTAMs initialize, they join the same XCF group and are dynamically recognized.

There are two considerations:

1. XCF signaling must be configured to use a Coupling Facility (CF), channel-to-channel connections (CTCs), or both.
2. **`XCFINIT=NO`** *must not* be specified in the ATCSTRxx member that is used during VTAM initialization.

Archived

Step 6: Define application major nodes for VTAM

Define the msys for Operations application major nodes to VTAM. The statements to perform this activity can be found in the sample INGVTAM member in ING.SINGSAMP, as shown in Figure A-7 on page 361. This will be a new member which can be named anything you choose, such as APMSOxx. It must be placed in the data set pointed to by the VTAMLST DD definition statement in the VTAM procedure. This data set must also be coded on the DSIVTAM data definition statement of the procedure used to start msys for Operations as described in “Step 2: Copy additional PROCs into PROCLIB data set” on page 354.

Once this member is in place, the current ATCCONxx member should be edited to include the new member name just created. This will ensure that VTAM activates these APPLs during startup.

```
*********************************************************************
**    THIS APPLICATION MAJNODE DEFINES msys/Ops (NVSS) TO VTAM      **
**    DOMAIN: MSO&SYSCLONE.                                         **
*********************************************************************
         VBUILD TYPE=APPL
*********************************************************************
* MSYS_OPS MAIN TASK                                                *
*********************************************************************
MSO&SYSCLONE.  APPL AUTH=(VPACE,ACQ,PASS),PRTCT=MSO&SYSCLONE.,       X
               MODETAB=AMODETAB,DLOGMOD=DSIL6MOD,                    X
               APPC=YES,PARSESS=YES,                                 X
               DMINWNL=4,DMINWNR=4,DSESLIM=8,VPACING=10,             X
               AUTOSES=2
*********************************************************************
* MSYS_OPS PRIMARY POI - (PROGRAM OPERATOR INTERFACE)               *
*********************************************************************
MSO&SYSCLONE.PPT APPL AUTH=(NVPACE,SPO),PRTCT=MSO&SYSCLONE.,EAS=1,    X
               MODETAB=AMODETAB,DLOGMOD=DSILGMOD
*********************************************************************
* MSYS_OPS SUBTASKS                                                 *
*********************************************************************
MSO&SYSCLONE.* APPL AUTH=(NVPACE,SPO,ACQ),PRTCT=MSO&SYSCLONE.,EAS=4,  X
               MODETAB,DLOGMOD=DSILGMOD
```

Figure A-7 Definition of msys for Operations application major nodes for VTAM

Alternatively, activation can be performed dynamically using, for example, the command: **V NET,ACT,ID=APMSOXX**

Archived

Step 7: Make determined security definition changes

Now you must make the determined security definition changes. The statements to perform this activity can be found in the sample CNMSAF1 in NETVIEW.DSIPARM as shown in Example A-4. The recommendation is to use an SAF product, and the statements in this member assume Security Server (RACF) is used. If this is not the case, then the statements need to be altered to conform to the product that is in use.

Extensive customization is necessary here, as highlighted in the following example. The highlighted statements also include required additions to the CNMSAF1 sample member.

Example: A-4 msys for Operations security definitions

```
//RONNSAF  JOB (POK,999),NORTHRUP,CLASS=C,MSGCLASS=T,NOTIFY=&SYSUID
//STP1    EXEC PGM=IKJEFT01
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *,DLM=@@
/*
/*********************************************************************
/*  To activate the classes needed for msys for Operations and protect
/*  against unauthorized logon, change 'domain_name' to the domain
/*  name specified in your msys for Operations startup procedure.
/*********************************************************************
SETROPTS CLASSACT(APPL)
SETROPTS CLASSACT(NETCMDS) GRPLIST
RDEF APPL MSO47 UACC(NONE)
RDEF APPL MSO54 UACC(NONE)
RDEF APPL MSO55 UACC(NONE)
/*
/*********************************************************************
/*  To define the task identifiers needed for msys for Operations,
/*  change 'domain_name' to your msys for Operations domain name and
/*  'SSIR_task_name' to the CNMCSSIR task name you specified in
/*  CNMSTYLE.
/*********************************************************************
/*
/*********************************************************************
/*  To define the task identifiers needed for msys for Operations,
/*  change 'domain_name' to your msys for Operations domain name and
/*  'SSIR_task_name' to the CNMCSSIR task name you specified in
/*  CNMSTYLE.
/*********************************************************************
ADDUSER MSO47PPT
ADDUSER MSO54PPT
ADDUSER MSO55PPT
ADDUSER MSO47SIR
ADDUSER MSO54SIR
ADDUSER MSO55SIR
/*
/*********************************************************************
/*  To define the default userid that will be associated with msys for
/*  Operations started tasks, the values 'STC', 'SYS1', 'SAFADMIN' and
/*  'MSO1234' may be changed. In most situations this type of userid
/*  will already be defined.
/*********************************************************************
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
ADDUSER  STC DFLTGRP(SYS1) OWNER(SAFADMIN) PASSWORD(MSO1234) OPERATIONS
SETROPTS RACLIST(STARTED) REFRESH
/*
/*********************************************************************
```

```
/*  msys for Operations autotasks - DO NOT CHANGE
/*********************************************************************
ADDUSER AUTO1
ALTUSER AUTO1    NETVIEW(IC(LOGPROF2) MSGRECVR(NO))
ADDUSER AUTO2
ALTUSER AUTO2    NETVIEW(IC(LOGPROF4) MSGRECVR(NO))
ADDUSER DBAUTO1
ALTUSER DBAUTO1  NETVIEW(IC(LOGPROF4) MSGRECVR(NO))
ADDUSER DSILCOPR
ALTUSER DSILCOPR NETVIEW(MSGRECVR(NO))
ADDUSER AUTBASE
ALTUSER AUTBASE  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTGSS
ALTUSER AUTGSS   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTMON
ALTUSER AUTMON   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTMSG
ALTUSER AUTMSG   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTNET1
ALTUSER AUTNET1  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTREC
ALTUSER AUTREC   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTSYS
ALTUSER AUTSYS   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTLOG
ALTUSER AUTLOG   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTCON
ALTUSER AUTCON   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTRPC
ALTUSER AUTRPC   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTWRK01
ALTUSER AUTWRK01 NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTWRK02
ALTUSER AUTWRK02 NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTWRK03
ALTUSER AUTWRK03 NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTJES
ALTUSER AUTJES   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTSHUT
ALTUSER AUTSHUT  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTXCF
ALTUSER AUTXCF   NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTXCF2
ALTUSER AUTXCF2  NETVIEW(IC(AOFRAAIC) MSGRECVR(NO))
ADDUSER AUTHWO01
ALTUSER AUTHWO01 NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHWO02
ALTUSER AUTHWO02 NETVIEW(IC(INGRX80S) CTL(GLOBAL) OPCLASS(1,2))
/*
/*********************************************************************
/* Edit the following defaults appropriately to define your operators.
/*********************************************************************
ADDUSER RONN
ALTUSER RONN     NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER HIR
ALTUSER HIR      NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER TIL
ALTUSER TIL      NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER FURNEAK
ALTUSER FURNEAK  NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
```

```
ADDUSER NIELSON
ALTUSER NIELSON  NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER WATS
ALTUSER WATS     NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER1
ALTUSER OPER1    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER2
ALTUSER OPER2    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER3
ALTUSER OPER3    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER4
ALTUSER OPER4    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER5
ALTUSER OPER5    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER OPER6
ALTUSER OPER6    NETVIEW(IC(LOGPROF1) MSGRECVR(NO))
ADDUSER NETOP1
ALTUSER NETOP1   NETVIEW(IC(LOGPROF1) MSGRECVR(YES))
ADDUSER NETOP2
ALTUSER NETOP2   NETVIEW(IC(LOGPROF1) MSGRECVR(YES))
/*
/*********************************************************************
/* Group your operators according to their responsibilities and
/* roles.  Add as many CONNECT statements as you need.  Each operator
/* can be connected to as many groups as needed.
/*
/* Operators are connected to a group according the functions they
/* are permitted run. They must also be connected to every lower group
/* as well. For example, Operators connected to MSYSOPS3 would also
/* require access to functions in the lower groups. To establish this
/* they must be connected to MSYSOPS1 and MSYSOPS2 in order to have
/* the appropriate access authority assigned to them.
/*
/* If users are not listed as member of any group, they will still
/* be able to use the INGPLEX and INGCF commands for display
/* purposes and any other msys for Operations commands that are
/* not specifically protected.
/*
/*********************************************************************
/* Users listed in this group are allowed to execute
/* administrative ACF COLD and INGAUTO commands.
/*********************************************************************
ADDGROUP MSYSOPS0
CONNECT  NETOP1   GROUP(MSYSOPS0) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS0) UACC(READ)
CONNECT  OPER1    GROUP(MSYSOPS0) UACC(READ)
CONNECT  OPER2    GROUP(MSYSOPS0) UACC(READ)
CONNECT  OPER3    GROUP(MSYSOPS0) UACC(READ)
CONNECT  OPER4    GROUP(MSYSOPS0) UACC(READ)
CONNECT  OPER5    GROUP(MSYSOPS0) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS0) UACC(READ)
CONNECT  HIR      GROUP(MSYSOPS0) UACC(READ)
CONNECT  TIL      GROUP(MSYSOPS0) UACC(READ)
CONNECT  FURNEAK  GROUP(MSYSOPS0) UACC(READ)
CONNECT  NIELSON  GROUP(MSYSOPS0) UACC(READ)
CONNECT  WATS     GROUP(MSYSOPS0) UACC(READ)
/*********************************************************************
/* Users listed in this group are allowed to execute FORCE and
/* REBUILD actions on structures.
/*********************************************************************
```

```
ADDGROUP MSYSOPS1
CONNECT  NETOP1   GROUP(MSYSOPS1) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS1) UACC(READ)
CONNECT  OPER1    GROUP(MSYSOPS1) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS1) UACC(READ)
CONNECT  HIR      GROUP(MSYSOPS1) UACC(READ)
CONNECT  TIL      GROUP(MSYSOPS1) UACC(READ)
CONNECT  FURNEAK  GROUP(MSYSOPS1) UACC(READ)
CONNECT  NIELSON  GROUP(MSYSOPS1) UACC(READ)
CONNECT  WATS     GROUP(MSYSOPS1) UACC(READ)
/*******************************************************************
/* Users listed in this group are allowed to execute the SETXCF
/* command with parm "ACOUPLE" and "PSWITCH".
/*******************************************************************
ADDGROUP MSYSOPS2
CONNECT  NETOP1   GROUP(MSYSOPS2) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS2) UACC(READ)
CONNECT  OPER2    GROUP(MSYSOPS2) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS2) UACC(READ)
CONNECT  HIR      GROUP(MSYSOPS2) UACC(READ)
CONNECT  TIL      GROUP(MSYSOPS2) UACC(READ)
CONNECT  FURNEAK  GROUP(MSYSOPS2) UACC(READ)
CONNECT  NIELSON  GROUP(MSYSOPS2) UACC(READ)
CONNECT  WATS     GROUP(MSYSOPS2) UACC(READ)
/*******************************************************************
/* Users listed in this group are allowed to execute the full
/* functionality of INGCF ENABLE and INGCF DRAIN (without HW
/* action ACTIVATE/DEACTIVATE).
/*******************************************************************
ADDGROUP MSYSOPS3
CONNECT  NETOP1   GROUP(MSYSOPS3) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS3) UACC(READ)
CONNECT  OPER3    GROUP(MSYSOPS3) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS3) UACC(READ)
CONNECT  HIR      GROUP(MSYSOPS3) UACC(READ)
CONNECT  TIL      GROUP(MSYSOPS3) UACC(READ)
CONNECT  FURNEAK  GROUP(MSYSOPS3) UACC(READ)
CONNECT  NIELSON  GROUP(MSYSOPS3) UACC(READ)
CONNECT  WATS     GROUP(MSYSOPS3) UACC(READ)
/*******************************************************************
/* Users listed in this group are allowed to do most restricted
/* base NVSS commands.
/*******************************************************************
ADDGROUP MSYSOPS4
CONNECT  NETOP1   GROUP(MSYSOPS4) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS4) UACC(READ)
CONNECT  OPER1    GROUP(MSYSOPS4) UACC(READ)
CONNECT  OPER2    GROUP(MSYSOPS4) UACC(READ)
CONNECT  OPER3    GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTO1    GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTO2    GROUP(MSYSOPS4) UACC(READ)
CONNECT  DBAUTO1  GROUP(MSYSOPS4) UACC(READ)
CONNECT  DSILCOPR GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTBASE  GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTGSS   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTMON   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTMSG   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTNET1  GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTREC   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTSYS   GROUP(MSYSOPS4) UACC(READ)
```

Archived

```
CONNECT  AUTLOG   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTCON   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTRPC   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTWRK01 GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTWRK02 GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTWRK03 GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTJES   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTSHUT  GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTXCF   GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTXCF2  GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTHW001 GROUP(MSYSOPS4) UACC(READ)
CONNECT  AUTHW002 GROUP(MSYSOPS4) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS4) UACC(READ)
CONNECT  HIR      GROUP(MSYSOPS4) UACC(READ)
CONNECT  TIL      GROUP(MSYSOPS4) UACC(READ)
CONNECT  FURNEAK  GROUP(MSYSOPS4) UACC(READ)
CONNECT  NIELSON  GROUP(MSYSOPS4) UACC(READ)
CONNECT  WATS     GROUP(MSYSOPS4) UACC(READ)
/*******************************************************************
/* Users listed in this group are allowed to do any commands.
/*******************************************************************
ADDGROUP MSYSOPS5
CONNECT  NETOP1   GROUP(MSYSOPS5) UACC(READ)
CONNECT  NETOP2   GROUP(MSYSOPS5) UACC(READ)
CONNECT  RONN     GROUP(MSYSOPS5) UACC(READ)
CONNECT  HIR      GROUP(MSYSOPS5) UACC(READ)
CONNECT  TIL      GROUP(MSYSOPS5) UACC(READ)
CONNECT  FURNEAK  GROUP(MSYSOPS5) UACC(READ)
CONNECT  NIELSON  GROUP(MSYSOPS5) UACC(READ)
CONNECT  WATS     GROUP(MSYSOPS5) UACC(READ)
/*
/*******************************************************************
/* To allow your operators to log on to msys for Operations, change
/* 'domain_name' to your domain name.  Be sure to add PERMIT
/* statements for any operators not connected to any groups.
/* NOTE: Autotasks do not have to be permitted to log on.
/*******************************************************************
PERMIT MSO47 CLASS(APPL) ID(MSYSOPS0) ACCESS(READ)
PERMIT MSO47 CLASS(APPL) ID(MSYSOPS1) ACCESS(READ)
PERMIT MSO47 CLASS(APPL) ID(MSYSOPS2) ACCESS(READ)
PERMIT MSO47 CLASS(APPL) ID(MSYSOPS3) ACCESS(READ)
PERMIT MSO47 CLASS(APPL) ID(MSYSOPS4) ACCESS(READ)
PERMIT MSO47 CLASS(APPL) ID(MSYSOPS5) ACCESS(READ)
PERMIT MSO54 CLASS(APPL) ID(MSYSOPS0) ACCESS(READ)
PERMIT MSO54 CLASS(APPL) ID(MSYSOPS1) ACCESS(READ)
PERMIT MSO54 CLASS(APPL) ID(MSYSOPS2) ACCESS(READ)
PERMIT MSO54 CLASS(APPL) ID(MSYSOPS3) ACCESS(READ)
PERMIT MSO54 CLASS(APPL) ID(MSYSOPS4) ACCESS(READ)
PERMIT MSO54 CLASS(APPL) ID(MSYSOPS5) ACCESS(READ)
PERMIT MSO55 CLASS(APPL) ID(MSYSOPS0) ACCESS(READ)
PERMIT MSO55 CLASS(APPL) ID(MSYSOPS1) ACCESS(READ)
PERMIT MSO55 CLASS(APPL) ID(MSYSOPS2) ACCESS(READ)
PERMIT MSO55 CLASS(APPL) ID(MSYSOPS3) ACCESS(READ)
PERMIT MSO55 CLASS(APPL) ID(MSYSOPS4) ACCESS(READ)
PERMIT MSO55 CLASS(APPL) ID(MSYSOPS5) ACCESS(READ)
SETROPTS RACLIST(APPL) REFRESH
/*
/*********************************************************************
/*  Add FACILITY class resources for the Internal Hardware Transport &
/*  dynamic CDS functions.
```

```
/*******************************************************************
SETROPTS GENERIC(FACILITY)
RDEF FACILITY HSA.ET32*                UACC(NONE)
RDEF FACILITY MVSADMIN.LOGR            UACC(NONE)
RDEF FACILITY MVSADMIN.XCF.ARM         UACC(NONE)
RDEF FACILITY MVSADMIN.XCF.CFRM        UACC(NONE)
RDEF FACILITY MVSADMIN.XCF.SFM         UACC(NONE)
/*******************************************************************
/* Add RDEF statements to the following list to define resources to
/* the NETCMDS class for any additional commands to which you wish to
/* restrict access.
/*******************************************************************
SETROPTS GENERIC(NETCMDS)
RDEF NETCMDS  *.*.*                     UACC(READ)
RDEF NETCMDS  *.*.ACF.COLD              UACC(NONE)
RDEF NETCMDS  *.*.INGAUTO               UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGCF.STR    UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGPLEX.CDS  UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGCF.CF     UACC(NONE)
RDEF NETCMDS  *.*.INGRCCHK.INGPLEX.HW   UACC(NONE)
RDEF NETCMDS  *.*.REFRESH               UACC(NONE)
RDEF NETCMDS  *.*.DEFAULTS              UACC(NONE)
RDEF NETCMDS  *.*.LOGONPW               UACC(NONE)
RDEF NETCMDS  *.*.TS                    UACC(NONE)
RDEF NETCMDS  *.*.RID                   UACC(NONE)
RDEF NETCMDS  *.*.PURGE                 UACC(NONE)
RDEF NETCMDS  *.*.ALLOCATE              UACC(NONE)
RDEF NETCMDS  *.*.FREE                  UACC(NONE)
RDEF NETCMDS  *.*.EXCMD                 UACC(NONE)
RDEF NETCMDS  *.*.MODIFY                UACC(NONE)
RDEF NETCMDS  *.*.VARY                  UACC(NONE)
RDEF NETCMDS  *.*.CNME2008              UACC(NONE)
RDEF NETCMDS  *.*.MVS                   UACC(NONE)
RDEF NETCMDS  *.*.START                 UACC(NONE)
RDEF NETCMDS  *.*.STOP                  UACC(NONE)
RDEF NETCMDS  *.*.SWITCH                UACC(NONE)
RDEF NETCMDS  *.*.RESETDB               UACC(NONE)
RDEF NETCMDS  *.*.AUTOTBL               UACC(NONE)
RDEF NETCMDS  *.*.AUTOTASK              UACC(NONE)
RDEF NETCMDS  *.*.EZLEF002              UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXCPU       UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXCPU.*     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXIO        UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXIO.*      UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQIN      UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQIN.*    UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQOUT     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXMQOUT.*   UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXSTG       UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.MAXSTG.*     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.SLOWSTG      UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.SLOWSTG.*    UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.REXXSTRF     UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.REXXSTRF.*   UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.TASK         UACC(NONE)
RDEF NETCMDS  *.*.OVERRIDE.TASK.*       UACC(NONE)
RDEF NETCMDS  *.*.CLOSE                 UACC(NONE)
/************* Protect access data sets ******************************
RDEF NETCMDS *.*.READSEC.DSIPARM.*     UACC(NONE)
RDEF NETCMDS *.*.WRITESEC.*            UACC(NONE)
```

```
RDEF NETCMDS *.*.WRITESEC.*.*          UACC(NONE)
/************* General "disallow" statements *************************
RDEF NETCMDS *.*.CNME1087              UACC(NONE)
RDEF NETCMDS *.*.DSIZKNYJ              UACC(NONE)
RDEF NETCMDS *.*.DSIUSNDM              UACC(NONE)
RDEF NETCMDS *.*.FOCALPT               UACC(NONE)
RDEF NETCMDS *.*.NPDA                  UACC(NONE)
/*
/*********************************************************************
/*  Add data set names with special authority requirements. Change
/*  'MSOPS' to the high level qualifier you have chosen.
/*********************************************************************
AD   MSOPS.**                          UACC(READ)
AD   MSOPS.*.DSILOGP                   UACC(READ)
AD   MSOPS.*.DSILOGS                   UACC(READ)
AD   MSOPS.*.DSILIST                   UACC(READ)
/*
/*********************************************************************
/* Add PERMIT statements to the following list to permit the
/* appropriate operators and groups to use the restricted commands
/* you added to the list above.
/*********************************************************************
PE *.*.ACF.COLD            CLASS(NETCMDS)  ID(MSYSOPS0) ACCESS(READ)
PE *.*.INGAUTO             CLASS(NETCMDS)  ID(MSYSOPS0) ACCESS(READ)
PE *.*.INGRCCHK.INGCF.STR  CLASS(NETCMDS)  ID(MSYSOPS1) ACCESS(READ)
PE *.*.INGRCCHK.INGPLEX.CDS CLASS(NETCMDS) ID(MSYSOPS2) ACCESS(READ)
PE *.*.INGRCCHK.INGCF.CF   CLASS(NETCMDS)  ID(MSYSOPS3) ACCESS(READ)
PE *.*.REFRESH             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.DEFAULTS            CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.LOGONPW             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.TS                  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.RID                 CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.PURGE               CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.ALLOCATE            CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.FREE                CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.EXCMD               CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.MODIFY              CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.VARY                CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.CNME2008            CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.MVS                 CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.START               CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.STOP                CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.SWITCH              CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.RESETDB             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.AUTOTBL             CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.AUTOTASK            CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.EZLEF002            CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXCPU     CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXCPU.*   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXIO      CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXIO.*    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQIN    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQIN.*  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQOUT   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXMQOUT.* CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXSTG     CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.MAXSTG.*   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.SLOWSTG    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.SLOWSTG.*  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.REXXSTRF   CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
```

```
PE *.*.OVERRIDE.REXXSTRF.*  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.TASK        CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.OVERRIDE.TASK.*      CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.READSEC.DSIPARM.*    CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.WRITESEC.*           CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.WRITESEC.*.*         CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE *.*.INGRCCHK.INGPLEX.HW  CLASS(NETCMDS)  ID(MSYSOPS4) ACCESS(READ)
PE HSA.ET32*.**             CLASS(FACILITY) ID(SYS1)     ACCESS(CONTROL)
PE MSOPS.*.DSILOGP          CLASS(DATASET)  ID(MSYSOPS5) ACCESS(CONTROL)
PE MSOPS.*.DSILOGS          CLASS(DATASET)  ID(MSYSOPS5) ACCESS(CONTROL)
PE MSOPS.*.DSILIST          CLASS(DATASET)  ID(MSYSOPS5) ACCESS(ALTER)
PE MVSADMIN.XCF.ARM         CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
PE MVSADMIN.XCF.CFRM        CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
PE MVSADMIN.XCF.SFM         CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
PE MVSADMIN.LOGR            CLASS(FACILITY) ID(SYS1)     ACCESS(ALTER)
/*
SETROPTS RACLIST(NETCMDS,FACILITY) REFRESH
@@
//
```

If you prefer, there is an alternative, less robust method of defining msys for Operations userids without using an SAF product. However, be aware that this is considerably less secure and that functions dependent on the Internal Hardware Transport will still require SAF authorization. In“Step 1: Create VSAM and non-VSAM data sets” on page 348, a shared user DSIPARM data set was created.

This data set name was placed ahead of the others in the DSIPARM concatenation in the MSOPS procedure as shown in “Step 2: Copy additional PROCs into PROCLIB data set” on page 354.

Then copy members DSIDMNK (shown in Example A-5) and DSIOPFU (shown in Example A-6) from NETVIEW.V1R4M0.DSIPARM to this data set.

Example: A-5 DSIDMNK member

```
*************************************************************
*  SPECIFY LOGON CHECKING, COMMAND CHECKING and SPAN CHECKING *
*************************************************************
      OPTIONS  OPERSEC=NETVPW,CMDAUTH=SCOPE,OPSPAN=NETV
*     OPTIONS  OPERSEC=SAFDEF,CMDAUTH=SAF,AUTHCHK=SOURCEID
*     OPTIONS  BACKTBL=CNMSBAK1
```

Example: A-6 DSIOPFU member

```
*********************************************************************
*    NAME(DSIOPFU)  SAMPLE(DSIOPFU)  RELATED-TO(DSIOPF)             *
*    DESCRIPTION: DSIPARM SAMPLE FOR CUSTOMER DEFINED OPERATORS.    *
*                                                                   *
*********************************************************************
*                                                                   *
*   THIS SAMPLE IS TO ALLOW CUSTOMERS TO INCLUDE ANY CUSTOMER       *
*   DEFINED OPERATORS INTO DSIOPF.  THERE SHOULD BE A               *
*   "%INCLUDE DSIOPFU" STATEMENT IN DSIOPF IN ORDER FOR THE OPERATOR *
*   STATEMENTS IN THIS SAMPLE TO BE INCLUDED AS PART OF DSIOPF.     *
*                                                                   *
*   PLACE THE OPERATOR STATEMENTS FOR YOUR OPERATORS BELOW.         *
*                                                                   *
*   WARNING: DO NOT CODE AN END STATEMENT IN THIS SAMPLE.           *
*                                                                   *
```

```
*   WARNING: DO NOT PLACE ANY EXISTING NETVIEW OPERATOR DEFINITIONS   *
*            INTO THIS SAMPLE, ESPECIALLY THE AUTOTASK OPERATORS      *
*            CURRENTLY DEFINED FOR SPECIFIC NETVIEW FUNCTIONS.        *
*                                                                     *
*   WARNING: BE SURE THAT THE NAMES YOU GIVE YOUR OPERATORS ARE NOT   *
*            ALREADY BEING DEFINED TO NETVIEW.  NETVIEW WILL ISSUE    *
*            A WARNING MESSAGE DURING INITIALIZATION IF DUPLICATE     *
*            OPERATORS HAVE BEEN DEFINED.                             *
*                                                                     *
***********************************************************************
RONN        OPERATOR    PASSWORD=RONN
            PROFILEN    DSIPROFA
NIELSON     OPERATOR    PASSWORD=NIELSON
            PROFILEN    DSIPROFA
HIR         OPERATOR    PASSWORD=HIR
            PROFILEN    DSIPROFA
TIL         OPERATOR    PASSWORD=TIL
            PROFILEN    DSIPROFA
WATS        OPERATOR    PASSWORD=WATS
            PROFILEN    DSIPROFA
FURNEAK     OPERATOR    PASSWORD=FURNEAK
            PROFILEN    DSIPROFA
```

Make the optional changes to these members as described in Example A-7 on page 370. If use of CF management capabilities (CF Drain & CF Enable) and partitioning of failed systems (IXC102A) are required functions, then the following RACF definitions must be made.

Example: A-7 RACF definitions for optional changes

```
//RONNSAF  JOB (POK,999),NORTHRUP,CLASS=C,MSGCLASS=T,NOTIFY=&SYSUID
//STP1    EXEC PGM=IKJEFT01
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *,DLM=@@
/*
/***********************************************************************
/*  MANDATORY RACF DEFINITIONS REQUIRED BY INTERNAL HARDWARE TRANSPORT
/***********************************************************************
/*  Define the default userid that will be associated with msys for
/*  Operations started tasks, the values 'STC', 'SYS1', 'SAFADMIN' and
/*  'MSO1234' may be changed. In most situations this type of userid
/*  will already be defined.
/***********************************************************************
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
ADDUSER  STC DFLTGRP(SYS1) OWNER(SAFADMIN) PASSWORD(MSO1234) OPERATIONS
SETROPTS RACLIST(STARTED) REFRESH
/*
/***********************************************************************
/*  msys for Operations Autotasks
/***********************************************************************
ADDUSER AUTHW001 NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTHW002 NETVIEW(IC(INGRX805) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTXCF   NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTXCF2  NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTRPC   NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
ADDUSER AUTBASE  NETVIEW(IC(AOFRAAIC) CTL(GLOBAL) OPCLASS(1,2))
/*
/***********************************************************************
/*  Add the FACILITY class resource profile for the Internal Hardware
/*  Transport
/***********************************************************************
```

```
SETROPTS GENERIC(FACILITY)
RDEF FACILITY HSA.ET32*                                UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
/*
/*********************************************************************
/* Set permissions for Internal Hardware Transport resources
/*********************************************************************
PE HSA.ET32*                CLASS(FACILITY) ID(SYS1)     ACCESS(CONTROL)
/*
@@
//
```

Step 8: Alter msys for Operations NVSS style sheet

Customizing NetView System Services (NVSS), the backbone of msys for Operations (HPZ8500), involves altering the NVSS Style Sheet and some additional members. The optional changes are determined by the security scheme chosen and whether the full NetView product is already used by the installation. All members are located in the NETVIEW.DSIPARM data set. The style sheet is CNMSTYLE and always requires changing.

Optional members are DSIDMNK, DSIOPFU, and DSICMPRC. Copy these members into the shared user defined DSIPARM data set, locate the relevant record, and make the changes highlighted in Example A-8.

Example: A-8 msys for Operations style sheet modifications

Customization for NVSS - No NetView License:

```
CNMSTYLE

 1. *DOMAIN = C&NV2I.01<-- Insert *
 DOMAIN = &DOMAIN.<-- Add statement

 2. *NetID = &CNMNETID.<-- Insert *

 3. *SSIname = C&NV2I.CSSIR<-- Insert *
 SSIname = &DOMAIN.SIR<-- Add statement

NOTE: Optional changes are necessary when a SAF product is not used:

DSIDMNK

 1. OPTIONS  OPERSEC=NETVPW,CMDAUTH=SCOPE,OPSPAN=NETV<-- Remove *
*OPTIONS  OPERSEC=SAFDEF,CMDAUTH=SAF,AUTHCHK=SOURCEID<-- Insert *
   *OPTIONS  BACKTBL=CNMSBAK1<-- Insert *

DSIOPFU
 2. opid       OPERATOR    PASSWORD=opid<-- Add set for each operator
               PROFILEN    DSIPROFA
```

Customization for Full NetView 1.4:

```
CNMSTYLE

 1. *DOMAIN = C&NV2I.01<-- Insert *
```

```
 DOMAIN = &DOMAIN.<-- Add statement

 2. *NetID = &CNMNETID.<-- Insert *

 3. *SSIname = C&NV2I.CSSIR<-- Insert *
 SSIname = &DOMAIN.SIR<-- Add statement

 4. TOWER = SA *AON  *MSM  *Graphics *AMI MVScmdMgt  <-- Remove * -> uncomment *SA

 5. *TOWER.SA  = license<-- Insert *

 6. TASK.CNMTAMEL.INIT=N<-- Change Y to N

 7. TASK.DUIDGHB.INIT=N<-- Change Y to N

 8. *%INCLUDE C&NV2I.STGEN<-- Insert *

 9. TASK.CNMCALRT.INIT=Y<-- Change N to Y
DSICMPRC

 1. EZLSPIPC   CMDMDL   MOD=EZLSPIPC,TYPE=R,RES=N,ECHO=N,SEC=BY  <-- Change Y to N
 2. EZLSRTVE   CMDMDL   MOD=EZLSRTVE,TYPE=R,RES=N,ECHO=N,SEC=BY  <-- Change Y to N

DSIDMNK
 1. LOADEXIT NONE  <-- Remove *

NOTE: Optional changes are necessary when a SAF product is not used:

 2. OPTIONS  OPERSEC=NETVPW,CMDAUTH=SCOPE,OPSPAN=NETV<-- Remove *
*OPTIONS  OPERSEC=SAFDEF,CMDAUTH=SAF,AUTHCHK=SOURCEID<-- Insert *
   *OPTIONS  BACKTBL=CNMSBAK1<-- Insert *

DSIOPFU
 1. opid        OPERATOR    PASSWORD=opid<-- Add set for each operator
                PROFILEN    DSIPROFA
```

Customization for NetView 5.1:

When using NetView 5.1 be aware that the DSIDMNK member has been removed. Statements previously in this member are now part of the style sheet. In addition, SAF is not the default security scheme. Because of this there are changes relating to the security setup irrespective of the scheme chosen.

```
CNMSTYLE

 1. *DOMAIN = C&NV2I.01<-- Insert *
 DOMAIN = &DOMAIN.<-- Add statement
2. *NetID = &CNMNETID.<-- Insert *

 3. *SSIname = CNMCSSIR<-- Insert *
 *SSIname = C&NV2I.CSSIR
 SSIname = &DOMAIN.SIR<-- Add statement

 4. *SECOPTS.OPERSEC = NETVPW<-- Insert * when SAF used
 *SECOPTS.OPERSEC = SAFCHECK
 *SECOPTS.OPERSEC = SAFPW
 SECOPTS.OPERSEC = SAFDEF<-- Remove * when SAF used
 *SECOPTS.OPERSEC = MINIMAL

 5. *SECOPTS.CMDAUTH = TABLE.CNMSCAT2<-- Insert *
```

```
 SECOPTS.CMDAUTH = SAF.CNMSBAK1<-- Remove * when SAF used
 *SECOPTS.CMDAUTH = SAF.PASS
 *SECOPTS.CMDAUTH = SAF.FAIL
 *SECOPTS.CMDAUTH = SCOPE.CNMSCOP1

 6. SECOPTS.AUTHCHK = SOURCEID<-- No changes
 *SECOPTS.AUTHCHK = TARGETID

 7. *SECOPTS.OPSPAN = NETV<-- Insert *
 SECOPTS.OPSPAN = SAF<-- Remove *

 8. TOWER =  SA  *AON  *MSM  *Graphics  MVScmdMgt *NPDA  *TARA *NLDM *AMI<-- Remove * (SA)
& Insert *'s
 9. *TOWER.SA  = license<-- Insert *

10. TASK.CNMTAMEL.INIT=N<-- Change Y to N

11. TASK.DUIDGHB.INIT=N<-- Change Y to N

12. *%INCLUDE C&NV2I.STGEN<-- Insert *

NOTE: Optional changes are necessary when a SAF product is not used:

 1. SECOPTS.OPERSEC = NETVPW<-- No changes
 *SECOPTS.OPERSEC = SAFCHECK
 *SECOPTS.OPERSEC = SAFPW

 *SECOPTS.OPERSEC = SAFDEF
*SECOPTS.OPERSEC = MINIMAL

 2. *SECOPTS.CMDAUTH = TABLE.CNMSCAT2<-- Insert *
 *SECOPTS.CMDAUTH = SAF.CNMSBAK1
 *SECOPTS.CMDAUTH = SAF.PASS
 *SECOPTS.CMDAUTH = SAF.FAIL
 SECOPTS.CMDAUTH = SCOPE.CNMSCOP1 <-- Remove *

 3. SECOPTS.AUTHCHK = SOURCEID<-- No changes - default
 *SECOPTS.AUTHCHK = TARGETID

 4. SECOPTS.OPSPAN = NETV<-- No changes - default
 *SECOPTS.OPSPAN = SAF

DSIOPFU
 1. opid        OPERATOR    PASSWORD=opid<-- Add set for each operator
                PROFILEN    DSIPROFA
```

Step 9: Enable msys for Operations functions

Turn on the msys for Operations functions you are interested in. By default, all functions that take automated action ship in disabled mode and are controlled by the AOFCUST member located in ING.SINGNPRM. Extensive comments are included within AOFCUST, as shown in Example A-9. Start by copying this to the shared DSIPARM data set previously created and make the desired changes there.

The AUTO section controls function enablement, as shown in Example A-9. Functions are enabled by removing comments (*) from the statements you wish to enable. Each function has a corresponding detailed section where installation-specific values are coded, such as the actions you permit msys for Operations to take, volumes available for dynamic allocation, data set HLQs, and so on.

Processor and logical partition configurations are also defined to msys for Operations within this member. This is done in the HW section, which has no relationship to the Auto section. During msys for Operations initialization, checks are made to see if z/OS is able to use an internal transport for direct hardware interaction. If so, this capability is enabled.

Note: It is critical that everything defined in this section exactly represents the definitions in HCD, and which are in use by the Support Element (SE), for any given processor. Failure to do this could result in hardware manipulation targeting the wrong CEC/LPAR.

Example: A-9 msys for Operations functions enablement

```
*-----------------------------------------------------------------
* AUTO section
*-----------------------------------------------------------------
* This section contains keywords representing automated
* functions:
*   CDS    Enables recovery of missing CDS allocations.
*   ENQ    Enables recovery of a long running ENQ detection.
*   LOG    Enables recovery of a syslog start failure.
*   LOGGER Enables recovery of a system logger offload
*          condition.
*   PAGE   Enables recovery of a local page dataser shortage.
*   WTO    Enables recovery of WTO/WTOR buffer shortage
*          conditions.
*   XCF    Enables recovery to prevent a sysplex outage when a
*          system leaves the sysplex due to a failure condition.
* Other keywords are not valid.
* These keywords only affect message initiated automation.
* They do not affect automation initiated using INGCF/INGPLEX.
*
* An asterisk '*' placed in column 1 marks a line as comment.
* Example:To enable the WTO/WTOR Buffer Shortage Recovery use:
*
*-----------------------------------------------------------------
AUTO(
 CDS
 ENQ
 LOG
 LOGGER
 PAGE
 WTO
 XCF
)
*-----------------------------------------------------------------
* COMMON section
*-----------------------------------------------------------------
* The definitions within this section are common to all other
* sections.
* Do NOT comment out this section!
*
* The following parameters must be defined per line.
*
* 1.The keyword TEMPHLQ.
```

```
*
*   TEMPHLQ: This keyword introduces a high level qualifier
*            which is used to assemble a data set name for
*            allocating temporary data sets needed by programs
*            running as started tasks.
The qualifier may consist of up to 17 characters
*            according to the OS/390 data set naming rules
*            (hlq1.hlq2.hlq3).
*            There must be only one line containing the TEMPHLQ
*            keyword within AOFCUST.
*
*            Note: Netview must have RACF ALTER access to the
*                  qualifier. The userid of the started tasks
*                  must have RACF UPDATE access to the qualifier.
*
* 2.The keyword STCJOBNM
*
*   STCJOBNM: This keyword introduces the job name being used
*             for programs running as started tasks.
*             The qualifier may consist of up to 8 characters
*             according to the OS/390 job naming rules.
*             There must be only one line containing the
*             STCJOBNM keyword within AOFCUST.
*             When not defined the job name of each started task
*             defaults to the procedure name.
*
*--------------------------------
COMMON(
 TEMPHLQ (MSOPS.TEMP)
)
*-----------------------------------------------------------------
* WTOBUF section
*-----------------------------------------------------------------
* The definitions within this section will be applied if 'WTOBUF'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
* With each statement you can define address spaces handled
* separately at the resolution of WTO/WTOR buffer shortage
* conditions.
* Three parameters must be defined per line.
* 1.The address space name. A wildcard character (i.e.'*') is
*   supported at the end of this parameter.
* 2.WTO,WTOR or *. This parameter indicates if the automation
*   shall be applied to WTO buffer shortage conditions,
*   or to WTOR buffer shortage conditions
*   or to both.
* 3.KEEP or CANCEL. This parameter indicates if an address
*   space must be kept or canceled upon a buffer shortage
*   condition which is caused by this address space.
*
* Note:1. The default for all job names not listed in this section
*      is KEEP.
*      2. If there are multiple statements within the
*      WTOBUF section they either should define all
*      job names of which address spaces to be kept (KEEP) or
*      they should define only those to be canceled (CANCEL).
*      3. '* * CANCEL' will override the default to cancel all
*      address spaces. It should be the only statement
*      within the WTOBUF section.
*
```

```
* If AOFCUST is shared between the images within a sysplex,
* these definitions are valid sysplex wide.It must be considered
* that the jobnames for your applications can be different on
* each system.
*
*
* Example:Applications with jobname CICS and jobname beginning
*         with IMS must not be canceled (must be kept) upon
*         WTO and WTOR buffer shortage conditions.
*         All other jobs can be canceled.
*
* WTOBUF(
*   CICS   *   KEEP
*   IMS*   *   KEEP
* )
*-----------------------------------------------------------------
WTOBUF(
   *     WTOR CANCEL
   *     WTO  CANCEL
)
*-----------------------------------------------------------------
* CDS section
*-----------------------------------------------------------------
* The definitions within this section will be applied if 'CDS'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section CDS related automation can be customized.
* Two parameters must be defined per line.
*
* 1.The keyword HLQ or VOL.
*
*   HLQ:This keyword introduces a high level qualifier
*       which is used to assemble a data set name for
*       creating and/or allocating an alternate CDS.
*       The qualifier may consist of up to 26 characters
*       according to the OS/390 data set naming rules
*       (hlq1.hlq2.hlq3).
*       The qualifier is appended by the CDS type and '.CDSOn'
*       where 'n'is a sequence number.
*       There must be only one line containing the HLQ keyword
*       within AOFCUST. The HLQ keyword must be the first word
*       in line.
*
*   VOL:This parameter introduces a list of volume names
*       per CDS type. The list contains the names of volumes
*       which are eligible when automation is going to
*       creating and/or allocate an alternate CDS.
*       There may be multiple 'VOL'definitions but only one
*       per type and per line.
*       The VOL keyword must be the first word in line.
*
* 2.The list of volume names per CDS type
*
*   (cds_type,vol1,vol2,vol3,vol4,vol5,vol6,vol7,vol8)
*
*     'cds_type' represents the CDS type.
*     In case of automation an alternate CDS of this type
*     will be allocated.
*     Valid values are SYSPLEX,ARM,CFRM,LOGR or SFM.
```

```
*
*     'vol1,vol2,vol3,...' represent volume names.
*     The volume names must be in accordance to the OS/390
*     volume naming rules.
*     A maximum of 8 volumes can be defined.
*
* Note: Do not specify SMS managed volumes.
*       Do not specify volumes which already contain allocated
*        couple data sets.
*
*-------------------------------
CDS(
 HLQ   SYS1.MSOPS.CDS,
 VOL  (CFRM,TOTDS3,TOTDS2),
 VOL  (ARM,TOTST5),
 VOL  (LOGR,TOTDS1,TOTDS4),
 VOL  (SFM,TOTST1,TOTST2),
VOL  (SYSPLEX,TOTDS0,TOTDS1)
)
*-----------------------------------------------------------------
* ENQ section
*-----------------------------------------------------------------
* The definitions within this section will be applied if 'ENQ'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section ENQ related automation can be customized.
*
* 1.The keywords DUMP, JOB, RES, SYMDEF and TITLE.
*
*   DUMP:  This keyword defines the DUMP option being used
*          for the SDATA parameter on the dump command. It
*          applies for the JOB(job,DUMP) statements only.
*          (Range: Any combination of SDATA dump values)
*
*   JOB:   This parameter defines what jobs cannot be cancelled
*          (JOB(job,KEEP)), what jobs can be cancelled without a
*          dump (JOB(job,NODUMP)), what jobs can be cancelled
*          with a dump using the default dump options
*          (JOB(job,DUMP), and finally what jobs can be
*          cancelled with a dump using the IEADMCxx PARMLIB
*          members.
*          (Range: jobnames including wild cards or 4 character
*                  address space ids)
*
*   RES:   This parameter defines the major and minor resources
*          being checked for long running ENQs. The time value
*          is the time after an ENQ is treated as long running.
*          (Range: 1 to 8  characters major resource name,
*                  1 to 50 characters minor resource name,
*                  30 to 999 seconds wait time)
*
*   SYMDEF:This parameter defines a system sysmbol and its
*          substitution value.
*          (Range: Any valid symbol definition)
*
*   TITLE: The parameter defines the title of each dump being
*          taken with the default dump options.
*          (Range: Up to 100 characters in mixed case)
*
```

```
*   JOB, RES, and SYMDEF statements can be defined as much as
*   needed. For DUMP, SYMDEF, and TITLE statements refer to
*   the MVS DUMP command for more details.
*
* 2.The default values for parameters being omitted
*
*     DUMP:  CSA GRSQ RGN SQA NOSUM TRT
*
*     JOB:   *,DUMP
*            Only, when no "JOB(*,..)" statement has been defined
*
*     TITLE: Dump by msys for Operations due to a lonq ENQ detection
*
*
* Example:
*
* ENQ(
*   DUMP (sdata_values)
*   JOB (0001,KEEP)
*   JOB (CICS*,KEEP)
*   JOB (IMS*,NODUMP)
*   JOB (ABC*,D0,D1)
*   JOB (*,DUMP)
*   RES (MAJOR1,MINOR1,30)
*   RES (MAJOR1,MINOR*,60)
*   RES (MAJOR2,*,120)
*   RES (MAJOR3*,*,300)
*   RES (*,*,999)
*   SYMDEF (*,&DUMPID01.='GLOBAL')
*   SYMDEF (*,&DUMPID02.='LOCAL')
*   TITLE (Dump by automation due to a long ENQ detection)
* )
*--------------------------------
ENQ(
 RES    (MSOPS*,*,45)
 RES    (MSOPSECA,*,30)
 DUMP   (CSA,GRSQ,RGN,SQA,NOSUM,TRT)
 JOB    (MSOPS*,DUMP)
 JOB    (*,NODUMP)
 TITLE  (MSOPS Initiated - Lonq ENQ Detection)
)
*-----------------------------------------------------------------
* PAGE section
*-----------------------------------------------------------------
* The definitions within this section will be applied if 'PAGE'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section PAGE related automation can be customized.
*
* 1.The keywords CYL, DSN, HLQ, JOB, and VOL.
*
*   CYL:This parameter defines the maximum number of cylinders
*       used for the space allocation to format a local page
*       data set dynamically. The minimum/maximum/default values
*       are 100/999/400. The recovery uses the maximum
*       available space between the minimum and the specified
*       value.
*       Note: On a 3390 DASD 100 CYLS are adequate to 70 MB.
*             The formatting process lasts approximately 18
```

```
*             seconds for this amount of space.
*
*   DSN:This parameter defines a pre-formatted spare page
*       data set to be used in the recovery situation.
*       Note: The data set must be allocated on a volume
*             shared by all systems in the sysplex.
*
*   HLQ:This keyword introduces a high level qualifier
*       which is used to assemble a data set name for
*       creating and allocating a page data set.
*       The qualifier may consist of up to 23(!) characters
*       according to the OS/390 data set naming rules
*       (hlq1.hlq2.hlq3).
*       The qualifier is appended the system name followed by
*       '.Vvvvvvv.Snn' where 'v' is the volume serial number and
*       'n' sequence number from 00 to 99.
*       Note: The high level qualifier must point to the master
*             catalog and must not be SMS managed.
*
*   JOB:This parameter defines what jobs cannot be cancelled
*       (KEEP) or can be cancelled (CANCEL) in case the
*       auxiliary shortage condition cannot be resolved and
*       the job is one of those jobs with the most rapidly
*       increasing storage requirements.
*       Trailing wild card is supported.
*
*   VOL:This parameter introduces a list of volume names
*       which are eligible when automation is going to
*       creating and allocate a new page data set.
*       The volume names must be in accordance to the OS/390

*       volume naming rules.
*       Note: The volume must be shared by all systems in the
*             sysplex.
*
*   All keywords except HLQ can be defined as many as needed.
*
* 2.The default values for parameters being omitted
*
*   CYL:   400
*
*   JOB:   *,KEEP
*          Only, when no "JOB(*,..)" statement has been defined
*
*
* Example:
*
* PAGE(
*   DSN (dsn1)
*   DSN (dsn2)
*   HLQ (hlq1.hlq2.hlq3)
*   CYL (nnn)
*   JOB (ABC,KEEP)
*   JOB (ABC*,CANCEL)
*   JOB (*,KEEP)
*   VOL (vol111,vol112,vol113)
*   VOL (vol211)
*   VOL (vol311,vol312)
* )
*--------------------------------
```

Archived

```
PAGE(
 HLQ (SYS1.MSOPS.PAGE)
 CYL (400)
 VOL (TOTTS2,TOTTS3)
 JOB (MSOPS*,CANCEL)
 JOB (*,KEEP)
 DSN (SYS1.MSOPS.LPAGE.VTOTTS1)
)
*-----------------------------------------------------------------
* HW section
*-----------------------------------------------------------------
* The definitions within this section will be applied independent
* of the definitions in the AUTO section. They apply to coupling
* facility related functions as well as to the automation of the
* message IXC102A. The definitions must reflect the actual
* hardware configuration. Otherwise the appropriate functions
* will not work properly.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section HW related automation can be customized.
*
* 1.The keywords CPC and IMAGE.
*
*   CPC:  This parameter defines a CPC (Central Processor
*         Complex).
*
*   IMAGE:This parameter defines an operating system running on
*         a CPC. To indicate that CPC runs in basic mode you
*         must define the CPC name as the LPAR name.
*
*   You can define as many CPC and IMAGE statements as needed
*   but only one per line.
*
* 2.The parameters in particular:
*
*   authcomm  - the authentication value being used for
*               communication with the Support Element.
*               Note: This value must match the value specified
*                     in the Support Element for SNMP communication.
*                     Keep in mind that SNMP handles upper case
*                     and lower characters differently.
*
*   cpcname   - the name of CPC hosting the operating system.
*
*   imagetype - the type of the operating system, either CF,
*               MVS, or OTHER.
*
*   lparname  - the name of LPAR running the operating system.
*               Note: When a CPC runs in basic mode specify the CPC
*                     name as the LPAR name.
*
*   netid.nau - the network address of the Support Element of
*               the CPC.
*
*   plexname  - the name of the sysplex.
*
*   sysname   - the name of the operating system.
*
*
* Example:
```

```
*
* HW(
*   CPC   (cpcname,netid.nau,authcomm)
*   IMAGE (sysname,lparname,cpcname,plexname,imagetype)
* )
*--------------------------------
HW(
   CPC   (P701,USIBMSC.SCZP701,AIBSNMP)
   CPC   (P801,USIBMSC.SCZP801,AIBSNMP)
   IMAGE (SC54,A1 ,P701,WTSCPLX1,MVS)
   IMAGE (SC55,A2 ,P701,WTSCPLX1,MVS)
   IMAGE (SC49,A3 ,P701,WTSCPLX1,MVS)
*  IMAGE (SC59,A4 ,P701,OPPLEX  ,MVS)
   IMAGE (SC04,A5 ,P701,WTSCPLX1,MVS)
*  IMAGE (SC58,A6 ,P701,PLEX58  ,MVS)
   IMAGE (SC61,A7 ,P701,WTSCPLX1,MVS)
   IMAGE (SC62,A8 ,P701,WTSCPLX1,MVS)
   IMAGE (SC67,A9 ,P701,WTSCPLX1,MVS)
*  IMAGE (SC57,I1 ,P701,PLEX57  ,MVS)
   IMAGE (SC69,A11,P701,WTSCPLX1,MVS)
   IMAGE (SC47,A12,P701,WTSCPLX1,MVS)
*  IMAGE (    ,A13,P701,        ,OTHER)
*  IMAGE (CF07,C1 ,P701,        ,CF)
   IMAGE (CF05,C2 ,P701,WTSCPLX1,CF)
*
   IMAGE (SC52,A1 ,P801,WTSCPLX1,MVS)
   IMAGE (SC53,A2 ,P801,WTSCPLX1,MVS)
   IMAGE (SC50,A3 ,P801,WTSCPLX1,MVS)
*  IMAGE (    ,A4 ,P801,        ,OTHER)
   IMAGE (SC42,A5 ,P801,WTSCPLX1,MVS)
   IMAGE (SC43,A6 ,P801,WTSCPLX1,MVS)
   IMAGE (SC66,A7 ,P801,WTSCPLX1,MVS)
*  IMAGE (SC63,A8 ,P801,SANDBOX ,MVS)
*  IMAGE (SC64,A9 ,P801,SANDBOX ,MVS)
*  IMAGE (SC65,A10,P801,SANDBOX ,MVS)
   IMAGE (SC48,A11,P801,WTSCPLX1,MVS)
*  IMAGE (    ,A12,P801,        ,OTHER)
*  IMAGE (CF04,C1 ,P801,SANDBOX ,CF)
*  IMAGE (CF03,C2 ,P801,SANDBOX ,CF)
   IMAGE (CF06,C3 ,P801,WTSCPLX1,CF)
)

*-----------------------------------------------------------------
* IXC102A section
*-----------------------------------------------------------------


* The definitions within this section will be applied if 'XCF'
* is defined in the AUTO section.
* An asterisk '*' placed in column 1 marks a line as comment.
*
* In this section IXC102A related automation can be customized.
*
* 1.The keywords CMD, DISABLE, and ENABLE.
*
*   CMD:    This parameter defines the command being sent to the
*           Support Element when the IXC102A message is trapped.
*           Even the Support Element accepts many commands only
*           the next four commands are accepted by the automation
*           to solve the recovery situation:
```

```
*
*             ACTIVATE [PN(image_profile_name)]
*
*             DEACTIVATE
*
*             SYSRESET [CLEAR]
*
*             LOAD [P(load_profile_name)] [CLEAR]
*
*           As the Support Element treats the affected system
*           still as running the automation automatically appends
*           each command with the parameter FORCE forcing the
*           Support Element to accept the command anyway.
*
*   DISABLE:This parameter defines the system(s) being excluded
*           from automation when an IXC102A message is issued for
*           the system.
*
*   ENABLE: This parameter defines the system(s) being automated
*           when an IXC102A message is issued for the system.
*
*
*   You can define DISABLE and ENABLE statements as many as needed
*   but only one per line.
*   You can also have multiple 'CMD' definitions but only one per
*   system and per line.
*
*
* 2.The parameters in particular:
*
*   image_profile_name - the IMAGE profile name designates
*                        information stored in the Support Element
*                        used to build a logical partition in a
*                        CPC and to IPL an operating system
*
*   load_profile_name  - the LOAD profile name designates
*                        information stored in the Support Element
*                        used to IPL an operating system
*
*   CLEAR              - clears the main storage
*
*
*   When you omit the profile name the Support Element uses the
*   profile which was used at the last operation.
*
*
* 3.The default values for parameters being omitted
*
*   The default command for a system defined in the HW section
*   is 'SYSRESET CLEAR'.
*
*
*
* Example:
*
* IXC102A(

*   CMD     (sysname5,'command')
*   CMD     (sysname6,'command')
*   DISABLE (sysname1,sysname2,sysname3)
```

```
*   DISABLE (sysname4)
*   ENABLE  (sysname5,sysname6)
*   ENABLE  (sysname7)
* )
*-------------------------------
IXC102A(
  CMD     (SC47,LOAD CLEAR)
  CMD     (SC54,LOAD)
  CMD     (SC55,SYSRESET)
  ENABLE  (SC04,SC42,SC43,SC47,SC48,SC49)
  ENABLE  (SC50,SC52,SC53,SC54,SC55)
  ENABLE  (SC61,SC62,SC66,SC67,SC69)
```

Step 10: Build the VTAM logon mode table AMODETAB

This step is optional, depending on whether prior installation actions have been taken. If the installation is currently running NetView as part of the normal process, then the actions described here will most likely have been done. If not, perform these additional tasks to build the VTAM logon mode table AMODETAB. This table defines the session protocols for the different devices and applications used by msys for Operations.

- Start by verifying whether AMODETAB is already in place. Browse the JCL statements used to start VTAM. Locate the VTAMLIB DD definition statement (which may address a single data set or multiple data sets concatenated together). Browse the applicable data sets, checking for a member AMODETAB. If found, browse the actual member and issue a FIND DSIL6MOD browse command. If everything checks out, you are done.

  `If nothing is found, then proceed with the creation of this member.`

  > **Note:** If AMODETAB is found but an entry for DSIL6MOD is not, then steps will need to be taken in conjunction with the installation's Networking Group to modify the current AMODETAB to include the DSIL6MOD statements referenced below.

- Prepare a data set into which AMODETAB will be linked. Although this can be done directly into SYS1.VTAMLIB, the recommendation is to create a user-defined VTAMLIB data set with the same attributes as SYS1.VTAMLIB, and make this the first data set in the VTAMLIB concatenated list.

  If a user-defined VTAMLIB data set already exists, then you are done here and can proceed to the next bullet.

  > **Note:** Remember that SYS1.VTAMLIB is authorized. Any new data set that is made part of the same concatenation must also be authorized. Refer to “Step 2: Copy additional PROCs into PROCLIB data set” on page 354 for information on how to do this.

- Compile and linkedit AMODETAB into the chosen data set. Two sample members, CNMSJ006 (JCL) and CNMS0001 (AMODETAB source), can be found in NETVIEW.V1R4M0.CNMSAMP. The job shown in Figure A-8 on page 384 is based on CNMSJ006, but reduced to the specific statements required. Make similar changes applicable to your installation and run the job to create the AMODETAB member. If the SYSLMOD data set that you chose already exists and was part of the procedure used to start VTAM, you are done.

  Otherwise, complete the final task.

```
//AMODETAB JOB (034D000,TS),NORTHRUP,CLASS=C,MSGCLASS=T,
//             REGION=6M,NOTIFY=&SYSUID
//ASM     EXEC PGM=ASMA90,PARM='NODECK,OBJECT'
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=SYS1.SISTMAC1,DISP=SHR
//SYSUT1   DD  UNIT=3390,SPACE=(CYL,(1,1))
//SYSUT2   DD  UNIT=3390,SPACE=(CYL,(1,1))
//SYSUT3   DD  UNIT=3390,SPACE=(CYL,(1,1))
//SYSLIN   DD  DSN=&&SYSGO,DISP=(,PASS),UNIT=3390,SPACE=(CYL,(1,1))
//SYSIN    DD  DSN=NETVIEW.V1R4M0.CNMSAMP(CNMS0001),DISP=SHR
//*
//LINK    EXEC PGM=HEWL,PARM='LIST,MAP,XREF,RENT',COND=(4,LT)
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  SPACE=(CYL,(1,1)),DISP=(NEW,PASS),UNIT=3390
//SYSLMOD  DD  DSN=MSOPS.CUSTOM.VTAMLIB(AMODETAB),DISP=SHR
//SYSLIN   DD  DSN=&&SYSGO,DISP=(OLD,DELETE)
//
```

Figure A-8 Job to create the AMODETAB member

Note: If your installation already has an AMODTAB defined but is missing the table entry for DSIL6MOD, the MODEENT statements shown in Example A-10 must be added to it.

Example: A-10 Logmode entry for LU 6.2 type applications

```
************************************************************************
*                                                                      *
*   LOGMODE ENTRY FOR 6.2 APPLICATIONS                                 *
*                                                                      *
************************************************************************
DSIL6MOD MODEENT LOGMODE=DSIL6MOD,FMPROF=X'13',TSPROF=X'07',           X
               PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'50B1',TYPE=X'00', X
               SSNDPAC=X'00',SRCVPAC=X'03',PSNDPAC=X'03',              X
               RUSIZES=X'8888',PSERVIC=X'06020000000000000002C00'
```

- Authorize the new user-defined VTAMLIB data set. Add it at the front of the VTAMLIB DD definition statements in the procedure used to start VTAM.

Step 11: REXX environment table entries

This step is optional, depending on whether prior installation actions have been taken. If the installation is currently running NetView as part of the normal process, then the actions described here will most likely have been done. If not, the number of entries in the REXX environment table (IRXANCHR) will need to be checked and increased, if necessary. The following steps can be used to determine the present value that is set; you can increase it, if you need to:

- Browse SYS1.LINKLIB(IRXANCHR). On the command line, type `HEX` and press Enter. The screen shown in Figure A-9 on page 385 is displayed. Check the highlighted value on your display. In this example, the change had already been applied: x'01F4' is a value of 500. If the value in your display is x'0190' (a value of 400) or more you are done.

  Otherwise, proceed to the next step to increase the number of table entries.

```
BROWSE    SYS1.LINKLIB(IRXANCHR)                    Line 00000000 Col 001 080
 Command ===>                                                Scroll ===> CSR
********************************* Top of Data **********************************


 ------------------------------------------------------------------------------
.Ø......IRXANCHR......+
28000001CDECDCCD00000044
00000100997153890000060E0
 ------------------------------------------------------------------------------
Ø ............................................................................
8F0000000000000000000000000000000000000000000000000000000000000000000000000000
0A1000000000000000000000000000000000000000000000000000000000000000000000000000
 ------------------------------------------------------------------------------
Ø..5695DF108 ......Ä.?
810FFFFCCFFF4000010636
05256954610801403F133F
 ------------------------------------------------------------------------------
Ø.dØ..569623400 .....
818800FFFFFFFFF400001
04401056962340001303F
 ------------------------------------------------------------------------------
............ .+ ..+
00000000000040440044
D0000400600000E001E0
 ------------------------------------------------------------------------------
IRXANCHR0100...4.............................................................
CDECDCCDFFFF000F0000000200000000000000000000000000000000000000000000000000000
9971538901000014000000080000000000000000000000000000000000000000000000000000
 ------------------------------------------------------------------------------
******************************** Bottom of Data ********************************
```

Figure A-9 Browse SYS1.LINKLIB(IRXANCHR)

- Apply the following SMP/E USERMOD, shown in Example A-11, to increase the number of table entries. The sample, IRXTSMPE, can be found in SYS1.SAMPLIB.
- Determine the FMID of MOD entry IRXANCHR from SMP/E. For z/OS V1R2, this is HTE26D2.
- Make the highlighted changes, apply the USERMOD, and then either re-IPL or issue a **F LLA,REFRESH** command after copying the updated IRXANCHR into the current SYS1.LINKLIB.

Note: Do not attempt to use AMASPZAP to make these changes. The number of table entries coded determines the size of this load module—and incorrectly changing it will result in a system that you will *not* be able to log on to.

Example: A-11 Applying SMP/E USERMOD

```
++ USERMOD (TSOANCH)
 /*******************************************************************
  USERMOD TO CHANGE THE NUMBER OF ENTRIES IN THE IRXANCHR TABLE TO
  SUPPORT A LARGER NUMBER OF CONCURRENT ENVIRONMENTS.
 *******************************************************************/.
++ VER (Z038) FMID(HTE26D2)     /* REPLACE XXXXXXX WITH YOUR CURRENT
                                 SYSTEM FMID             @YA60165*/.
++ SRC (IRXANCHR) SYSLIB(SAMPLIB) DISTLIB(ASAMPLIB).
```

```
       TITLE 'IRXANCHR - THE REXX ENVIRONMENT TABLE'
       MACRO
       IRXANCHR &ENTRYNUM=40
*/***START OF SPECIFICATIONS*******************************************/
*/*                                                                   */
*/*  MACRO-NAME = IRXANCHR                                            */
*/*                                                                   */
*/*  COPYRIGHT =                                                      */
*/*    5685-085 COPYRIGHT IBM CORP. 1991                              */
*/*    THIS PRODUCT CONTAINS RESTRICTED MATERIALS OF IBM,             */
*/*    REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083.         */
*/*                                                                   */
*/*  DESCRIPTIVE-NAME = Macro to build IRXANCHR                       */
*/*                                                                   */
*/*  FUNCTION = IRXANCHR is a macro to be used by an installation     */
*/*             to create the REXX Environment Table.  If an          */
*/*             installation decides that the default number of       */
*/*             permitted REXX environments is too small (or too      */
*/*             large) they can update it via this macro.             */
*/*                                                                   */
*/*  INSTALLATION =                                                   */
*/*             This macro as shipped in SYS1.SAMPLIB will create     */
*/*             (via SMP/E) a new IRXANCHR load module with 40        */
*/*             REXX environments.  To change the number of allowable*/
*/*             environments, you must:                               */
*/*                                                                   */
*/*                  1. change the ENTRYNUM= parameter on the         */
*/*                     IRXANCHR macro invocation at the end of this */
*/*                     sample to the desired value (default is 40.) */
*/*                                                                   */
*/*                  2. change the FMID in the ++VER line to the      */
*/*                     FMID of your current TSO/E release.           */
*/*                                                                   */
*/*                  3. install following the instructions for SMP/E */
*/*                     user modifications.                           */
*/*                                                                   */
*/*  INVOCATION = MACRO SPECIFICATION IS:                             */
*/*                                                                   */
*/*               IRXANCHR ENTRYNUM=nn                                */
*/*                                                                   */
*/*               ENTRYNUM=nn specifies the number                    */
*/*               of elements for the array.                          */
*/*                                                                   */
*/*               ENTRYNUM=40 is the default.                         */
*/*                                                                   */
*/*  CHANGE ACTIVITY =                                                */
*/*                                                                   */
*/*        OY36194 -  Created for TSO/E Version 2 Release 1  @YA36194*/
*/*                                                                   */
*/*        OY51911 -  Versioned into JTE23X2.  FMID on ++VER         */
*/*                   statement updated to JTE23X2           @YA51911*/
*/*                                                                   */
*/*        OY60165 -  Changed the FMID to XXXXXXX which the          */
*/*                   installation will replace with their          */
*/*                   current TSO/E FMID.                    @YA60165*/
*/*                                                                   */
*/***END OF SPECIFICATIONS*********************************************/
&ID      SETC  'IRXANCHR'            eye catcher
&VERSION SETC  '0100'                version number
&TOTAL   SETA  &ENTRYNUM             number of entries in table
```

```
&LENGTH  SETA  40                    size of each entry
&ID      CSECT                       this is a load module
&ID      AMODE 31                    AMODE = 31 bit addressing
&ID      RMODE ANY                   RMODE = anywhere
ID       DC   CL8'&ID'               insert eye catcher
VERSION  DC   CL4'&VERSION'          insert version number
TOTAL    DC   F'&TOTAL'              total number of entries
USED     DC   F'0'                   number of used entries (0)
LENGTH   DC   F'&LENGTH'             length of each entry
        DC   XL8'0'                 RESERVED
FIRST    DS   0D                     first entry: double word boundry
        DC   (&TOTAL)XL&LENGTH'0'   total entries
        MEND                        end of macro
*/*****************************************************************/
*/*                                                               */
*/*  IRXANCHR - The REXX environment table                        */
*/*                                                               */
*/*  To change the number of allowable REXX environments, you must: */
*/*                                                               */
*/*                                                               */
*/*                  1. change the ENTRYNUM= parameter on the     */
*/*                     IRXANCHR macro invocation to the desired  */
*/*                     value (default is 40.)                    */
*/*                                                               */
*/*                  2. change the FMID in the ++VER line to the  */
*/*                     FMID of your current TSO/E release.       */
*/*                                                               */
*/*                  3. install following the instructions for SMP/E */
*/*                     user modifications.                       */
*/*                                                               */
*/*****************************************************************/
        IRXANCHR ENTRYNUM=400
        END
```

Step 12: Perform hardware customization on SEs

This step describes the hardware customization that must be performed on every Support Element (SE) that was defined in the AOFCUST policy in “Step 9: Enable msys for Operations functions” on page 373.

Note: The panels shown in this section may differ slightly from the ones you see in your environment, as they tend to change over time with the advent of new HMC levels.

1. Log on as ACSADMIN. If you do this from the change management HMC, every CPC on the HMC/SE LAN can be reached. These changes must be performed on every CPC where SCLP/SNMP hardware control is desired. Figure A-10 on page 388 shows the HMC logon panel.

Figure A-10 HMC Panel: Hardware Management Console Logon

2. Get into **Groups** -> **Defined CPCs**.

Figure A-11 HMC Panel: Hardware Management Console Workplace (1)

3. Select the CPC that you are changing and drag to (or double-click) **Single Object Operations**. This takes you into the SE for that CPC. (These changes can also be performed directly at the SE by logging on there). When making changes to multiple CPCs, however, using the HMC is more convenient.

Archived

Figure A-12 HMC Panel: Hardware management Console Workplace (2)

Now you see Figure A-13, the Single Task Operations Coordination panel.

Figure A-13 HMC Panel: Single Task Operations Coordination

4. Get into **Console Actions** -> **Support Element Settings**.

Figure A-14 HMC Panel: Primary Support Element Workplace (1)

5. Make note of the Primary SE IP Address, as it is required when creating the first community name. Now select the **API** tab.

Figure A-15 HMC Panel: Support Element Settings (1)

6. As shown in Figure A-16 on page 391, **Enable the Support Element Console Application Program Interface** must be selected. Fill In the **Community Name.** The value coded here must match the value coded in the HW Section of AOFCUST on the CPC statement. This field is case-sensitive; use upper case insert, or overwrite any existing value. SNMP agent parameters should already be set. If they are not, set them as shown.

Note that changes on this panel require the SE to be rebooted, so do that when setup is complete. Click **Apply**.

Figure A-16 HMC Panel: Support Element Settings (2)

Figure A-17 Event notification information

7. Get into **Console Actions** -> **SNMP Configuration**.

Figure A-18 HMC Panel: Primary Support Element Workplace (2)

8. Two community names need to be created on this panel. The Name field is case-sensitive; use upper case.

   a. Create an entry identical to the name specified in Support Element Settings in step 6.

   > **Note:** Protocol must be UDP.
   > **Name** in this instance is **AIB**.
   > The Address value must be the Primary SE address.
   > The Network Mask value must be 255.255.255.255.
   > Access Type must be read only.

   b. Create an entry identical to the value coded in the HW Section of AOFCUST.

   > **Note:** Protocol must be UDP.
   > Name in this instance is AIBSNMP.
   > The Address value must be 127.0.0.1 .
   > The Network Mask value must be 255.255.255.255.
   > Access Type must be read/write.

Click **Add** to add the new name to Community Names. Click **OK** when both community names have been created.

Figure A-19 HMC Panel: SMHP Configuration (1)

Figure A-20 HMC Panel: SMHP Configuration (2)

9. End the Single Object Operations session by selecting **Console Actions** -> **Logoff**.

Figure A-21 HMC Panel: Primary Support Element Workplace (3)

10. Log off ACSADMIN by selecting **Console Actions** -> **Logoff**.

L

Figure A-22 HMC Panel: Hardware Management Console Workplace (3)

11. Log on as SYSPROG.

Figure A-23 HMC Panel: Hardware management Console Log on

12. Get into **Groups** -> **Defined CPCs**.

Archived

Figure A-24 HMC Panel: Hardware Management Console Workplace (4)

13. Select the CPC that you are changing and drag it to (or double-click) the Single Object Operations icon in the CPC Recovery task window.

    This takes you back into the SE for that CPC, which is necessary because this stage of the setup cannot be completed under ACSADMIN.

Archived

Figure A-25 HMC Panel: Hardware management Console Workplace (5)

Now review Figure A-26, HMC Workplace (6).

Figure A-26 HMC Panel: Hardware Management Console Workplace (6)

14. Get into **Task List** -> **CPC Operational Customization**.

Cycle the task window until the CPC Operational Customization is displayed (or select it from the Task List View).

Figure A-27 HMC Panel: Primary Support Element Workplace (4)

15. Get into **Groups** and double-click **Change LPAR Security**.

Figure A-28 HMC Panel: Primary Support Element Workplace (5)

16. Click the check boxes under **Cross Partition Authority** for those Logical Partitions on which the Internal Hardware Transport is to be enabled. Click **Save and change**.

Figure A-29 HMC Panel: Change Logical Partition Security (1)

17. Click **OK**.

Figure A-30 HMC Panel: Change Logical Partition Security (2)

18. End the Single Object Operations session by selecting **Console Actions** and double-clicking **Log off**.

Figure A-31 HMC Panel: Primary Support Element Workplace (6)

This completes the setup for a *single* CPC. This process must be repeated for every CPC that will be under the msys for Operations sphere of control. Repeat steps 1 through 18 for the remaining CPCs that require this customization.

Archived

B

Sample GRS exit ISGNQXIT

This appendix contains sample exit code that replaces sample exit ISGGREX0 (available from the ITSO since OS/390 Release 6).

This appendix contains the following:

- Sample exit code for exit ISGNQXIT

Archived

© Copyright IBM Corp. 2003

B.1 Sample exit ISGNQXIT download

This sample exit code is provided to support shared DASD with z/OS and OS/390 systems across GRSplexes. It is available from the Redbooks Web server (you must type the “SG” in upper case):

ftp://www.redbooks.ibm.com/redbooks/SG246581/

Download the file: isgnqpat.zip

Alternatively, you can go to:

http://www.redbooks.ibm.com

Select **Redbooks Online** -> **Additional Materials** -> **SG246581**.

B.1.1 Sample exit zip file

To obtain the exit, you must download ISGNQXIT.ZIP. The zip file contains:

- ISGNQPAT.ASM.TXT
- ISGNQXIT.README.TXT

The following is derived from the README file:

1) The file ISGNQPAT.ASM must be uploaded in BINARY with LRECL=80.

2) When requested, respond to support type: PATTERN.

The ISGNQXIT exit has been modified to support type PATTERN for the RNL Exclusion table scan. A customer using the exit can now use type PATTERN definitions in all RNL tables, including Conversion. The documentation in the IBM Redbook *z/OS Version 1 Release 2 Implementation*, SG24-6235 says PATTERN is not supported; with this new version, it is supported for the RNL Exclusion table scan.

For example, a customer can use the following RNL Conversion definition to convert all Reserves:

```
RNLDEF RNL(CON) TYPE(PATTERN)
 QNAME(*)                        /*  TEST FOR GRS EXIT        */
```

The entries in the RNL Conversion Table for special QNAME HWRESERV with RNAME being a volser (used by the exit to guarantee HW RESERVE for cross-sysplex volume share), can *only* be type SPECIFIC or GENERIC, because using type PATTERN for these definitions may generate confusion.

The exit, during the RNL Conversion table scan, skips the type PATTERN entries, searches for qname HWRESERV, and then compares the volser using the length of the RNAME. As mentioned, type PATTERN is not supported if you have QNAME(HWRESERV).

Examples:

```
RNLDEF RNL(CON) TYPE(SPECIFIC)
QNAME(HWRESERV)               /*volser SBOX23 is shared cross sysplex*/
RNAME(SBOX23)

RNLDEF RNL(CON) TYPE(GENERIC)
QNAME(HWRESERV)               /*volsers starting with BOOK are shared*/
RNAME(BOOK)
```

B.2 Sample exit description

It is possible to support shared DASD with z/OS and OS/390 systems across GRSplexes, as shown in Figure B-1, by using the exit code provided in “Sample exit code” on page 410, and placing it in the GRS ISGNQXIT exit.

Figure B-1 Shared DASD between sysplexes

This exit code allows all RESERVE requests for specified volumes in the RNL list to result in an HW RESERVE whenever the resource name is specified in the RNL list. By using the new ISGNQXIT exit, and adding an RNL definition to the conversion table with a QNAME not used by the system and RNAMEs that identify the volumes to share outside of GRS, the RESERVE requests for the volumes included in the RNL definitions will always result in an HW RESERVE.

Restriction: The exit, however, does not propagate cross-sysplex ENQ requests with scope=SYSTEMS. For additional information, see “Exit restrictions” on page 406.

Figure B-2 shows an RNL definition that would be used by the ISGNQXIT exit.

```
RNLDEF RNL(CON) TYPE(SPECIFIC|GENERIC)
QNAME(HWRESERV)
RNAME(VOLSER|volser-prefix)
```

Figure B-2 Sample RNL list to define sysplex DASD sharing

The same RESERVE resource requests, which address other volumes, should have the possibility to be filtered through the conversion RNLs to have the HW RESERVE eliminated.

B.2.1 GRS sample ISGNQXIT exit logic

The ISGNQXIT exit receives control for all RESERVE-ENQ-DEQ macros before the GRS RNLs processing logic. The exit traps the ENQ requests with a UCB pointer (scope SYSTEMS and HW RESERVE), and bypasses all the other ENQ-DEQ requests.

The exit locates the “VOLSER” inside the parameter list, and checks if the resource “HWRESERV” and “VOLSER” are present in the RNL conversion table. For the QNAME “HWRESERV”, only specific and generic RNL type entries are supported. For additional information, see “Exit restrictions” on page 406.

Note: If no match is found, the exit returns to GRS with no action.

Match found in RNL

The RNL exclusion list is searched to see if an RNL entry matches the request's major-minor name (specific, generic, and pattern type entries are supported). If this match is found, control is returned to GRS with no action. The ENQ request is then filtered by GRS through the RNL exclusion table, and, because it matches, the scope is changed to SYSTEM with the HW Reserve issued.

If a match is not found, the bypass RNL processing bit is set in the parameter list and control is returned to GRS. GRS RNLs processing is not done, the request scope remains SYSTEMS, and the HW RESERVE is issued. (There is some overhead due to a double serialization being used, instead of one.)

RNL exclusion table search

The reason why the exit does the RNL exclusion table search is because it is has been done since OS/360; that is, to issue a RESERVE macro (ENQ scope SYSTEMS with the UCB pointer that implies a HW RESERVE), and remove the RESERVE request with a DEQ for the same major-minor name, scope SYSTEMS with or without a UCB pointer.

B.2.2 Scanning the RNL exclusion table examples

The following scenarios describe the exit doing a no scan or scan of the RNL exclusion table and an ENQ/DEQ combination where the DEQ is issued without a UCB pointer.

No scan of RNL exclusion table

If the exit does not scan the RNL exclusion table, the following occurs:

- The RESERVE request with major-minor name present in the RNL exclusion table is trapped by the exit and maintains the scope SYSTEMS plus HW RESERVE, and GRS RNL processing is bypassed.
- The request may be ended with a DEQ without UCB pointer, major-minor name present in the RNL Exclusion table.
- The exit takes no action; control is given to GRS for RNL processing.
- GRS finds the major-minor name in the RNL exclusion table and changes the scope to SYSTEM.
- The DEQ may abend with a system code 130 because GRS knows about a scope SYSTEMS request.

In summary, a RESERVE request trapped by the exit has scope SYSTEMS plus HW Reserve, and may be ended with a DEQ scope SYSTEMS without a UCB pointer. In this scenario, without the RNL exclusion table scan, a DEQ with a major-minor name present in the RNL exclusion table would have the scope changed to SYSTEM by GRS RNL processing, and the DEQ would not remove the HW RESERVE because it is associated to a request with scope SYSTEMS. This could cause the DEQ to abend with a system code 130.

For example, RESERVEs for major name SYSIGGV2 are removed with DEQs with scope=SYSTEMS without a UCB pointer. Therefore, the RNL exclusion table scan prevents the exit from trapping RESERVE requests that will be excluded by GRS RNL processing, and therefore becoming scope=SYSTEM with a HW RESERVE.

Scan of RNL exclusion table

Following is the same scenario, with the exit scanning the RNL Exclusion table, and the same ENQ/DEQ combination where the DEQ is issued without a UCB pointer:

- A RESERVE request with major-minor name present in the RNL Exclusion table is not trapped by the exit.
- GRS RNL processing changes the scope to SYSTEM and HW RESERVE is issued.
- The request may be ended with a DEQ without a UCB pointer, major-minor name present in the RNL exclusion table.
- The exit takes no action; control is given to GRS for RNL processing.
- GRS finds the major-minor name in the RNL exclusion table and changes the request to SYSTEM
- The request is removed because GRS knows about the scope SYSTEM request.

Figure B-3 shows the exit logic with the RNL processing flow.

Figure B-3 Exit logic with the RNL processing flow

Note: Because the RNL conversion table is used, any change to the table can be activated through the following z/OS operator command:

```
SET GRSRNL=xx
```

B.2.3 RNL example

Figure B-4 shows an example of an RNL conversion table to cross-sysplex HW RESERVE volume XA9RES and volumes with prefixes of CIX.

Volumes that should not have RESERVE requests converted (always reserved) are indicated to the exit using the RNL conversion table. The parameter is an RNL definition with the special QNAME(HWRESERV), and with the RNAME indicating the volser(s). The exit supports SPECIFIC and GENERIC type entries for QNAME(HWRESERV), and a PATTERN type entry is not allowed; the RNAME is a required parameter. For additional information, see “Exit restrictions” on page 406. The special QNAMEs can be indicated in any order in the conversion table.

It is recommended that the VOLUME(S) behind the QNAME(HWRESERV) should not be dynamically changed unless the devices are offline or not allocated.

The qname HWRESERV, shown in Figure B-4, is hardcoded and is defined at label HRDWNAME in the ISGNQXIT example.

```
-----------------------------------------------------------------------------------------
Exclusion list examples
-----------------------------------------------------------------------------------------
RNLDEF RNL(EXCL) TYPE(PATTERN)
QNAME(SYSIGGV2)
RNAME(UCAT.?OS3*)                  /* ucat on HWRESERV volumes */

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSIGGV2)
RNAME(UCAT.VBOOK01)                /* ucat on HWRESERV volume */

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSZJES2)

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSCTLG)
-----------------------------------------------------------------------------------------
Conversion list examples
-----------------------------------------------------------------------------------------
RNLDEF RNL(CON) TYPE(PATTERN)
QNAME(*)                           /*Convert all Reserves */

RNLDEF RNL(CON) TYPE(SPECIFIC)
QNAME(HWRESERV)                     /*SPECIAL NAME*/
RNAME(XA9RES)                       /*ALWAYS RESERVE XA9RES*/

RNLDEF RNL(CON) TYPE(GENERIC)
QNAME(HWRESERV)                     /*SPECIAL NAME*/
RNAME(CIX)                          /*ALWAYS RESERVE VOLUMES*/
                                    /*BEGINNING WITH CIX   */
```

Figure B-4 RESERVE conversion RNLexample

B.2.4 Exit restrictions

The sample ISGNQXIT exit has some restrictions that should be understood and evaluated very carefully, as follows:

- The exit does not propagate cross-sysplex global ENQs (scope=SYSTEMS); it only guarantees that the HW RESERVEs are issued for volumes behind qname HWRESERV in the RNL conversion table for all RESERVE requests. Applications that logically serialize a DASD resource (PDS member or data set) with a global ENQ (scope=SYSTEMS) cannot depend on the exit to serialize resources on DASD shared between sysplexes.

  For example, ISPF serializes the edit of a PDS member with an ENQ scope=SYSTEMS. If a user is editing a PDS member, another user in the same GRSplex is notified that the member is in use if he tries to edit the same member. If a second user is in another GRSplex, he can edit the member. ISPF uses a RESERVE macro to protect only the write of the updated member.

  In this scenario, the first member's update will be overridden by the second member's update. The same situation should be expected with the Linkage-Editor that uses a RESERVE macro to protect only the write of the load module to disk.

- Type PATTERN is not supported for the special QNAME HWRESERV. The reason to restrict the RNL entry type to generic and specific is to allow the use of the RNLDEF RNL(CON) TYPE(PATTERN) QNAME (*) to convert all Reserves.

  Using the previous definition, and allowing the special QNAME HWRESERV to be type PATTERN, would have resulted in an always match for an all QNAME HWRESERV search, and as a consequence having all Reserve requests not converted.

- It is possible to dynamically add and remove volumes behind QNAME(HWRESERV) with the **SET GRSRNL=xx** z/OS command, but the system does not check if the volumes have outstanding RESERVE requests before activating the new RNLs. It is recommended that the volume(s) specified behind the QNAME(HWRESERV) should not be dynamically changed unless the device is off-line or not allocated.

B.2.5 Recommendations

When a reserve request is intercepted by the exit, the request scope remains SYSTEMS and the HW RESERVE is issued: a double serialization. A scenario where the exit controls volumes containing user catalogs and the systems in a sysplex are frequently accessing and updating the catalogs, will result in continuously cross-obtaining RESERVEs with the probability that the systems may eventually deadlock. Therefore, for volumes shared cross-sysplex and containing user catalogs, it is recommended to have the RNL exclusion list entries for catalog QNAME SYSIGGV2 and RNAME ucat-name, and specify either type GENERIC, SPECIFIC, or PATTERN. For an example, see the exclusion list example in Figure B-2 on page 403.

Use the GRS option SYNCHRES(YES) (synchronous reserve). It can be activated through either the GRSCNFxx parmlib member or the **SETGRS** command. The command has system scope. The SYNCHRES option allows an installation to specify whether the system should obtain a hardware RESERVE for a device prior to granting a global resource serialization ENQ. This option might protect jobs that have a delay between a hardware RESERVE request being issued and the first I/O operation to the device. Prior to the implementation of the SYNCHRES option, the opportunity for a deadlock situation was more likely to occur.

B.2.6 Exit compatibility

To support sysplex DASD sharing in configurations where the sysplexes may have GRS with and without wildcard support, the ISGNQXIT dynamic exit is compatible with the ISGGREX1 ITSO exit provided for MVS and OS/390 systems without GRS wildcard support. Both exits have the same scope and functionality. The ISGNQXIT exit is a replacement for the ISGGREX1 exit beginning with z/OS Version 1 Release 2.

B.2.7 Exit installation and activation

The exit can be activated by one of the following methods:

- With a program using the CSVDYNEX macro.
- With the **SETPROG EXIT** z/OS operator command—and assuming that the exit has been linked in SYS1.USER.LINKLIB with the name ISGNQPAT—issue the command as follows:

  ```
  SETPROG EXIT,ADD,EX=ISGNQXIT,MOD=ISGNQPAT,DSN=SYS1.USER.LINKLIB
  ```

- Using an EXIT STATEMENT in the PROGXX parmlib member, specify the member as follows:

  ```
  EXIT ADD
  EXITNAME(ISGNQXIT)
  MODNAME(ISGNQPAT)
  STATE=ACTIVE
  DSNAME(SYS1.USER.LINKLIB)
  ```

Following is the recommended procedure to activate the exit after an IPL:

- Activate the exit in all systems sharing DASD across the sysplexes.
- Update the RNL conversion table in the GRSRNLxx parmlib member.
- Activate the RNLs in all systems sharing DASD across the sysplexes.

For additional information, refer to *z/OS MVS Installation Exits*, SA22-7593.

B.2.8 EXIT process verification

To verify if the exit is behaving as expected, you can use the ENQ/RESERVE/DEQ Monitor. Monitor selection 1, MAJOR Names Display, has been extended with a new column to indicate if the request has been modified by exit ISGNQXIT.

Monitor selection 1

Figure B-5 shows a MAJOR Names display example.

Archived

```
                        ENQ/DEQ Monitor - Major Name List      Row 1 to 20 of 29

 Enter S to select a Major Name for details     .
       L major on command line to locate a Major.   Elapsed seconds:      99

    Sel.   ----------   -----   ----   -----   -------  -Average-   -Reserved-
    Field  Major Name   Scope   Exit   RNL     Counter     msec        seconds
    _      SYSZJES2    *RES                        101       21            2
    _      SYSZVVDS    *RES    YES                   2        2            0
    _      SYSVTOC     *RES    YES                   2       69            0
    _      SYSIGGV2    *RES                          2       14            0
    _      SPFEDIT      SYSS                         9
    _      IGDCDSXS     SYSS                         7
    _      CHANGEQU     SYSS                        10
    _      AUDITCOD     SYSS           NO           26
    _                   SYSS           NO           60
    _      SYSZVVDS     SYSS                       168
    _      SYSZRACF     SYSS                         8
    _      SYSZMCS      SYSS           NO            7
    _      SYSZIOS      SYSS           NO           11
    _      SYSZENQM     SYSS           NO            1
    _      SYSZDSCB     SYSS                         2
    _      SYSZATR      SYSS           NO           30
    _      SYSVSAM      SYSS                        29
    _      SYSIGGV2     SYSS                        96
    _      SYSDSN      *SYSS                        11
    _      SIBIXFP      SYS                          1
```

Figure B-5 ENQ/DEQ Monitor selection 1

Figure B-5 shows that RESERVE requests for Major names SYSZVVDS and SYSVTOC have been processed by exit ISGNQXIT, and that the HW RESERVEs have been issued.

Monitor selection 3

Using Monitor selection 3 and the VOLUME list display, you can see the volumes where HW Reserves have been issued. Figure B-6 shows two volumes:

- BOOK01 shared across two sysplexes and therefore with a HWRESERV entry in the RNL conversion table
- SBOX23 with JES2 checkpoint that has its RESERVE major name in the RNL Exclusion table

```
                        ENQ/DEQ Monitor - VOLUME List          Row 1 to 2 of 2

 Enter S to  select a Volume for details
       A for active Reserves on  Volume
       L volume on command line to locate a Volume
   *   indicates volume where reserves are not converted
-- - ------ ------- Dev. Max  --------- ------------ Reserve Time -------------
S.   Volume Tot.Res nbr   Res Elap(sec)  Avg.(ms)  Min.(ms)  Max.(ms) Tot.(sec)
 _ * BOOK01      6  2601   04       192         0         0         0         0
 _ * SBOX23    247  2558   01       248         0        17        69         0
 ****************************** Bottom of data ******************************
```

Figure B-6 Volume List display with monitor selection 3

Selecting the entry for volume BOOK01, the major-minor name combinations of the RESERVE macros are displayed, as shown in Figure B-7.

```
                      ENQ/DEQ Monitor - VOLUME Entry List      Row 1 to 3 of 3

 Volser. . . . . . : BOOK01      Average Reserve Time (ms) : 0
 Tot.nr of Reserve : 6           Minimum Reserve Time (ms) : 0
 Dev.nr. . . . . . : 2601        Maximum Reserve Time (ms) : 0
 Max Reserve Cnt. .: 04          Total   Reserve Time (sec): 0
 Elapsed Time (sec): 192         Volume  Reserve Rate (min): 2

 Interval
- Rate -- ----- -------- ----------------------  ------------ Time ------------
S  min.   Count MajName  Minor name (max 22 ch)  Avg ms  Min ms  Max ms Tot sec
_     0       2 SYSIGGV2 UCAT.VBOOK01                14       7      22       0
_     0       2 SYSVTOC  BOOK01                      68      58      79       0
_     0       2 SYSZVVDS BOOK01                       2       2       2       0
****************************** Bottom of data ******************************
```

Figure B-7 BOOK01 detail display

For additional information about the GRS monitor, refer to *z/OS MVS Planning: Global Resource Serialization*, SA22-7600.

B.2.9 Sample exit code

Example B-1 on page 410 is the sample exit code to place into GRS exit ISGNQXIT.

Example: B-1 Sample exit code to place into GRS exit ISGNQXIT

```
//MERONI   JOB (999,POK),CLASS=A,                                  00010000
//         MSGLEVEL=(1,1),MSGCLASS=X,                              00020000
//         NOTIFY=&SYSUID                                          00030000
//ASMH   EXEC PGM=ASMA90,REGION=1024K,                             00040020
//            PARM='DECK,XREF(SHORT),SYSPARM(V1R1M1)'              00040120
//SYSPRINT  DD SYSOUT=*                                            00050000
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,5)),DISP=(NEW,DELETE)      00060000
//SYSPUNCH  DD DSN=&OBJ(AUDIT),DISP=(,PASS,DELETE),UNIT=SYSDA,     00070000
//             SPACE=(TRK,(1,5,5))                                 00080000
//SYSPUNCH  DD DUMMY                                               00090000
//SYSGO     DD DUMMY                                               00100000
//SYSLIN    DD DUMMY                                               00110000
//SYSLIB    DD DSN=SYS1.MACLIB,DISP=SHR                            00120000
//          DD DSN=SYS1.MODGEN,DISP=SHR                            00130000
//SYSIN DD *                                                       00140000
         TITLE 'ISGNQXIT - ENQ/DEQ EXIT ROUTINE'                   00150000
ISGNQXIT CSECT                                                     00160000
*/*  START OF SPECIFICATIONS ****                                  00170000
*-------------------------------------------------------------------*  00180000
* Cross Sysplex DASD sharing                                       *  00190000
* Always H/W RESERVE Volumes shared cross GRS complexes            *  00200000
*-------------------------------------------------------------------*  00210000
*                                                                  *  00220000
* It is mandatory to support SHARED DASD with OS/390 Systems       *  00230000
* cross GRS-plexes that all RESERVE requests that address these    *  00240000
* VOLUMES result in an H/W RESERVE whatever the RESOURCE NAME is.  *  00250000
*                                                                  *  00260000
* The same RESOURCES, that address other VOLUMES, should have the  *  00270000
* possibility to be filtered through the CONVERSION RNLs and have  *  00280000
```

```
* the H/W RESERVE eliminated.                                        *     00290000
*                                                                    *     00300000
* Using ISGNQXIT GRS exit and by adding to the CONVERSION table      *     00310000
* a QNAME not used by the System and RNAMEs that identify the        *     00320000
* Volumes to share outside GRS, the RESERVE requests for the         *     00330000
* Volumes included in the following definition will always result *     00340000
* in a H/W RESERVE.                                                  *     00350000
*                                                                    *     00360000
*      RNLDEF RNL(CON) TYPE(SPECIFIC/generic)                        *     00370000
*      QNAME(HWRESERV)                                               *     00380000
*      RNAME(VOLSER/volser-prefix)                                   *     00390000
*                                                                    *     00400000
* LOGIC:                                                             *     00410000
*                                                                    *     00420000
* ISGNQXIT  receives control for all RESERVE/DEQ  requests           *     00430000
* (SYSTEMS+H/W RESERVE), locates the 'VOLSER'  and checks if the     *     00440000
* resource 'HWRESERV' 'VOLSER' is present in the conversion          *     00450000
* table. If found the RNL processing is bypassed, if not normal      *     00460000
* RNL scan is performed.                                            *     00470000
*                                                                    *     00480000
* Because the RNL CONVERSION table is used, any change to the        *     00490000
* table can be activated through OS/390 operator command:            *     00500000
* SET GRSRNL=xx.                                                     *     00510000
*                                                                    *     00520000
* The name 'HWRESERV' is hard-coded and is defines at label          *     00530000
* HRDWNAME in the ISGNQXIT example.                                  *     00540000
*                                                                    *     00550000
* NOTE: the VOLUME(S) behind the QNAME(HWRESERV) should not be       *     00560000
* dynamically changed unless the device(s) is(are) offline.          *     00570000
*                                                                    *     00580000
* TYPE PATTERN IS NOT SUPPORTED FOR HWRESERV RNL ENTRIES.            *     00590016
*                                                                    *     00600000
*--------------------------------------------------------------------*     00610000
*                CONVERSION LIST EXAMPLE                             *     00620000
*--------------------------------------------------------------------*     00630000
*                                                                    *     00640000
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *     00650000
* QNAME(SYSVTOC)                                                     *     00660000
*                                                                    *     00670000
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *     00680000
* QNAME(SYSIGGV2)                                                    *     00690000
*                                                                    *     00700000
* RNLDEF RNL(CON) TYPE(SPECIFIC)                                     *     00710000
* QNAME(HWRESERV)                    /*SPECIAL NAME*/              *     00720000
* RNAME(XA9RES)                      /*ALWAYS RESERV  XA9RES*/     *     00730000
*                                                                    *     00740000
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *     00750000
* QNAME(HWRESERV)                    /*SPECIAL NAME*/              *     00760000
* RNAME(CIX)                         /*ALWAYS RESERV VOLUMES*/     *     00770000
*                                    /*BEGINNING WITH CIX   */     *     00780000
*                                                                    *     00790000
***                        END OF                                   *     00800000
***   RESERVE CONVERSION RESOURCE NAME LIST - SAMPLE                 *     00810000
**********************************************************************     00820000
* RESTRICTION:                                                       *     00830000
*                                                                    *     00840000
* 1-The exit does not propagate cross Sysplex ENQ requests with      *     00850000
* scope=SYSTEMS, only guarantees that the H/W Reserve is issued      *     00860000
* for Volumes behind qname HWRESERV in RNL Conversion table.         *     00870000
*                                                                    *     00880000
```

```
* 2-Type PATTERN is not supported for HWRESERV RNL entries        *     00890000
*                                                                  *     00900000
*Note-Restriction of type PATTERN not supported for EXCLUSION RNL *     00901000
*     HAS BEEN REMOVED                                             *     00901100
*                                                                  *     00901200
*******************************************************************     00901300
* COMPATIBILITY:                                                   *     00901400
*                                                                  *     00901500
* ISGNQXIT is compatible with ISGGREX1 ITSO exit provided for      *     00901600
* OS/390 Systems without GRS WILDCARD support.                     *     00901700
*                                                                  *     00901800
*******************************************************************     00901900
*                                                                  *     00902000
                                                                          00903000
*******************************************************************     00904000
**EXIT INSTALLATION                                                *     00905000
*                                                                  *     00906000
* THE EXIT CAN BE INSTALLED WITH ONE OF THE FOLLOWING METHODS      *     00907000
*                                                                  *     00908000
* 1.WITH A PROGRAM USING CSVDYNEX MACRO                            *     00909000
*                                                                  *     00910000
* 2.WITH SETPROG EXIT OPERATOR COMMAND                             *     00920000
*   EX. SETPROG EXIT,ADD,EX=ISGNQXIT,MOD=ISGNQXIT,                 *     00930000
*       DSN=SYS1.USER.LINKLIB                                      *     00940000
*                                                                  *     00950000
* 3.USING EXIT STATEMENT OF THE PROGXX PARMLIB MEMBER              *     00960000
*       EXIT ADD                                                   *     00970000
*            EXITNAME(ISGNQXIT)                                    *     00980000
*            MODNAME(ISGNQXIT)                                     *     00990000
*            STATE=ACTIVE                                          *     01000000
*            DSNAME(SYS1.USER.LINKLIB)                             *     01010000
*                                                                  *     01020000
*                                                                  *     01030000
*NOTE: Recommended procedure to activate the exit after IPL:       *     01040000
*      1.Activate the exit in all Systems sharing ALWAYS RESERVE   *     01050000
*        VOLUMES                                                   *     01060000
*      2.Activate the RNLS in all Systems sharing ALWAYS RESERVE   *     01070000
*        VOLUMES. The Volume(s) behind the QNAME(HWRESERV) should *     01080000
*        not be dynamically added/removed unless the devices are   *     01090000
*        OFFLINE                                                   *     01100000
*******************************************************************     01110000
       EJECT                                                              01120000
*******************************************************************     01130000
*                                                                          01140000
* INPUT R1=NQXP, R13=STANDARD SAVE AREA, R14=RETURN ADDRESS                01141012
*       R15=ENTRY POINT                                                    01142012
*                                                                          01143012
*                                                                          01144013
*       REGISTERS-SAVED =   R0 - R12, R14, R15                             01150012
*                                                                          01300000
*       REGISTERS-RESTORED = R0 - R12, R14                                 01310012
*                                                                          01330000
*       RETURN-CODES =   R15 = 0 -                                         01350012
*                              4 - NOT USED                                01360000
*                                                                          01370000
*       EXIT-ERROR = NONE                                                  01380000
*                                                                          01390000
*       WAIT-STATE-CODES = NONE                                            01400000
*                                                                          01410000
*       EXTERNAL-REFERENCES = IEANTCR - IEANTRT (NAME/TOKEN)               01420012
```

```
*                                                                       01430000
*       ROUTINES = ISGGRHSO  (GRS RNL SEARCH)                           01440012
*                                                                       01450000
*       DATA-AREAS = WORKAREA                                           01460012
*                                                                       01470000
*       CONTROL-BLOCKS = CVT       R                                    01490012
*                        NQXP      R/W                                  01500012
*                        GVT       R                                    01510000
*                                                                       01520000
*       TABLES = 1. RESERVE CONVERSION RNL        (ISGGCRNL)            01540012
*                2. EXCLUSION RNL                 (ISGGERNL)            01541012
*                                                                       01550000
*       SERIALIZATION = LOCAL, CMSEQDQ LOCKS                            01560012
*                                                                       01570000
*                                                                       01571012
*01*  CHANGE-ACTIVITY =  INITIAL RELEASE 1.0                   08/01   01580000
*                                                                       01590000
*02*  RELEASE 1.1.1                                           06/02    01650008
*     Type PATTERN support for RNL EXCL table search                    01660015
*                                                                       01670000
*     MESSAGES = NONE.                                                  01680000
*                                                                       01690000
*     ABEND-CODES = NONE.                                               01700000
*                                                                       01710000
**** END OF SPECIFICATIONS **                                        */ 01720000
         EJECT                                                          01730000
************************************************************************ 01740000
*                                                                     * 01750000
*        REGISTER ASSIGNMENTS                                         * 01760000
*                                                                     * 01770000
************************************************************************ 01780000
*                                                                       01790015
RNLEPTR  EQU   2                    POINTER TO AN RNL ENTRY (RNLE)     01820000
FLENRNLE EQU   4                    LENGTH OF FIXED PART OF AN RNLE    01830000
RNAMELEN EQU   5                    LENGTH OF RNAME                    01840008
BASEREG  EQU   11                   BASE REGISTER                      01850000
WORKREG  EQU   12                   WORK REGISTER                      01860000
*                                                                       01900000
R0       EQU   0                                                        01910000
R1       EQU   1                                                        01920008
R2       EQU   2                    POINTER TO AN RNLE                 01930000
R3       EQU   3                    CVT, GVT                           01940000
R4       EQU   4                    Length of RNLE fixed part          01950008
R5       EQU   5                    Length of RNAME                    01960000
R6       EQU   6                    RETURN REG, RNL SEARCH             01970000
R7       EQU   7                    RNL list to search I=1,E=2,C=3     01980000
R8       EQU   8                                                       01990000
R9       EQU   9                    NQPX POINTER                       02000012
R10      EQU   10                                                      02010000
R11      EQU   11                   BASE REG                           02020000
R12      EQU   12                   WORK REG                           02030000
R13      EQU   13                   SAVE AREA POINTER                  02040000
R14      EQU   14                   RETURN ADDRESS                     02050000
R15      EQU   15                                                      02060000
       EJECT                                                            02070000
************************************************************************ 02080000
*                                                                     * 02090000
*        LOGIC FLOW FOR ISGNQXIT                                      * 02100000
*                                                                     * 02110000
************************************************************************ 02120000
```

```
*/*                                                                  */ 02130000
*/*++ 'ISGNQXIT': ENTRY TO ENQ DEQ EXIT                               */ 02140000
*/*++ ESTABLISH ADDRESSABILITY                                        */ 02150000
*/*++ IF DEQ                            --> RETURN                    */ 02160000
*/*++ IF THE NQXP  UCB POINTER NOT ZERO --> RETURN                    */ 02170012
*/*                                                                  */ 02180000
*/*++   LOCATE RNL CONVERSION TABLE POINTER                            */ 02190000
*/*++     DO WHILE MATCH NOT FOUND AND NOT LAST ENTRY IN THE RNL      */ 02200000
*/*++       IF RNL MAJOR NAME IS HWRESERV                             */ 02210000
*/*++         IF RNLE MINOR NAME EQUALS NQXP VOLSER                   */ 02220015
*/*++           GOTO SCAN_RNL_EXCLUSION_TABLE                         */ 02230000
*/*++             IF RNLE FOUND ---->  RETURN                         */ 02230115
*/*++             IF RNLE NOT_FOUND                                   */ 02230215
*/*++                SET BYPASS_RNL_PROCESSING                        */ 02231015
*/*++                RETURN                                           */ 02232015
*/*++             ENDIF                                               */ 02240015
*/*++         ENDIF    (END OF RNAME COMPARISON)                      */ 02241015
*/*++       ENDIF  (END OF CHECK FOR RNL MAJOR NAME HWRESERV          */ 02250000
*/*++     ENDDO  (REPEAT SEQUENCE UNTIL MATCH FOUND OR END OF RNL)    */ 02260000
*/*++    ENDIF  (END OF ENXP  PROCESSING)                             */ 02270000
*/*++ RETURN                                                          */ 02280000
*/*                                                                  */ 02290000
*/*  :SCAN_RNL_EXCLUSION_TABLE                                        */ 02300015
*/*++   OBTAIN LOCAL + CMSNQDQ LOCKS                                  */ 02310015
*/*++     FIRST PASS THROUGH                                          */ 02320015
*/*++       OBTAIN WORK AREA IN ESQA                                  */ 02321015
*/*++       CREATE NAME/TOKEN PAIR WITH POINTER TO WORK AREA          */ 02322015
*/*++     NOT FIRST PASS THROUGH                                      */ 02330015
*/*++       RETRIEVE NAME/TOKEN PAIR & POINTER TO WORK AREA           */ 02331015
*/*++     INITIALIZE WORK AREA WITH Q/RNAME & RNLE TO SEARCH          */ 02340015
*/*++     CALL ISGGRHSO (GRS RNLE SEARCH ROUTINE)                     */ 02341015
*/*++   RELEASE LOCAL + CMSNQDQ LOCKS                                 */ 02341115
*/*++   RETURN OFFSET 0=NO_MATCH  4=MATCH                             */ 02342015
*/*  :END_OF_EXCLUSION_RNL SEARCH                                     */ 02350015
*/*++ END 'ISGNQXIT'                                                  */ 02380000
*/*                                                                  */ 02390000
*/*******************************************************************/ 02400000
         SPACE 3                                                      02410000
ISGNQXIT AMODE 31                                                     02420000
ISGNQXIT RMODE ANY                                                    02430000
         MODID BR=NO,MODLBL=ITSOQXIT                                  02440017
ISGCVXIT DS    OH                                                     02450000
         ENTRY ISGCVXIT                                               02480000
         SPACE                                                        02490000
         STM   14,12,12(13)             SAVE ENTRY REGS               02500000
         LR    BASEREG,R15                                            02510000
         USING ISGCVXIT,BASEREG                                       02520000
         LR    R9,R1                                                  02521008
         USING NQXP,R9                  ENQ PLIST ADDRESSABILITY      02530008
*                                                                     02540000
         TM    NQXP_STATEFLAGS1,NQXP_SF1_ENQ  RETURN IF DEQ           02560008
         BZ    MODEXIT                                                02570008
*                                                                     02590000
         ICM   R6,15,NQXP_OP_UCB       UCB POITER                    02600000
         BZ    MODEXIT                  NO RESERVE, EXIT              02610000
*                                       LOCATE RNL CONV               02620000
         L     R3,FLCCVT-PSA            CVT                           02630000
         USING CVT,R3                                                 02640000
         L     R3,CVTGVT                GET GRS VECTOR                02650000
         DROP  3                                                      02660000
```

```
         USING GVT,R3                                                   02670000
         TM    GVTVFLAG,GVTVCRNL        RESERVE CONVERSION INVALID      02680000
         BO    MODEXIT                  YES, EXIT                       02690000
*                                                                       02700000
         TM    GVTVFLAG,GVTVERNL        SYSTEMS EXCLUSION  INVALID      02710000
         BO    MODEXIT                  YES, EXIT                       02720000
*                                                                       02730000
         USING RNLE,RNLEPTR                                             02740000
         ICM   RNLEPTR,15,GVTRCRNL      RESERVE CONVERTION RNL          02750000
         BZ    MODEXIT                  NO RNL CONVERSION               02760000
         L     R14,COMP1                RETURN ADDRESS                  02770000
*                                                                       02780000
*-----------------------------------------------------------------*     02790000
* NQXP_RD_VOLSER = VOLSER OF REQUESTED UCB                              02800000
*-----------------------------------------------------------------*     02810000
*                                                                       02820012
COMPRNL1 EQU   *                                                        02830009
         TM    RNLEFLGS,RNLELAST       AT END OF THE RNL ?              02840000
         BO    MODEXIT                 YES => RNL SCAN DONE, NOMATCH    02850009
*                                      RETURN                           02860000
* COMPARE QNAME HWRESERV   WITH RNLE                                    02880000
*                                                                       02890000
         TM    RNLEFLGS,RNLEPATT       PATTERN TYPE ENTRY NOT           02900000
         BO    NEXTRNL1                SUPPORTED                        02910000
*                                                                       02920000
         CLC   HRDWNAME,RNLEQNME       CHECK IF QNAME HWRESERV          02930000
         BNE   NEXTRNL1                NO => GET NEXT RNL ENTRY         02940000
*                                                                       02950011
* COMPARE THE VOLSER, MUST BE SPECIFIED                                 02960011
*                                                                       02961011
         SLR   RNAMELEN,RNAMELEN       CLEAR WORK REG                   02970000
         IC    RNAMELEN,RNLERNML       GET LENGTH OF RNAME IN THE RNLE  02980000
*                                                                       02990000
         LTR   RNAMELEN,RNAMELEN       MINOR LENGTH CANNOT BE ZERO,     03000000
         BZ    NEXTRNL1                VOLSER MUST BE INDICATED         03010000
*                                                                       03020000
         BCTR  RNAMELEN,0              ADJUST LENGTH                    03030000
         EX    RNAMELEN,COMPRNM1       COMPARE THE RNAME (VOLSER)       03040000
         BNE   NEXTRNL1                UNEQUAL => GET NEXT RNLE         03050000
*                                      VOLUME SERIAL FOUND IN THE RNL   03070000
*                                      CHK IF RESOURSE IS IN RNL EXCL   03080000
*-----------------------------------------------------------------*     03090000
* EXCLUSION RNL LIST SEARCH                                             03100000
*-----------------------------------------------------------------*     03110000
         ICM   RNLEPTR,15,GVTSERNL     SYSTEMS EXCLUSION  RNL           03120000
         BZ    BYP_RNL                 NO RNL EXCLUSION,                03130000
         LA    R7,2                    search Exclusion RNL             03131000
*                                      SET BYPASS RNL PROCESSING        03140000
         BAS   R6,RNL_SRCH             RNL serach routine               03150000
*                                                                       03160000
         B     BYP_RNL     OFFSET --> 0   NO MATCH, set bypass RNL     03162000
         B     MODEXIT     OFFSET --> 4   MATCH, EXIT                  03162111
*                                                                       03163000
BYP_RNL  EQU   *                                                        03220014
*-----------------------------------------------------------------     03221015
*  Support NQXP_FLAGS1 name change                                      03221115
*-----------------------------------------------------------------     03221215
         AIF   ('&SYSTEM_ID'(1,1) EQ '0').OS390                         03222012
*                                 'OS/390 02.XX.00'                     03223212
         AIF   ('&SYSTEM_ID' EQ 'z/OS 01.01.00').OS390                  03223412
```

```
*                                       SET BYPASS RLN PROCESSING        03226012
         OI    NQXP_REQUESTFLAGS1,NQXP_RF1_BYPASSRNLS  Z/OS 1.2 AND UP  03230012
         AGO   .ZOS                                                     03230112
.OS390   OI    NQXP_FLAGS1,NQXP_RF1_BYPASSRNLS    OS/390 AND Z/OS 1.1   03233112
.ZOS     ANOP                                                          03234012
*-----------------------------------------------------------------     03235015
         B     MODEXIT                                                 03240000
*                                                                      03242000
COMPRNM1 CLC   RNLERNME(0),NQXP_RD_VOLSER       COMPARE VOLSER         03243000
*OMPRNM1 CLC   RNLERNME(0),UCBVOLI-UCBOB(R6)    COMPARE VOLSER          03244000
*-----------------------------------------------------------------     03570800
* R13 SAVE AREA, R14, RETURN ADDRESS, R15 RET-CODE                     03570900
*-----------------------------------------------------------------     03571000
MODEXIT1 EQU   *                                                       03571209
         LA    R15,4                   *TEST                           03571309
         B     CC_EXT                                                  03571409
MODEXIT  EQU   *                                                       03571509
         SR    R15,R15                 RETURN CODE 0                   03571609
CC_EXT   EQU   *                                                       03571709
         L     R14,12(0,R13)           RECOVER THE RETURN ADDRESS      03571809
         LM    R0,R12,20(R13)          RECOVER OTHERS EXCEPT R15       03571909
         BSM   0,R14                   RETURN TO THE CALLER            03572009
*                                                                      03572109
*-----------------------------------------------------------------     03572900
* GET THE ADDRESS OF THE NEXT RNL ENTRY                                03573000
*-----------------------------------------------------------------     03573100
NEXTRNL1 EQU   *                                                       03573200
         LA    FLENRNLE,RNLERNME-RNLE  LENGTH OF FIXED PART OF RNLE    03573300
         SLR   WORKREG,WORKREG         CLEAR WORK REG                  03573400
         IC    WORKREG,RNLERNML        GET RNAME LENGTH (VARIABLE)     03573500
         ALR   WORKREG,FLENRNLE        ADD FIXED + VARIABLE LENGTHS    03573600
         ALR   RNLEPTR,WORKREG         GET ADDRESS OF NEXT RNL ENTRY   03573700
         BSM   0,R14                   CHECK THE NEW RNL ENTRY         03573800
         DROP  RNLEPTR                                                 03573900
*-----------------------------------------------------------------*    03574000
* END  OF THE NEXT RNL ENTRY                                           03574100
*-----------------------------------------------------------------*    03574200
*                                                                      03574300
*-----------------------------------------------------------------     03574400
*                                                                      03574500
*     SUBROUTINES                                                      03574600
*                                                                      03574900
RNL_SRCH EQU   *                                                       03575100
*-----------------------------------------------------------------     03576000
* RNL search routine                                                   03580000
*                                                                      03590000
* Input R9=NQXP , R3=GVT,  R6=return                                   03591012
*       R7=RNL list to search: Input=1, Exclusion=2, Conversion=3      03591100
*                                                                      03591200
* Output R6+0=RNLE not found  R6+4=RNLE found                          03591307
*                                                                      03591407
*   obtain LOCAL + CMSEQDQ locks                                       03591500
*   retrieve name token pair for ISGNQXIT -IEANTRT-                    03591607
*      if does not exists                                              03591700
*            STORAGE OBTAIN 352 byte work area in ESQA                 03591800
*            compare and swap  pointer in name token                   03591900
*              if work area pointer is already present                 03592007
*              STORAGE RELEASE, iterate (back to retrieve name token)  03592107
*            release CMSEQDQ + LOCAL locks                             03592200
*            create name token pair -IEANTCR-                          03592300
```

```
*            iterate (back to main entry- obtain locks)                 03592400
*      if exists                                                        03592500
*            get pointer to work area, init with qname+rname            03592600
*            call rnl search routine GVTGRHSO                           03592700
*             R0=2 Exclusion RNL search  (1=Inclusion, 3=Conversion)    03592800
*             R1=pointer to work area, R2=rname length,                 03592900
*             R3=value in GVTRSE                                        03593000
*------------------------------------------------------------------------ 03593100
*                                                                        03593200
* SETLOCK USES - R0 -1 -14 -15 when keyword REGS=USE is specified       03593300
*                                                                        03593400
*------------------------------------------------------------------------ 03593500
ITERATE  EQU   *                                                        03593600
*                                                                        03593900
         SETLOCK OBTAIN,TYPE=LOCAL,REGS=USE,RELATED=ISGNQXIT,          X03594000
               MODE=UNCOND                                              03594100
*                                                                        03594200
         SETLOCK OBTAIN,TYPE=CMSEQDQ,REGS=USE,RELATED=ISGNQXIT,        X03594309
               MODE=UNCOND                                              03594409
*                                                                        03594500
*------------------------------------------------------------------------ 03594800
*                                                                        03594900
*        RETRIEVE name/token pair, LOCAL and CMS locks can be held      03595000
*                                                                        03595100
*        IEANTRT USES - R0 -1 -14 as work, R15=rc                       03595200
*                       RC=0 ok, TOKEN points to w.a., RC=4 not found   03595300
*------------------------------------------------------------------------ 03595400
RETRY    EQU   *                                                        03595500
*                                                                        03595600
         CALL  IEANTRT,(LEVEL,NAME,TOKEN,RETCODE)                       03595700
*                                                                        03596000
         CLC   RETCODE(4),ZERO                                          03596100
         BNE   NOT_FND              NAME/TOKEN NOT YET CREATED          03596200
         B     NAME_FND                                                 03596300
*                                                                        03597000
*------------------------------------------------------------------------ 03597200
NOT_FND  EQU   *                                                        03597300
*------------------------------------------------------------------------ 03598900
* 16 byte save regs 1-2-3-13, 72 byte standard save area for search rtn 03599000
*  8 byte qname, 256 byte rname                                          03599100
*------------------------------------------------------------------------ 03599200
*                                                                        03599300
         STORAGE OBTAIN,LENGTH=352,SP=245,LOC=31,ADDR=(1)               03599400
*                                                                        03599500
         SR    R14,R14              CS to serialize multi concurrent    03599600
         CS    R14,R1,TOKEN         entries to the exit, WA in TOKEN    03599700
         BZ    CRE_NT                                                   03599900
*                                                                        03600000
         STORAGE RELEASE,LENGTH=352,SP=245,ADDR=(1)                     03600100
*                                                                        03600200
         B     RETRY                                                    03600300
*                                                                        03600400
*------------------------------------------------------------------------ 03600500
*                                                                        03600600
* SETLOCK USES - R0 -1 -14 -15 when keyword REGS=USE is specified       03600700
*                                                                        03600800
*------------------------------------------------------------------------ 03600900
CRE_NT   EQU   *                                                        03601200
*                                                                        03601300
         SETLOCK RELEASE,TYPE=CMSEQDQ,REGS=USE,RELATED=ISGNQXIT         03601609
```

```
*                                                                       03601700
         SETLOCK RELEASE,TYPE=LOCAL,REGS=USE,RELATED=ISGNQXIT           03601800
*                                                                       03601900
*---------------------------------------------------------------------- 03602000
*                                                                       03602100
*        CREATE   name/token pair, no locks can be held                 03602200
*                                                                       03602300
*        IEANTCR USES - R0 -1 -14 as work, R15=rc                       03602400
*                       RC=0 ok, TOKEN created, RC=4 alredy exists      03602500
*----------------------------------------------------------------------* 03602600
*                                                                       03602700
         CALL  IEANTCR,(LEVEL,NAME,TOKEN,PERSIST,RETCODE)               03602800
*                                                                       03602900
         B     ITERATE                                                  03603015
*-TEST   CLC   RETCODE(4),FOUR       RC=0 CREATED, RC=4                 03603515
*-TEST   BNH   ITERATE               NAME/TOKEN ALREADY EXISTS          03603615
*-TEST   B     ERROR                 *test  should not occur            03603815
*                                                                       03604000
*---------------------------------------------------------------------- 03604100
*        R9=NQPL  R3=GVT                                                03604314
*        R6=RETURN ADDRESS  +0 not_found   +4 found                     03604400
*        R7=RNL list to search                                          03604500
*----------------------------------------------------------------------* 03604600
NAME_FND EQU   *                                                        03604700
         L     R12,TOKEN                 GET WORK AREA                  03605000
         USING WORKAREA,R12                                             03605100
         STM   R1,R3,SAVE_REG            SAVE R1-R3                     03605200
         ST    R13,SAVE_R13              SAVE R13                       03605300
         LA    R13,SAVEAREA              STANDARD SAVE AREA             03605400
         MVC   QNAME(8),NQXP_OP_QNAME                                   03605500
         XC    RNAME(256),RNAME          CLEAR RNAME IN WORK AREA       03605606
         ICM   R10,15,NQXP_OP_RNAME      RNAME POINTER                 03605700
         SR    R2,R2                                                    03605800
         IC    R2,NQXP_OP_RNAMELEN       RNAME LEN                      03605900
         BCTR  R2,0                       -1                            03606000
         EX    R2,MOVE_RNM               MOVE RNAME                     03606100
         LA    R2,1(0,R2)                +1 RNAME length R2             03606218
         ST    R2,SAVEAREA               SAVE RNAME length FOR DIAG     03606318
         LA    R1,OFFREG(0,R12)          R1 pointer to qname-rname      03606400
         LR    R0,R7                     Exclusion list search          03606500
         L     R15,GVTGRHS0              RNL serch routine              03606600
         L     R3,GVTRSE                 set R3 to value in GVTRSE      03606700
         DROP  R3                                                       03606800
*                                                                       03606900
         BASR  R14,R15                                                  03607000
*                       R15=0 no_match  R15=RNLE pointer, match         03607100
         LTR   R15,R15                                                  03607200
         BZ    NO_RNLE                                                  03607300
         LA    R15,4                                                    03607400
NO_RNLE  EQU   *                                                        03607500
         LA    R6,0(R15,R6)          RETURN offset 0 = No_MATCH         03607600
*                                                   4 = MATCH           03607700
         LM    R1,R3,SAVE_REG        RESTORE R1-3                       03607800
         L     R13,SAVE_R13          RESTORE R13                        03607900
*---------------------------------------------------------------------- 03608000
*                                                                       03608100
* SETLOCK USES - R0 -1 -14 -15 when keyword REGS=USE is specified       03608200
*                                                                       03608300
*---------------------------------------------------------------------- 03608400
*                                                                       03608700
```

```
         SETLOCK RELEASE,TYPE=CMSEQDQ,REGS=USE,RELATED=ISGNQXIT         03608809
*                                                                       03608900
         SETLOCK RELEASE,TYPE=LOCAL,REGS=USE,RELATED=ISGNQXIT           03609000
*                                                                       03609100
         BR    R6                      RETURN                           03610203
*                                                                       03650000
MOVE_RNM MVC   RNAME(0),0(R10)                   MOVE RNAME     **       03650200
*                                                                       03650300
         DROP  R12                                                      03650415
         DROP  R9                                                       03650515
*--test--------------                                                  03650615
*ERROR    EQU   *                                                      03650715
*         LR    R8,R1                                                  03650815
*         WTO   'NAME/TOKEN CREATE ERROR, SHOULD NOT OCCUR'            03650915
*         LR    R1,R8                                                  03651015
*         BR    R6                      RETURN                         03651115
*--test--------------                                                  03652015
*-----------------------------------------------------------------     03680015
* END RNL SCAN                                                         03681015
*-----------------------------------------------------------------     03682015
*-------------------------------------------------------------------*  03690000
* DATA USED                                                         *  03700000
*-------------------------------------------------------------------*  03710000
         DS    0F                                                       03720000
COMP1    DC   A(X'80000000'+COMPRNL1)                                   03750000
*                                                                       03761000
HRDWNAME DC    CL8'HWRESERV'           MAJOR NAME FOR ALWAYS RESERVE    03761100
*                                                                       03761200
         DS    0F                                                       03762000
NAME     DC    CL8'ISGNQXIT'                                            03763000
         DC    CL8'ITSOEXIT'                                            03764000
*        DC    X'0000'             16 BYTE FILLER                       03765000
TOKEN    EQU   *                                                        03766000
         DC    4X'00'              WORK AREA POINTER                    03767000
         DC    12X'00'             OTHER 12 BYTES OF TOKEN              03768000
PERSIST  DC    x'00000001'         NAME/TOKEN PERSIST                   03769002
RETCODE  DC    4X'00'                                                   03769100
LEVEL    DC    X'00000004'         SYSTEM LEVEL                         03769200
FOUR     EQU   LEVEL                                                    03769300
ZERO     DC    X'00000000'         CONSTANT                             03769400
*                                                                       03769500
*                                                                       03769600
WORKAREA DSECT                                                          03769700
SAVE_REG DS    12X'00'             SAVE REG 1-2-3                       03769800
SAVE_R13 DS    04X'00'             SAVE REG 13                          03769900
SAVEAREA DS    18F'0'              STANDARD SAVE AREA FOR RNL SEARCH    03770000
OFFREG   EQU   *-WORKAREA                                               03770100
QNAME    DS    08C' '              MAJOR NAME                           03770200
RNAME    DS    256C' '             MINOR NAME                           03770300
WORK_LN  EQU   *-WORKAREA                                               03770400
*-------------------------------------------------------------------*  03771000
         DROP  BASEREG                                                  03780000
         EJECT                                                          03790000
         ISGRNLE                                                        03800000
         PRINT NOGEN                                                    03810000
         ISGYNQXP                                                       03820000
         ISGGVT                                                         03830000
         CVT  DSECT=YES                                                 03840000
         IHAPSA                                                         03850000
         IEFUCBOB DEVCLAS=DA                                            03860000
```

```
         IEANTASM                                                       03861000
*                                                                       03870000
         END                                                            03880000
//LINK  EXEC  PGM=IEWL,PARM='LIST,RENT,XREF'                            03890000
//SYSUT1 DD SPACE=(CYL,(1,1)),UNIT=SYSDA                                03900000
//SYSLMOD DD DSN=SYS1.SANDBOX.LINKLIB,DISP=SHR                          03910011
//SYSLIB  DD DSN=SYS1.CSSLIB,DISP=SHR                                   03911000
//SYSPRINT DD SYSOUT=*                                                  03920000
//SYSPUNCH  DD DSN=&OBJ(AUDIT),DISP=(OLD,PASS)                          03930000
//SYSUDUMP  DD SYSOUT=*                                                 03940000
//SYSLIN DD *                                                           03950000
 INCLUDE SYSPUNCH(AUDIT)                                                03960000
 ENTRY ISGCVXIT                                                         03970000
 NAME ISGNQPAT(R)                                                       03980019
```

Note: ENTRY ISGCVXIT is mandatory, whereas ISGNQPAT can be any name.

Archived

C

Sample GRS exit ISGNQXITFAST

This appendix contains sample exit code for GRS exit point ISGNQXITFAST that is introduced by APAR OW56028.

This APAR addresses GRS performance problems associated with ISGNQXIT, and is introduced as follows:

- Adds new exit points for exploitation by OEM serialization products such as CA-MII (MIM)
- Adds a faster alternative to ISGNQXIT for some environments
- Fixes an initialization problem where GRS did not detect any exit routines that were added via PROGxx

Installations that are on z/OS V1R2 and above and are using ENQ/DEQ installation exits, such as CA Multi-Image Integrity, may experience a performance degradation. This performance degradation will be apparent in address spaces that issue significant numbers of ENQ requests.

A joint fix for this problem is made available from IBM with APAR OW56028, and Computer Associates from their development team.

Note: You can also contact CA-MII support for additional methods to mitigate the performance problem.

This appendix contains the following:

- Sample exit code for exit ISGNQXITFAST

Note: The exit routine environments for ISGNQXIT (shown in Appendix B, “Sample GRS exit ISGNQXIT” on page 401) and for ISGNQXITFAST are different. Therefore, a new sample exit is required to support Cross-Sysplex DASD Sharing.

© Copyright IBM Corp. 2003

C.1 Sample exit ISGNQXITFAST download

This sample exit code is provided to support shared DASD with z/OS and OS/390 systems across GRSplexes. It is available from the Redbooks Web server (you must type the “SG” in upper case):

ftp://www.redbooks.ibm.com/redbooks/SG246581/

Download the file: isgnqxitfast.zip

Alternatively, you can go to:

http://www.redbooks.ibm.com

Select **Redbooks Online** -> **Additional Materials**. -> **SG246581**.

C.1.1 Sample exit zip file

To obtain the exit, you must download ISGNQXITFAST.ZIP. The zip file contains:

- ISGNQFST.ASM.TXT
- ISGNQFST.BIN
- ISGNQFST.README.TXT

The following is derived from the README file:

1) The file ISGNQFST.ASM must be uploaded in BINARY with LRECL=80.

2) When requested, respond to support type: PATTERN.

The ISGNQXITFAST exit has been modified to support type PATTERN for the RNL Exclusion table scan. A customer using the exit can now use type PATTERN definitions in all RNL tables, including Conversion.

For example, a customer can use the following RNL Conversion definition to convert all Reserves:

```
RNLDEF RNL(CON) TYPE(PATTERN)
 QNAME(*)                          /*  TEST FOR GRS EXIT        */
```

The entries in the RNL Conversion Table for special QNAME HWRESERV with RNAME being a volser (used by the exit to guarantee HW Reserve for cross-sysplex volume share), can *only* be type SPECIFIC or GENERIC, because using type PATTERN for these definitions may generate confusion.

The exit, during the RNL Conversion table scan, skips the type PATTERN entries, searches for qname HWRESERV, and then compares the volser using the length of the RNAME. As mentioned, type PATTERN is not supported if you have QNAME(HWRESERV).

Examples:

```
RNLDEF RNL(CON) TYPE(SPECIFIC)
QNAME(HWRESERV)               /*volser SBOX23 is shared cross sysplex*/
RNAME(SBOX23)

RNLDEF RNL(CON) TYPE(GENERIC)
QNAME(HWRESERV)               /*volsers starting with BOOK are shared*/
RNAME(BOOK)
```

C.2 Sample exit description

It is possible to support shared DASD with z/OS and OS/390 systems across GRSplexes, as shown in Figure C-1, by using the exit code provided in “Sample exit code” on page 431, and placing it in the GRS ISGNQXITFAST exit.

Figure C-1 Shared DASD between sysplexes

C.2.1 GRS exit points

For each ENQ/DEQ/RESERVE request with SCOPE=SYSTEM or SCOPE=SYSTEMS, the system invokes the ENQ/DEQ installation exit points.

For this purpose, GRS with wildcard support provides exit point ISGNQXIT. With APAR OW56028, GRS introduces three new exit points including ISGNQXITFAST, which is intended to offer a higher performance alternative to ISGNQXIT.

Both installation exits can modify attributes of the ENQ/DEQ/RESERVE request prior to Resource Names List (RNL) processing; the only difference is the environment with which the exits receive control. For additional information, refer to *z/OS MVS Installation Exits*, SA22-7593.

For the purpose of Cross-Sysplex DASD Sharing, only one of the provided sample installation exits ISGNQXITFAST or ISGNQXIT should be active. Both installation exits provide the same functionality and have the same external parameters.

C.2.2 GRS sample ISGNQXITFAST exit logic

Using the ISGNQXITFAST or ISGNQXIT GRS installation exit—and by adding, to the CONVERSION table, RNL definitions with QNAME not used by the system and RNAMEs that identify the volumes to share outside GRS—the RESERVE requests for the volumes included in the above RNL definitions will always result in an HW Reserve.

Restriction: The exit, however, does not propagate cross-sysplex ENQ requests with scope=SYSTEMS. For additional information, see “Exit restrictions” on page 427.

Example C-1 on page 424 shows an RNL definition that would be used by the ISGNQXIT exit.

Example: C-1 Sample RNL list to define sysplex DASD sharing

```
RNLDEF RNL(CON) TYPE(SPECIFIC|GENERIC)
QNAME(HWRESERV)
RNAME(VOLSER|volser-prefix)
```

The same RESERVE resource requests, which address other volumes, should have the possibility to be filtered through the conversion RNLs to have the HW reserve eliminated.

The ISGNQXITFAST or ISGNQXIT installation exit receives control for all RESERVE-ENQ-DEQ macros before the GRS RNLs processing logic. The exit traps the ENQ requests with a UCB pointer (scope SYSTEMS and HW RESERVE), and bypasses all the other ENQ-DEQ requests.

The exit locates the 'VOLSER' inside the parameter list and checks if the resource 'HWRESERV' and 'VOLSER' are present in the RNL conversion table. For the qname 'HWRESERV', only specific and generic RNL type entries are supported. For additional information, see “Exit restrictions” on page 427.

Note: If no match is found, the exit returns to GRS with no action.

Match found in RNL

The RNL exclusion list is searched to see if an RNL entry matches the request's major-minor name (specific, generic, and pattern type entries are supported). If this match is found, control is returned to GRS with no action. The ENQ request is then filtered by GRS through the RNL exclusion table and, because it matches, the scope is changed to SYSTEM with the HW Reserve issued.

If a match is not found, the bypass RNL processing bit is set in the parameter list and control is returned to GRS. GRS RNLs processing is not done, the request scope remains SYSTEMS, and the HW Reserve is issued. There is some overhead due to a double serialization being used instead of one.

RNL exclusion table search

The reason why the exit does the RNL exclusion table search is because it is has been done since OS/360; that is, to issue a RESERVE macro (ENQ scope SYSTEMS with the UCB pointer that implies a HW Reserve), and remove the RESERVE request with a DEQ for the same major-minor name, scope SYSTEMS, with or without a UCB pointer.

C.2.3 Scanning the RNL exclusion table examples

The following scenarios describe the exit doing a scan or no scan of the RNL exclusion table and an ENQ/DEQ combination where the DEQ is issued without a UCB pointer.

No scan of RNL exclusion table

If the exit does not scan the RNL exclusion table, the following occurs:

- The RESERVE request with major-minor name present in the RNL exclusion table is trapped by the exit and maintains the scope SYSTEMS plus HW Reserve, and GRS RNL processing is bypassed.
- The request may be ended with a DEQ without UCB pointer, major-minor name present in the RNL Exclusion table.
- The exit takes no action; control is given to GRS for RNL processing.

- GRS finds the major-minor name in the RNL exclusion table and changes the scope to SYSTEM.
- The DEQ may abend with a system code 130 because GRS knows about a scope SYSTEMS request.

In summary, a RESERVE request trapped by the exit has scope SYSTEMS plus HW Reserve, and may be ended with a DEQ scope SYSTEMS without a UCB pointer. In this scenario, without the RNL exclusion table scan, a DEQ with a major-minor name present in the RNL exclusion table would have the scope changed to SYSTEM by GRS RNL processing, and the DEQ would not remove the HW RESERVE because it is associated to a request with scope SYSTEMS. This could cause the DEQ to abend with a system code 130.

For example, RESERVEs for major name SYSIGGV2 are removed with DEQs with scope=SYSTEMS without a UCB pointer. Therefore, the RNL exclusion table scan prevents the exit from trapping RESERVE requests that will be excluded by GRS RNL processing, and therefore becoming scope=SYSTEM with a HW RESERVE.

Scan of RNL exclusion table

The same scenario with the exit scanning the RNL Exclusion table, and the same ENQ/DEQ combination where the DEQ is issued without a UCB pointer is as follows:

- A RESERVE request with major-minor name present in the RNL Exclusion table is not trapped by the exit.
- GRS RNL processing changes the scope to SYSTEM and HW Reserve is issued.
- A DEQ without a UCB pointer, major-minor name present in the RNL exclusion table.
- The exit takes no action, control is given to GRS for RNL processing.
- GRS finds the major-minor name in the RNL exclusion table and changes the request to SYSTEM
- The request is removed because GRS knows about the scope SYSTEM request.

Figure C-2 on page 426 shows the exit logic with the RNL processing flow.

Archived

Figure C-2 Exit logic with the RNL processing flow

Note: Because the RNL conversion table is used, any change to the table can be activated through the following z/OS operator command:

```
SET GRSRNL=xx
```

C.2.4 RNL example

Example C-2 on page 427 shows an example of a RNL conversion table to cross-sysplex HW RESERVE volume XA9RES and volumes with prefixes of CIX.

Volumes that should not have RESERVE requests converted (always reserved) are indicated to the exit using the RNL conversion table. The parameter is an RNL definition with the special QNAME(HWRESERV), and with the RNAME indicating the volser(s). The exit supports SPECIFIC and GENERIC type entries for QNAME(HWRESERV), and a PATTERN type entry is not allowed, the RNAME is a required parameter. For additional information, see ""Exit restrictions" on page 427. The special qnames can be indicated in any order in the conversion table.

It is recommended that the VOLUME(S) behind the QNAME(HWRESERV) should not be dynamically changed unless the devices are offline or not allocated.

The qname HWRESERV, shown in Example C-2 on page 427, is hardcoded and is defined at label HRDWNAME in the ISGNQXIT example.

Example: C-2 RESERVE conversion RNLexample

```
-----------------------------------------------------------------------------------------
Exclusion list examples
-----------------------------------------------------------------------------------------
RNLDEF RNL(EXCL) TYPE(PATTERN)
QNAME(SYSIGGV2)
RNAME(UCAT.?OS3*)                  /* ucat on HWRESERV volumes */

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSIGGV2)
RNAME(UCAT.VBOOK01)                /* ucat on HWRESERV volume */

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSZJES2)

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSCTLG)
-----------------------------------------------------------------------------------------
Conversion list examples
-----------------------------------------------------------------------------------------
RNLDEF RNL(CON) TYPE(PATTERN)
QNAME(*)                           /*Convert all Reserves */

RNLDEF RNL(CON) TYPE(SPECIFIC)
QNAME(HWRESERV)                     /*SPECIAL NAME*/
RNAME(XA9RES)                       /*ALWAYS RESERVE XA9RES*/

RNLDEF RNL(CON) TYPE(GENERIC)
QNAME(HWRESERV)                     /*SPECIAL NAME*/
RNAME(CIX)                          /*ALWAYS RESERVE VOLUMES*/
                                    /*BEGINNING WITH CIX   */
```

C.2.5 Exit restrictions

The sample ISGNQXITFAST and ISGNQXIT exits have some restrictions that should be understood and evaluated very carefully, as follows:

- The exit does not propagate cross-sysplex global ENQs (scope=SYSTEMS); it only guarantees that the HW RESERVEs are issued for volumes behind qname HWRESERV in the RNL conversion table for all RESERVE requests. Applications that logically serialize a DASD resource (PDS member or data set) with a global ENQ (scope=SYSTEMS) cannot depend on the exit to serialize resources on DASD shared between sysplexes.

  For example, ISPF serializes the edit of a PDS member with an ENQ scope=SYSTEMS. If a user is editing a PDS member, another user in the same GRSplex is notified that the member is in use if he tries to edit the same member. If a second user is in another GRSplex, he can edit the member. ISPF uses a RESERVE macro to protect only the write of the updated member. In this scenario, the first member's update will be overridden by the second member's update. The same situation should be expected with the Linkage-Editor that uses a RESERVE macro to protect only the write of the load module to disk.

- Type PATTERN is not supported for the special QNAME HWRESERV. The reason to restrict the RNL entry type to generic and specific is to allow the use of the RNLDEF RNL(CON) TYPE(PATTERN) QNAME (*) to convert all reserves. With the previous definition, and allowing the special QNAME HWRESERV to be type PATTERN, would have resulted in an always match for an all QNAME HWRESERV search, and as a consequence having all reserve requests not converted.

- It is possible to dynamically add and remove volumes behind QNAME(HWRESERV) with the **SET GRSRNL=xx** z/OS command, but the system does not check if the volumes have outstanding RESERVE requests before activating the new RNLs. It is recommended that the volume(s) specified behind the QNAME(HWRESERV) should not be dynamically changed unless the device is of-line or not allocated.

C.2.6 Recommendations

When a reserve request is intercepted by the exit, the request scope remains SYSTEMS and the HW Reserve is issued: a double serialization. A scenario where the exit controls volumes containing user catalogs and the systems in a sysplex are frequently accessing and updating the catalogs, will result in continuously cross-obtaining RESERVEs with the probability that the systems may eventually deadlock. Therefore, for volumes shared cross-sysplex and containing user catalogs, it is recommended to have the RNL exclusion list entries for catalog QNAME SYSIGGV2 and RNAME ucat-name, and specify either type GENERIC, SPECIFIC, or PATTERN. For an example, see the exclusion list example in Figure C-2 on page 426.

Use the GRS option SYNCHRES(YES) (synchronous reserve). It can be activated through either the GRSCNFxx parmlib member or the **SETGRS** command. The command has system scope. The SYNCHRES option allows an installation to specify whether the system should obtain a hardware RESERVE for a device prior to granting a global resource serialization ENQ. This option might protect jobs that have a delay between a hardware RESERVE request being issued and the first I/O operation to the device. Prior to the implementation of the SYNCHRES option, the opportunity for a deadlock situation was more likely to occur.

C.2.7 Exit compatibility

To support sysplex DASD sharing in configurations where the sysplexes may have GRS with and without wildcard support, the ISGNQXIT dynamic exit is compatible with the ISGGREX1 ITSO exit provided for MVS and OS/390 systems without GRS wildcard support. Both exits have the same scope and functionality. The ISGNQXIT exit is a replacement for the ISGGREX1 exit beginning with z/OS Version 1 Release 2.

C.2.8 Exit installation and activation

The exit can be activated by one of the following methods:

- With a program using the CSVDYNEX macro.
- With the **SETPROG EXIT** z/OS operator command—and assuming that the exit has been linked in SYS1.USER.LINKLIB with the name ISGNQFST— issue the command as follows:

```
SETPROG EXIT,ADD,EX=ISGNQXIT,MOD=ISGNQXITFAST,DSN=SYS1.USER.LINKLIB
```

- Using an EXIT STATEMENT in the PROGXX parmlib member, specify the member as follows:

```
EXIT ADD
EXITNAME(ISGNQXITFAST)
MODNAME(ISGNQFST)
STATE=ACTIVE
DSNAME(SYS1.USER.LINKLIB)
```

Following is the recommended procedure to activate the exit after an IPL:

- Activate the exit in all systems sharing DASD across the sysplexes.
- Update the RNL conversion table in the GRSRNLxx parmlib member.

- Activate the RNLs in all systems sharing DASD across the sysplexes.

For additional information, refer to *z/OS MVS Installation Exits*, SA22-7593.

C.2.9 Installation exits ISGNGXITFAST and ISGNQXIT considerations

GRS implementation of two exit points for the same purpose has the following considerations to clarify the rationale behind the exits; the considerations are provided here as a series of questions and answers:

1. Why does GRS have two installation exit points, ISGNQXITFAST and ISGNQXIT?

   Both installation exits have the same scope. ISGNQXITFAST has been introduced by GRS with APAR OW56028 and is intended to offer a higher performance alternative to ISGNQXIT.

2. Which exit receives control first?

   ISGNQXITFAST receives control first.

3. If the first exit ISGNQXITFAST indicates something to GRS, does the second exit ISGNQXIT receive control with information about the decision taken by the first exit, or will this exit be bypassed?

   ISGNQXIT will receive control with information about the decision taken by ISGNQXITFAST.

4. What kind of problems can be encountered by having both exits active?

   There should be no problem having multiple exits at one exit point because the last changes always prevail.

5. Should it be recommended to have only one exit active?

   Yes; even if both exits can be active, we recommend that you have only *one* exit active.

6. Is there any functional difference in the ITSO-provided installation exits ISGNQXITFAST and ISGNQXIT?

   Both installation exits provide the same functionality and have the same external parameters.

C.2.10 EXIT process verification

To verify if the exit is behaving as expected, you can use the ENQ/RESERVE/DEQ Monitor. Monitor selection 1, MAJOR Names Display, has been extended with a new column to indicate if the request has been modified by exit ISGNQXIT.

Monitor selection 1

Example C-3 on page 430 shows a MAJOR Names display example.

Example: C-3 ENQ/DEQ Monitor selection 1

```
                        ENQ/DEQ Monitor - Major Name List      Row 1 to 20 of 29

 Enter S to select a Major Name for details     .
       L major on command line to locate a Major.   Elapsed seconds:      99

    Sel.   ----------   -----   ----   -----   -------  -Average-   -Reserved-
    Field  Major Name   Scope   Exit   RNL     Counter     msec       seconds
    _      SYSZJES2    *RES                        101       21            2
    _      SYSZVVDS    *RES     YES                  2        2            0
    _      SYSVTOC     *RES     YES                  2       69            0
    _      SYSIGGV2    *RES                          2       14            0
    _      SPFEDIT      SYSS                         9
    _      IGDCDSXS     SYSS                         7
    _      CHANGEQU     SYSS                        10
    _      AUDITCOD     SYSS           NO           26
    _                   SYSS           NO           60
    _      SYSZVVDS     SYSS                       168
    _      SYSZRACF     SYSS                         8
    _      SYSZMCS      SYSS           NO            7
    _      SYSZIOS      SYSS           NO           11
    _      SYSZENQM     SYSS           NO            1
    _      SYSZDSCB     SYSS                         2
    _      SYSZATR      SYSS           NO           30
    _      SYSVSAM      SYSS                        29
    _      SYSIGGV2     SYSS                        96
    _      SYSDSN      *SYSS                        11
    _      SIBIXFP      SYS                          1
```

Example C-3 shows that RESERVE requests for Major names SYSZVVDS and SYSVTOC have been processed by exit ISGNQXIT, and that the HW Reserves have been issued.

Monitor selection 3

Using Monitor selection 3 and the VOLUME list display, you can see the volumes where HW Reserves have been issued. Example C-4 shows two volumes:

- BOOK01 shared across two sysplexes, and therefore with a HWRESERV entry in the RNL conversion table
- SBOX23 with JES2 checkpoint that has its RESERVE major name in the RNL Exclusion table

Example: C-4 Volume List display with Monitor selection 3

```
                         ENQ/DEQ Monitor - VOLUME List         Row 1 to 2 of 2

 Enter S to  select a Volume for details
       A for active Reserves on  Volume
       L volume on command line to locate a Volume
   *   indicates volume where reserves are not converted
-- - ------ ------- Dev. Max  --------- ------------ Reserve Time -------------
S.   Volume Tot.Res nbr   Res Elap(sec)  Avg.(ms)  Min.(ms)  Max.(ms) Tot.(sec)
 _ * BOOK01      6  2601   04       192         0         0         0         0
 _ * SBOX23    247  2558   01       248         0        17        69         0
 ****************************** Bottom of data ********************************
```

Selecting the entry for volume BOOK01, the major-minor name combinations of the RESERVE macros are displayed, as shown in Example C-5 on page 431.

Example: C-5 BOOK01 detail display

```
                      ENQ/DEQ Monitor - VOLUME Entry List      Row 1 to 3 of 3

 Volser. . . . . . : BOOK01      Average Reserve Time (ms) : 0
 Tot.nr of Reserve : 6           Minimum Reserve Time (ms) : 0
 Dev.nr. . . . . . : 2601        Maximum Reserve Time (ms) : 0
 Max Reserve Cnt. .: 04          Total   Reserve Time (sec): 0
 Elapsed Time (sec): 192         Volume  Reserve Rate (min): 2

 Interval
- Rate -- ----- -------- ----------------------  ------------ Time ------------
S  min.   Count MajName  Minor name (max 22 ch)  Avg ms  Min ms  Max ms Tot sec
_     0       2 SYSIGGV2 UCAT.VBOOK01                14       7      22       0
_     0       2 SYSVTOC  BOOK01                      68      58      79       0
_     0       2 SYSZVVDS BOOK01                       2       2       2       0
****************************** Bottom of data ********************************
```

For additional information about the GRS monitor, refer to *z/OS MVS Planning: Global Resource Serialization*, SA22-7600.

C.2.11 Sample exit code

Example C-6 is the sample exit code to place into GRS exit ISGNQXIT.

Example: C-6 Sample exit code to place into GRS exit ISGNQXIT

```
//MERONI   JOB (999,POK),CLASS=A,                                    00010000
//         MSGLEVEL=(1,1),MSGCLASS=X,                                00020000
//         NOTIFY=&SYSUID                                            00030000
//*-------------------------------------------------------------     00031005
//* This code depend on nqxp_workarea that did not exist             00031107
//* at ISGYNQXP macro version 0.                                     00031209
//*-------------------------------------------------------------     00032005
//ASMH   EXEC PGM=ASMA90,REGION=1024K,                               00040000
//           PARM='DECK,XREF(SHORT),SYSPARM(-V1R2M1-)'               00040101
//SYSPRINT  DD SYSOUT=*                                              00050000
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,5)),DISP=(NEW,DELETE)        00060000
//SYSPUNCH  DD DSN=&OBJ(AUDIT),DISP=(,PASS,DELETE),UNIT=SYSDA,       00070000
//             SPACE=(TRK,(1,5,5))                                   00080000
//SYSPUNCH  DD DUMMY                                                 00090000
//SYSGO     DD DUMMY                                                 00100000
//SYSLIN    DD DUMMY                                                 00110000
//SYSLIB    DD DSN=SYS1.MACLIB,DISP=SHR                              00120000
//         DD DSN=SYS1.MODGEN,DISP=SHR                               00130000
//SYSIN DD *                                                         00140000
        TITLE 'ISGNQXITFAST - ENQ/DEQ EXIT ROUTINE'                  00150009
ISGNQFST CSECT                                                       00160010
*/*  START OF SPECIFICATIONS ****                                   00170000
*-------------------------------------------------------------------*00180000
* Cross Sysplex DASD sharing                                        *00190000
* Always H/W RESERVE Volumes shared cross GRS complexes             *00200000
*-------------------------------------------------------------------*00210000
*                                                                   *00220000
* It is mandatory to support SHARED DASD with OS/390 Systems        *00230000
* cross GRS-plexes that all RESERVE requests that address these     *00240000
* VOLUMES result in an H/W RESERVE whatever the RESOURCE NAME is.   *00250000
*                                                                   *00260000
* The same RESOURCES, that address other VOLUMES, should have the   *00270000
* possibility to be filtered through the CONVERSION RNLs and have   *00280000
```

```
* the H/W RESERVE eliminated.                                        *
*                                                                    *
* Using ISGNQXIT GRS exit and by adding to the CONVERSION table      *
* a QNAME not used by the System and RNAMEs that identify the        *
* Volumes to share outside GRS, the RESERVE requests for the         *
* Volumes included in the following definition will always result *
* in a H/W RESERVE.                                                  *
*                                                                    *
*      RNLDEF RNL(CON) TYPE(SPECIFIC/generic)                        *
*      QNAME(HWRESERV)                                               *
*      RNAME(VOLSER/volser-prefix)                                   *
*                                                                    *
* LOGIC:                                                             *
*                                                                    *
* ISGNQXIT  receives control for all RESERVE/DEQ  requests           *
* (SYSTEMS+H/W RESERVE), locates the 'VOLSER'  and checks if the     *
* resource 'HWRESERV' 'VOLSER' is present in the conversion          *
* table. If found the RNL processing is bypassed, if not normal      *
* RNL scan is perormed.                                              *
*                                                                    *
* Because the RNL CONVERSION table is used, any change to the        *
* table can be activated through OS/390 operator command:            *
* SET GRSRNL=xx.                                                     *
*                                                                    *
* The name 'HWRESERV' is hard-coded and is defines at label          *
* HRDWNAME in the ISGNQXIT example.                                  *
*                                                                    *
* NOTE: the VOLUME(S) behind the QNAME(HWRESERV) should not be       *
* dynamically changed unless the device(s) is(are) offline.          *
*                                                                    *
* TYPE PATTERN IS NOT SUPPORTED FOR HWRESERV RNL ENTRIES.            *
*                                                                    *
*--------------------------------------------------------------------*
*                CONVERSION LIST EXAMPLE                             *
*--------------------------------------------------------------------*
*                                                                    *
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *
* QNAME(SYSVTOC)                                                     *
*                                                                    *
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *
* QNAME(SYSIGGV2)                                                    *
*                                                                    *
* RNLDEF RNL(CON) TYPE(SPECIFIC)                                     *
* QNAME(HWRESERV)                    /*SPECIAL NAME*/               *
* RNAME(XA9RES)                      /*ALWAYS RESERV  XA9RES*/      *
*                                                                    *
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *
* QNAME(HWRESERV)                    /*SPECIAL NAME*/               *
* RNAME(CIX)                         /*ALWAYS RESERV VOLUMES*/      *
*                                    /*BEGINNING WITH CIX   */      *
*                                                                    *
***                        END OF                                   *
***   RESERVE CONVERSION RESOURCE NAME LIST - SAMPLE                 *
**********************************************************************
* RESTRICTION:                                                       *
*                                                                    *
* 1-The exit does not propagate cross Sysplex ENQ requests with      *
* scope=SYSTEMS, only guarantees that the H/W Reserve is issued      *
* for Volumes behind qname HWRESERV in RNL Conversion table.         *
*                                                                    *
```

```
* 2-Type PATTERN is not supported for HWRESERV RNL entries        *     00890000
*                                                                  *     00900000
*Note-Restriction of type PATTERN not supported for EXCLUSION RNL *     00901000
*     HAS BEEN REMOVED                                             *     00901100
*                                                                  *     00901200
*******************************************************************     00901300
* COMPATIBILITY:                                                   *     00901400
*                                                                  *     00901500
* ISGNQXIT is compatible with ISGGREX1 ITSO exit provided for      *     00901600
* OS/390 Systems without GRS WILDCARD support.                     *     00901700
*                                                                  *     00901800
*******************************************************************     00901900
*                                                                  *     00902000
                                                                          00903000
*******************************************************************     00904000
**EXIT INSTALLATION                                                *     00905000
*                                                                  *     00906000
* THE EXIT CAN BE INSTALLED WITH ONE OF THE FOLLOWING METHODS      *     00907000
*                                                                  *     00908000
* 1.WITH A PROGRAM USING CSVDYNEX MACRO                            *     00909000
*                                                                  *     00910000
* 2.WITH SETPROG EXIT OPERATOR COMMAND                             *     00920000
*   EX. SETPROG EXIT,ADD,EX=ISGNQXITFAST,MOD=ISGNQFST,             *     00930008
*       DSN=SYS1.USER.LINKLIB                                      *     00940000
*                                                                  *     00950000
* 3.USING EXIT STATEMENT OF THE PROGXX PARMLIB MEMBER              *     00960000
*       EXIT ADD                                                   *     00970000
*            EXITNAME(ISGNQXITFAST)                                *     00980008
*            MODNAME(ISGNQFST)                                     *     00990008
*            STATE=ACTIVE                                          *     01000000
*            DSNAME(SYS1.USER.LINKLIB)                             *     01010000
*                                                                  *     01020000
*                                                                  *     01030000
*NOTE: Recommnded procedure to activate the exit after IPL:        *     01040000
*      1.Activate the exit in all Systems sharing ALWAYS RESERVE   *     01050000
*        VOLUMES                                                   *     01060000
*      2.Activate the RNLS in all Systems sharing ALWAYS RESERVE   *     01070000
*        VOLUMES. The Volume(s) behind the QNAME(HWRESERV) should *     01080000
*        not be dynamically added/removed unless the devices are   *     01090000
*        OFFLINE                                                   *     01100000
*******************************************************************     01110000
       EJECT                                                              01120000
*******************************************************************     01130000
*                                                                          01140000
* INPUT R1=NQXP, R13=STANDARD SAVE AREA, R14=RETURN ADDRESS                01141000
*       R15=ENTRY POINT                                                    01142000
*                                                                          01143000
*                                                                          01144000
*       REGISTERS-SAVED =   R0 - R12, R14, R15                             01150000
*                                                                          01300000
*       REGISTERS-RESTORED = R0 - R12, R14                                 01310000
*                                                                          01330000
*       RETURN-CODES =   R15 = 0 -                                         01350000
*                              4 - NOT USED                                01360000
*                                                                          01370000
*       EXIT-ERROR = NONE                                                  01380000
*                                                                          01390000
*       WAIT-STATE-CODES = NONE                                            01400000
*                                                                          01430000
*       ROUTINES = ISGGRHSO  (GRS RNL SEARCH)                              01440000
```

```
*                                                                       01450000
*       DATA-AREAS = WORKAREA                                           01460000
*                                                                       01470000
*       CONTROL-BLOCKS = CVT       R                                    01490000
*                        NQXP      R/W                                  01500000
*                        GVT       R                                    01510000
*                                                                       01520000
*       TABLES = 1. RESERVE CONVERSION RNL        (ISGGCRNL)            01540000
*                2. EXCLUSION RNL                 (ISGGERNL)            01541000
*                                                                       01550000
*       SERIALIZATION = LOCAL, CMSEQDQ LOCKS                            01560000
*                                                                       01570000
*                                                                       01571000
*01*  CHANGE-ACTIVITY =  INITIAL RELEASE 1.0                   08/01   01580000
*                                                                       01590000
*02*  RELEASE 1.1.1                                            06/02   01650000
*     Type PATTERN support for RNL EXCL table search                    01660000
*                                                                       01670000
*03*  RELEASE 1.2.1                                            09/02   01671001
*     Work are for exit from NQPL (nqxp_workarea)                      01672004
*                                                                       01673001
*     MESSAGES = NONE.                                                  01680000
*                                                                       01690000
*     ABEND-CODES = NONE.                                               01700000
*                                                                       01710000
**** END OF SPECIFICATIONS **                                        */ 01720000
         EJECT                                                          01730000
********************************************************************** 01740000
*                                                                    * 01750000
*        REGISTER ASSIGNMENTS                                        * 01760000
*                                                                    * 01770000
********************************************************************** 01780000
*                                                                       01790000
RNLEPTR  EQU   2                    POINTER TO AN RNL ENTRY (RNLE)     01820000
FLENRNLE EQU   4                    LENGTH OF FIXED PART OF AN RNLE    01830000
RNAMELEN EQU   5                    LENGTH OF RNAME                    01840000
BASEREG  EQU   11                   BASE REGISTER                      01850000
WORKREG  EQU   12                   WORK REGISTER                      01860000
*                                                                       01900000
R0       EQU   0                                                       01910000
R1       EQU   1                                                       01920000
R2       EQU   2                    POINTER TO AN RNLE                 01930000
R3       EQU   3                    CVT, GVT                           01940000
R4       EQU   4                    Length of RNLE fixed part          01950000
R5       EQU   5                    Length of RNAME                    01960000
R6       EQU   6                    RETURN REG, RNL SEARCH             01970000
R7       EQU   7                    RNL list to search I=1,E=2,C=3     01980000
R8       EQU   8                                                       01990000
R9       EQU   9                    NQPX POINTER                       02000000
R10      EQU   10                                                      02010000
R11      EQU   11                   BASE REG                           02020000
R12      EQU   12                   WORK REG                           02030000
R13      EQU   13                   SAVE AREA POINTER                  02040000
R14      EQU   14                   RETURN ADDRESS                     02050000
R15      EQU   15                                                      02060000
       EJECT                                                            02070000
********************************************************************** 02080000
*                                                                    * 02090000
*        LOGIC FLOW FOR ISGNQXITFAST EXIT                            * 02100008
*                                                                    * 02110000
```

```
*********************************************************************** 02120000
*/*                                                                   */ 02130000
*/*++ 'ISGNQXITFAST': ENTRY TO ENQ DEQ EXIT                           */ 02140008
*/*++ ESTABLISH ADDRESSABILITY                                        */ 02150000
*/*++ IF DEQ                            --> RETURN                    */ 02160000
*/*++ IF THE NQXP  UCB POINTER NOT ZERO --> RETURN                    */ 02170000
*/*                                                                   */ 02180000
*/*++   LOCATE RNL CONVERSION TABLE POINTER                            */ 02190000
*/*++     DO WHILE MATCH NOT FOUND AND NOT LAST ENTRY IN THE RNL      */ 02200000
*/*++       IF RNL MAJOR NAME IS HWRESERV                             */ 02210000
*/*++         IF RNLE MINOR NAME EQUALS NQXP VOLSER                   */ 02220000
*/*++           GOTO SCAN_RNL_EXCLUSION_TABLE                         */ 02230000
*/*++             IF RNLE FOUND ----->  RETURN                        */ 02230100
*/*++             IF RNLE NOT_FOUND                                   */ 02230200
*/*++                SET BYPASS_RNL_PROCESSING                        */ 02231000
*/*++                RETURN                                           */ 02232000
*/*++             ENDIF                                               */ 02240000
*/*++         ENDIF    (END OF RNAME COMPARISON)                      */ 02241000
*/*++       ENDIF  (END OF CHECK FOR RNL MAJOR NAME HWRESERV          */ 02250000
*/*++     ENDDO  (REPEAT SEQUENCE UNTIL MATCH FOUND OR END OF RNL)    */ 02260000
*/*++    ENDIF  (END OF ENXP  PROCESSING)                             */ 02270000
*/*++ RETURN                                                          */ 02280000
*/*                                                                   */ 02290000
*/*  :SCAN_RNL_EXCLUSION_TABLE                                        */ 02300000
*/*++   OBTAIN LOCAL + CMSNQDQ LOCKS IF NOT HELD ON ENTRY              */ 02310008
*/*++       GET WORK AREA FROM NQPL                                   */ 02321001
*/*++     INITIALIZE WORK AREA WITH Q/RNAME & RNLE TO SEARCH          */ 02340000
*/*++     CALL ISGGRHSO (GRS RNLE SEARCH ROUTINE)                     */ 02341000
*/*++   RELEASE LOCAL + CMSNQDQ LOCKS IF OBTAINED ON ENTRY            */ 02341108
*/*++   RETURN OFFSET 0=NO_MATCH  4=MATCH                             */ 02342000
*/*  :END_OF_EXCLUSION_RNL SEARCH                                     */ 02350000
*/*++ END 'ISGNQXIT'                                                  */ 02380000
*/*                                                                   */ 02390000
*/*******************************************************************/ 02400000
         SPACE 3                                                       02410000
ISGNQFST AMODE 31                                                      02420010
ISGNQFST RMODE ANY                                                     02430010
         MODID BR=NO,MODLBL=ITSOQFST                                   02440010
ISGCVXIT DS    0H                                                      02450000
         ENTRY ISGCVXIT                                                02480000
         SPACE                                                         02490000
         STM   14,12,12(13)            SAVE ENTRY REGS                 02500000
         LR    BASEREG,R15                                             02510000
         USING ISGCVXIT,BASEREG                                        02520000
         LR    R9,R1                                                   02521000
         USING NQXP,R9                 ENQ PLIST ADDRESSABILITY        02530000
*                                                                      02540000
         TM    NQXP_STATEFLAGS1,NQXP_SF1_ENQ  RETURN IF DEQ            02560000
         BZ    MODEXIT                                                 02570000
*                                                                      02590000
         ICM   R6,15,NQXP_OP_UCB       UCB POITER                     02600000
         BZ    MODEXIT                  NO RESERVE, EXIT               02610000
*                                       LOCATE RNL CONV                02620000
         L     R3,FLCCVT-PSA            CVT                            02630000
         USING CVT,R3                                                  02640000
         L     R3,CVTGVT                GET GRS VECTOR                 02650000
         DROP  3                                                       02660000
         USING GVT,R3                                                  02670000
         TM    GVTVFLAG,GVTVCRNL        RESERVE CONVERSION INVALID     02680000
         BO    MODEXIT                  YES, EXIT                      02690000
```

```
*                                                                       02700000
         TM    GVTVFLAG,GVTVERNL        SYSTEMS EXCLUSION  INVALID     02710000
         BO    MODEXIT                  YES, EXIT                      02720000
*                                                                       02730000
         USING RNLE,RNLEPTR                                            02740000
         ICM   RNLEPTR,15,GVTRCRNL      RESERVE CONVERTION RNL         02750000
         BZ    MODEXIT                  NO RNL CONVERSION              02760000
         L     R14,COMP1                RETURN ADDRESS                 02770000
*                                                                       02780000
*-----------------------------------------------------------------*     02790000
* NQXP_RD_VOLSER = VOLSER OF REQUESTED UCB                              02800000
*-----------------------------------------------------------------*     02810000
*                                                                       02820000
COMPRNL1 EQU   *                                                       02830000
         TM    RNLEFLGS,RNLELAST       AT END OF THE RNL ?              02840000
         BO    MODEXIT                 YES => RNL SCAN DONE, NOMATCH    02850000
*                                      RETURN                           02860000
* COMPARE QNAME HWRESERV   WITH RNLE                                    02880000
*                                                                       02890000
         TM    RNLEFLGS,RNLEPATT       PATTERN TYPE ENTRY NOT           02900000
         BO    NEXTRNL1                SUPPORTED                        02910000
*                                                                       02920000
         CLC   HRDWNAME,RNLEQNME       CHECK IF QNAME HWRESERV          02930000
         BNE   NEXTRNL1                NO => GET NEXT RNL ENTRY         02940000
*                                                                       02950000
* COMPARE THE VOLSER, MUST BE SPECIFIED                                 02960000
*                                                                       02961000
         SLR   RNAMELEN,RNAMELEN       CLEAR WORK REG                   02970000
         IC    RNAMELEN,RNLERNML       GET LENGTH OF RNAME IN THE RNLE  02980000
*                                                                       02990000
         LTR   RNAMELEN,RNAMELEN       MINOR LENGTH CANNOT BE ZERO,     03000000
         BZ    NEXTRNL1                VOLSER MUST BE INDICATED         03010000
*                                                                       03020000
         BCTR  RNAMELEN,0              ADJUST LENGTH                    03030000
         EX    RNAMELEN,COMPRNM1       COMPARE THE RNAME (VOLSER)       03040000
         BNE   NEXTRNL1                UNEQUAL => GET NEXT RNLE         03050000
*                                      VOLUME SERIAL FOUND IN THE RNL   03070000
*                                      CHK IF RESOURSE IS IN RNL EXCL   03080000
*-----------------------------------------------------------------*     03090000
* EXCLUSION RNL LIST SEARCH                                             03100000
*-----------------------------------------------------------------*     03110000
         ICM   RNLEPTR,15,GVTSERNL     SYSTEMS EXCLUSION  RNL           03120000
         BZ    BYP_RNL                 NO RNL EXCLUSION,                03130000
         LA    R7,2                    search Exclusion RNL             03131000
*                                      SET BYPASS RNL PROCESSING        03140000
         BAS   R6,RNL_SRCH             RNL serach routine               03150000
*                                                                       03160000
         B     BYP_RNL     OFFSET --> 0   NO MATCH, set bypass RNL     03162000
         B     MODEXIT     OFFSET --> 4   MATCH, EXIT                  03162100
*                                                                       03163000
BYP_RNL  EQU   *                                                       03220000
*-----------------------------------------------------------------     03221000
*  Support NQXP_FLAGS1 name change                                      03221100
*-----------------------------------------------------------------     03221200
         AIF   ('&SYSTEM_ID'(1,1) EQ 'O').OS390                        03222000
*                                'OS/390 02.XX.00'                      03223200
         AIF   ('&SYSTEM_ID' EQ 'z/OS 01.01.00').OS390                 03223400
*                                      SET BYPASS RLN PROCESSING        03226000
         OI    NQXP_REQUESTFLAGS1,NQXP_RF1_BYPASSRNLS  Z/OS 1.2 AND UP  03230000
         AGO   .ZOS                                                    03230100
```

```
.OS390   OI    NQXP_FLAGS1,NQXP_RF1_BYPASSRNLS    OS/390 AND Z/OS 1.1   03233100
.ZOS     ANOP                                                           03234000
*-----------------------------------------------------------------------  03235000
         B     MODEXIT                                                   03240000
*                                                                        03242000
COMPRNM1 CLC   RNLERNME(0),NQXP_RD_VOLSER       COMPARE VOLSER           03243000
*OMPRNM1 CLC   RNLERNME(0),UCBVOLI-UCBOB(R6)    COMPARE VOLSER           03244000
*-----------------------------------------------------------------------  03570800
* R13 SAVE AREA, R14, RETURN ADDRESS, R15 RET-CODE                       03570900
*-----------------------------------------------------------------------  03571000
MODEXIT1 EQU   *                                                         03571200
         LA    R15,4                   *TEST                             03571300
         B     CC_EXT                                                    03571400
MODEXIT  EQU   *                                                         03571500
         SR    R15,R15                 RETURN CODE 0                     03571600
CC_EXT   EQU   *                                                         03571700
         L     R14,12(0,R13)           RECOVER THE RETURN ADDRESS        03571800
         LM    R0,R12,20(R13)          RECOVER OTHERS EXCEPT R15         03571900
         BSM   0,R14                   RETURN TO THE CALLER              03572000
*                                                                        03572100
*-----------------------------------------------------------------------  03572900
* GET THE ADDRESS OF THE NEXT RNL ENTRY                                  03573000
*-----------------------------------------------------------------------  03573100
NEXTRNL1 EQU   *                                                         03573200
         LA    FLENRNLE,RNLERNME-RNLE  LENGTH OF FIXED PART OF RNLE      03573300
         SLR   WORKREG,WORKREG         CLEAR WORK REG                    03573400
         IC    WORKREG,RNLERNML        GET RNAME LENGTH (VARIABLE)       03573500
         ALR   WORKREG,FLENRNLE        ADD FIXED + VARIABLE LENGTHS      03573600
         ALR   RNLEPTR,WORKREG         GET ADDRESS OF NEXT RNL ENTRY     03573700
         BSM   0,R14                   CHECK THE NEW RNL ENTRY           03573800
         DROP  RNLEPTR                                                   03573900
*-----------------------------------------------------------------*      03574000
* END  OF THE NEXT RNL ENTRY                                             03574100
*-----------------------------------------------------------------*      03574200
*                                                                        03574300
*-----------------------------------------------------------------       03574400
*                                                                        03574500
*     SUBROUTINES                                                        03574600
*                                                                        03574900
RNL_SRCH EQU   *                                                         03575100
*-----------------------------------------------------------------------  03576000
* RNL search routine                                                     03580000
*                                                                        03590000
* Input R9=NQXP , R3=GVT,  R6=return                                     03591000
*       R7=RNL list to search: Input=1, Exclusion=2, Conversion=3        03591100
*                                                                        03591200
* Output R6+0=RNLE not found  R6+4=RNLE found                            03591300
*                                                                        03591400
*logic:                                                                  03591501
*   obtain LOCAL + CMSEQDQ locks if not held on entry                    03591607
*       get pointer to work area FROM NQPL, init with qname+rname        03592601
*            call rnl search routine GVTGRHS0                            03592700
*              R0=2 Exclusion RNL search  (1=Inclusion, 3=Conversion)    03592804
*              R1=pointer to work area, R2=rname length,                 03592904
*              R3=value in GVTRSE                                        03593004
*   release LOCAL + CMSEQDQ locks if obtained on entry                   03593108
*   return R6+0=RNLE not found  R6+4=RNLE found                          03593201
*                                                                        03593304
*-------------------------------------------------------------------------- 03593400
*                                                                        03593511
```

```
         L     R12,NQXP_WORKAREA         GET WORK AREA FROM NQPL       03593611
         USING WORKAREA,R12                                            03593711
*------------------------------------------------------------------- 03593811
*  Test if LOCAL and CMSEQDQ are held on entry                        03593907
*------------------------------------------------------------------- 03594011
         SETLOCK TEST,TYPE=CMS,BRANCH=(HELD,OK_LOCK),RELATED=ISGNQXIT   03594107
*                                                                     03594207
*------------------------------------------------------------------- 03594307
*                                                                     03594407
* SETLOCK OBTAIN/RELEASE                                              03594507
*         uses - R0 -1 -14 -15 when keyword REGS=USE is specified     03594607
*                                                                     03594707
*------------------------------------------------------------------- 03594807
*                                                                     03595407
         SETLOCK OBTAIN,TYPE=LOCAL,REGS=USE,RELATED=ISGNQXIT,         X03595507
               MODE=UNCOND                                            03595607
*                                                                     03595707
         SETLOCK OBTAIN,TYPE=CMSEQDQ,REGS=USE,RELATED=ISGNQXIT,       X03595807
               MODE=UNCOND                                            03595907
*                                                                     03596007
         B     LOCK_OBT                                               03597007
*                                                                     03598007
*------------------------------------------------------------------- 03604100
*        R9=NQPL  R3=GVT                                              03604300
*        R6=RETURN ADDRESS  +0 not_found   +4 found                   03604400
*        R7=RNL list to search                                        03604500
*-------------------------------------------------------------------* 03604600
*                                                                     03604701
OK_LOCK EQU    *                                                      03604807
         MVI   LOCK_HLD,X'FF'            indicate locks held on entry  03605207
LOCK_OBT EQU   *                                                      03605307
         STM   R1,R3,SAVE_REG            SAVE R1-R3                    03605500
         ST    R13,SAVE_R13              SAVE R13                      03605600
         LA    R13,SAVEAREA              STANDARD SAVE AREA            03605700
         MVC   QNAME(8),NQXP_OP_QNAME                                  03605800
         XC    RNAME(256),RNAME          CLEAR RNAME IN WORK AREA      03605900
         ICM   R10,15,NQXP_OP_RNAME      RNAME POINTER                 03606000
         SR    R2,R2                                                   03606100
         IC    R2,NQXP_OP_RNAMELEN       RNAME LEN                     03606200
         BCTR  R2,0                       -1                           03606300
         EX    R2,MOVE_RNM               MOVE RNAME                    03606400
         LA    R2,1(0,R2)                +1 RNAME length R2            03606500
         ST    R2,SAVEAREA               SAVE RNAME length FOR DIAG    03606600
         LA    R1,OFFREG(0,R12)          R1 pointer to qname-rname     03606700
         LR    R0,R7                     Exclusion list search         03606800
         L     R15,GVTGRHS0              RNL serch routine             03606900
         L     R3,GVTRSE                 set R3 to value in GVTRSE     03607000
         DROP  R3                                                      03607100
*                                                                     03607200
         BASR  R14,R15                                                 03607300
*                          R15=0 no_match  R15=RNLE pointer, match    03607400
         LTR   R15,R15                                                 03607500
         BZ    NO_RNLE                                                 03607600
         LA    R15,4                                                   03607700
NO_RNLE  EQU   *                                                      03607800
         LA    R6,0(R15,R6)          RETURN offset 0 = No_MATCH        03607900
*                                           4 = MATCH                 03608000
         LM    R1,R3,SAVE_REG        RESTORE R1-3                      03608100
         L     R13,SAVE_R13          RESTORE R13                       03608200
*                                           4 = MATCH                 03608307
```

```
         TM    LOCK_HLD,X'FF'             locks held on entry          03608407
         BNO   LOCK_RLS                                               03608507
         BR    R6                                                     03608607
*                                                 4 = MATCH           03608707
*-------------------------------------------------------------------- 03608800
*                                                                      03608900
* SETLOCK USES - R0 -1 -14 -15 when keyword REGS=USE is specified     03609000
*                                                                      03609100
*-------------------------------------------------------------------- 03609200
*                                                                      03609300
LOCK_RLS EQU   *                                                      03609407
         SETLOCK RELEASE,TYPE=CMSEQDQ,REGS=USE,RELATED=ISGNQXIT       03609600
*                                                                      03609700
         SETLOCK RELEASE,TYPE=LOCAL,REGS=USE,RELATED=ISGNQXIT         03609800
*                                                                      03609900
         BR    R6                      RETURN                         03610200
*                                                                      03650000
MOVE_RNM MVC   RNAME(0),0(R10)                    MOVE RNAME     **     03650200
*                                                                      03650300
         DROP  R12                                                    03650400
         DROP  R9                                                     03650500
*-------------------------------------------------------------------*  03680004
* END RNL SCAN                                                     *  03681004
*-------------------------------------------------------------------*  03682004
*-------------------------------------------------------------------*  03690000
* DATA USED                                                        *  03700000
*-------------------------------------------------------------------*  03710000
         DS    0F                                                     03720000
COMP1    DC   A(X'80000000'+COMPRNL1)                                 03750000
*                                                                      03761000
HRDWNAME DC    CL8'HWRESERV'          MAJOR NAME FOR ALWAYS RESERVE   03761100
*                                                                      03761200
         DS    0F                                                     03762000
ZERO     DC    X'00000000'          CONSTANT                          03769400
*                                                                      03769500
*                                                                      03769600
WORKAREA DSECT                                                        03769700
LOCK_HLD DS    4X'00'               LOCAL AND CMSEQDQ LOCKS HELD ON   03769807
*                                   entry                             03769907
SAVE_REG DS    12X'00'              SAVE REG 1-2-3                    03770000
SAVE_R13 DS    04X'00'              SAVE REG 13                       03770100
SAVEAREA DS    18F'0'               STANDARD SAVE AREA FOR RNL SEARCH 03770200
OFFREG   EQU   *-WORKAREA                                             03770300
QNAME    DS    08C' '               MAJOR NAME                        03770400
RNAME    DS    256C' '              MINOR NAME                        03770500
WORK_LN  EQU   *-WORKAREA                                             03770600
*-------------------------------------------------------------------*  03771000
         DROP  BASEREG                                                03780000
         EJECT                                                        03790000
         PRINT NOGEN                                                  03810000
         ISGRNLE                                                      03811002
         ISGYNQXP                                                     03820000
         ISGGVT                                                       03830000
         CVT  DSECT=YES                                               03840000
         IHAPSA                                                       03850000
         IEFUCBOB DEVCLAS=DA                                          03860000
*                                                                      03870000
         END                                                          03880000
//LINK  EXEC  PGM=IEWL,PARM='LIST,RENT,XREF'                          03890000
//SYSUT1 DD SPACE=(CYL,(1,1)),UNIT=SYSDA                              03900000
```

```
//SYSLMOD DD DSN=SYS1.SANDBOX.LINKLIB,DISP=SHR                          03910000
//SYSPRINT DD SYSOUT=*                                                  03920000
//SYSPUNCH  DD DSN=&OBJ(AUDIT),DISP=(OLD,PASS)                          03930000
//SYSUDUMP  DD SYSOUT=*                                                 03940000
//SYSLIN DD *                                                           03950000
 INCLUDE SYSPUNCH(AUDIT)                                                03960000
 ENTRY ISGCVXIT                                                         03970000
 NAME ISGNQFST(R)                                                       03980008
```

Note: ENTRY ISGCVXIT is mandatory, whereas ISGNQFST can be any name.

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 444.

- *z/OS Version 1 Release 2 Implementation*, SG24-6235
- *OS/390 MVS Parallel Sysplex Capacity Planning*, SG24-4680
- *CICS and VSAM Record-level Sharing: Planning Guide*, SG24-4765
- *CICS and VSAM Record-level Sharing: Implementation Guide*, SG24-4766
- *CICS and VSAM Record-level Sharing: Recovery Considerations*, SG24-4768
- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547
- *z/OS V1R3 DFSMS Technical Guide*, SG24-6569

Other resources

These publications are also relevant as further information sources:

- *z/OS System Management Facilities (SMF)*, SA22-7630
- *z/OS RMF Performance Management Guide*, SC33-7992
- *z/OS MVS System Messages, Volume 10 (IXC-IZP)*, SA22-7640
- *z/OS Auth Assm Services Reference ALE-DYN*, SA22-7609
- *z/OS Auth Assm Services Reference ENF-IXG*, SA22-7610
- *z/OS Auth Assm Services Reference LLA-SDU*, SA22-7611
- *z/OS Installation Exits*, SA22-7593
- *z/OS Planning: Global Resource Serialization*, SA22-7600
- *DFSMShsm Implementation and Customization Guide,* SC35-0418
- *DFSMS z/OS DFSMSdfp Storage Administration Reference*, SC26-4920
- *Managed System Infrastructure for Setup Installation Version 1 Release 4,* SC33-7997
- *Managed System Infrastructure for Operations Setting Up and Using,* SC33-7968
- *z/OS Programming: Resource Recovery*, SA22-7616.
- *z/OS Assembler Services Reference,* SA22-7607
- *z/OS Assembler Services Guide*, SA22-7605
- *z/OS Communication Server IPv6 Network and Application Design Guide,* SC31-8885
- *z/OS DFSMS Migration*, GC26-7398
- *z/OS and z/OS.e Planning for Installation*, GA22-7504
- *z/OS Planning for Installation*, GA22-7504
- *z/OS Migration*, GA22-7580

© Copyright IBM Corp. 2003

- *z/OS Parallel Sysplex Test Report*, SA22-7663
- *z/OS Setting up a Sysplex*, SA22-7625
- *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617
- *z/OS System Commands* SA22-7627
- *z/OS System Messages*, SA22-7640
- *z/OS Security Server LDAP Server Administration and Use*, SC24-5923
- *z/OS Hardware Configuration Definition User's Guide*, SC33-7988
- *z/OS MVS Initialization and Tuning Reference,* SA22-7592
- *z/OS Security Server RACF System Programmer's Guide*, SA22-7681
- *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683
- *z/OS Security Server Network Authentication Service Administration,* SC24-5926.
- *z/OS TSO/E General Information,* SA22-7784
- *z/OS UNIX System Services Planning*, GA22-7800
- *z/OS UNIX System Services Command Reference*, SA22-7802
- *z/OS UNIX System Services Programming Assembler Callable Services Reference*, SA22-7803
- *z/OS Using REXX and z/OS UNIX System Services*, SA22-7806
- *z/OS Distributed File Service zSeries File System Administration*, SC24-5989
- *z/OS JES2 Diagnosis*, GA22-7531
- *z/OS MVS Planning: Workload Management*, SA22-7602-03

Referenced and other relevant Web sites

These Web sites are also relevant as further information sources:

- z/OS marketing and service withdrawal dates:
  http://www.ibm.com/servers/eserver/zseries/zos/support/zos_eos_dates.html/
- Samples of ServerPac Installing Your Order (IYO):
  http://www.ibm.com/servers/eserver/zseries/zos/installation/#pubs
- z/OS marketing and service withdrawal dates:
  http://www.ibm.com/servers/eserver/zseries/zos/support/zos_eos_dates.html/
- z/OS Hot Topics Newsletters:
  http://www.ibm.com/servers/s390/os390/bkserv/hot_topics.html
- List of vendors who support z/OS.e:
  http://www.ibm.com/servers/eserver/zseries/solutions/s390da/r13e.html/
- Multimedia presentation describing z/OS.e:
  http://www.ibm.com/servers/eserver/zseries/zose/
- z/OS.e documentation:
  http://www.ibm.com/servers/eserver/zseries/zose/bkserv
- z800 Software Pricing Configuration Technical Paper:
  http://www.ibm.com/eserver/zseries/library/techpapers/pdf/gm130121.pdf/

- LANRES code is withdrawn in the z/OS V1R4. For migration information, see the white paper at:

  http://www.ibm.com/eserver/zseries/library/whitepapers/gm130035.html/
- General z/OS and z/OS.e information:

  http://www.ibm.com/servers/eserver/zseries/zos/
- General z/OS software prerequisites, coexistence, release migrations, and fallback information:

  http://www.ibm.com/servers/eserver/zseries/zos/bkserv/find_books.html/
- z/OS user group requirements:

  http://www.ibm.com/servers/eserver/zseries/zos/bkserv/user_group_reqs.html/
- z/OS data access and storage management information:

  http://www.storage.ibm.com/software/sms/index.html/
- z/OS Network File Systems (NFS) information:

  http://www.ibm.com/servers/eserver/zseries/zos/nfs/
- z/OS wizard information:

  http://www.ibm.com/eserver/zseries/zos/wizards/
- z/OS and zSeries general Q and A:

  http://www.ibm.com/servers/eserver/zseries/faq/
- z/OS technical documents and flashes:

  http://www.ibm.com/support/techdocs/
- Your customized z/OS (and other) product announcement, IBM eNews electronic newsletters:

  http://isource.ibm.com/
- z/OS migration and installation:

  http://www.ibm.com/servers/eserver/zseries/zos/installation/
- msys for Setup:

  http://www.ibm.com/eserver/zseries/msys
- z/OS Parallel Sysplex CFSizer:

  http://www.ibm.com/servers/eserver/zseries/cfsizer/
- zSeries CF level considerations:

  http://www.ibm.com/servers/eserver/zseries/pso/cftable.html
- IBM Systems Journal Vol. 36, No. 2, 1997: *S/390 Parallel Sysplex Cluster* article about VSAM RLS:

  http://www.research.ibm.com/journal/sj/362/strickland.html
- Enterprise Workload Management, eWLM prototype discussion:

  http://www.research.ibm.com/thinkresearch/pages/2002/20020529_ewlm.shtml
- WLMZOS tool that evaluates the change in execution velocities for your environment when you migrate from OS/390 to z/OS:

  http://www.ibm.com/servers/eserver/zseries/zos/wlm/tools/velocity.html

Archived

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Archived

Index

Symbols

$JDDETAILS command 74
$jddetails command 75
$JDHISTORY command 77
$JDJES command 73
$JDMONITOR command 74
$JDSTATUS command 66, 72
$JSTOP command 79
$PJES2 command 67
*I,MAIN command 55
*I,S command 55, 58
*RETURN command 61
*s dsi command 61
*S JSS command 57
*s,main,flush command 60

A

access control lists 190
accessor environment element 129
ACEE 129
ACL 190
ACL inheritance 130
ACL support 191
activity logging 235
ADDUSER command
  RESTRICTED attribute 123
AES 245
AES algorithm 243
aggrgrow specification 196
AIM 180
AIM stage 3 180
alerts 72
ALLOCAS address space 260
ALLOCDS job 26
ALTUSER command
  RESTRICTED attribute 123
AOFCUST member 27
APAR OW56028 421, 423, 429
Application environments 89
application identity mapping 180
ATS Star 260
AUTOGID keyword 183
AUTOID keyword 183
automatic UID/GID assignment 183
automount command 160
AUTOUID keyword 214

B

base ACL entries 126
binder 3
BPX.SUPERUSER 123, 125, 127
BPX.SUPERUSER profile 25
BPXISETS job 25
BPXMCDS 178
BPXMCDS couple data 10
BPXMCDS couple data set 168, 179
BPXPRMxx parmlib member
  LIMMSG parameter 170
BRLM 178
browser certificates 222
byte range lock manager 178
byte-range locking 179

C

cache
  metadata 203
centralized BRLM 178
certificate life cycle 221
CF CPU capacity 306
CF Request Time Ordering 280
CF workload
  duplexed structures 305
CF-to-CF links 301
checkpoint status record 62
chmod command 125, 142
chmount command 166, 177
Classification rule 89
Coefficients 89
Communications Server NPF 2
confighfs command 172
copytree utility 165
cp command 143
CPC and CFs
  distance between 299
CRQ structure 263
CSFEUTIL option 243
CSVDYNEX macro 408, 428

D

d omvs,f command 197
DB2
  group buffer pools 293
DDF transactions 103
dependent enclave 102
df command 143
DFSMShsm recall activity 275
DFSMStvs 332
digital certificate 221
directory list enhancements 148
distributed BRLM 178–179
DSI 61
DSI processing 58
DSIPARM data set 26
duplicate file system name 202
dynamic aggregate extension 195
Dynamic alias management 89
dynamic alias management 101

Archived

© Copyright IBM Corp. 2003

dynamic system interchange 58, 61

E

effective user ID 149
EIM 209
EIM architecture 210
Enclaves 102
enclaves
    performance block state reporting 113
enqueue promotion 106
enqueue promotion interval 106
Enterprise Identity Mapping Services 209
ESS 100
eWLM 119
execution velocities
    APAR OW55665 118
    WLMZOS tool 119
extended ACL entries 126

F

File Security Packet 191
file security packet 122
file system
    quota 196
find command 143
flush command 60
FMID JMS1743 30
FMID JMSI743 30
FOMISCHO job 25
fsgrow option 196
FSP 122, 191
FSSEC class 128

G

GDPS/PPRC hyperswap function 284
getfacl command 126, 133
getfacl shell command 126
group ownership option 187
GRSCNFxx parmlib member 407, 428
GRSRNLxx parmlib member 408, 428
gskkyman command 244
GSKSRVR started task 246–247
gsktrace command 247

H

hardware management console 23
HBB7705 256
HCD definition
    wlmpav 101
health monitor commands 71
hot start with refresh 58

I

I/O priority management 89, 100
IATUTJCT utility 60
IBM Enterprise Storage Server 100
IBM zSeries 800 22
ICB links 301
ICSF 242
    TSO panels 242
IEAIPSxx parmlib member
    PVLDP keyword 105
IEAOPTxx parmlib member
    ERV value 106
IEASYSxx parmlib member
    IPS 98
    RSVNONR 6, 35
IEFAUTOS function 260
IGDSMSxx PARMLIB member 342
IGDSMSxx parmlib member
    RLS_MaxCfFeatureLevel(A) 265
IKJTSOxx parmlib member
    broadcast keyword 32
    TSO/E broadcast data set 5, 33
IMSC signals 273
independent enclave 102
Install Definition Utility 98
INSTALLED 280
IOEAGFMT format utility 199
IOEAGFMT utility 199
IOEFSPRM configuration file 191, 195, 201, 203
IPSec 220
IPv6 238
IPv6 addresses 246
IRLM 296
IRLM lock structure 298
IRLM locks 105
IRRGMAP class 180
IRRICE report 213
IRRIRA00 conversion utility 180
IRRIRA00 utility 180
IRRPFACL macro 126
IRRUMAP class 180
ISGGREX1 ITSO exit 407, 428
ISGNQXIT exit 402–403, 406, 409, 422, 427, 430
ISGNQXITFAST exit 423
ISGNQXITFAST exit point 421
ISHELL panel
    mounts 175
IWMINSTL 4
IXCL1DSU utility 168, 256
IXLCACHE functions 271

J

JES2 checkpoint lock 69
JES2 coexistence support 29
JES2 health monitor 66
JES2 monitor commands
    RACF profiles 71
JES2 termination 67
JES2MON address space 66
JES3 checkpoint protection 61
JES3 coexistence 28, 63
JES3 DSI processing 61
JES3 main processor enforcement 60
JES3 MAINPROC refresh 50

K
kadmin command 238
Kerberos APIs 238
Kerberos authentication system 238
Kerberos principals 239
key distribution center 239
key rings 245

L
LANRES 2, 443
LDAP directory 209
LDAP password encryption 229
ldapadduuids utility 234
ldapmodrdn client utility 234
ldapsearch 234
LIMMSG parameter 170
link speeds 301
log file
  cache 203
LOGR CDS
  level HBB7705 256
LOGR couple data set
  SMDUPLEX keyword 256
logstream attribute
  updates 251
logstream attribute support 256
logstream attributes 252
ls command 142

M
MAINPROC statement 12, 50
map file
  system symbols 164
master catalog flag 26
metadata backing cache 203
mount command 195–197
MPFLSTxx member 26
MPMSGCLS 29
MQSeries 290
msys for Operations 44, 348
  command dialogs 45
msys for Setup
  coexistence 30
Multiprise 3000 servers 22
multisystem enclave 104
mv command 143

N
NDBM registry 239
NetSEAL 209
not installed 280
notice messages 72
notices 72
NPF 2

O
OMVS restart support 155
OMVS shutdown support 155–156
ONLYAT keyword 186

P
page data set protection 36
Page data sets
  GRS ENQ 37
parallel access volumes 100
Parallel Sysplex enhancements 249
PAV devices 101
pax command 143
pax utility 25
PCI Cryptographic Coprocessor 230
PCICC
  key generation 230
PFS 190
physical file system 190
PKCS #12 support 244
PKI Services 220, 227, 229
Policy Director Authorization Services 209
private keys 245
probe messages 70
probe processing 69
PROGXX parmlib member 408, 428

Q
quota 196

R
RACF
  SEARCH enhancement 212
RACF classes
  FSSEC 127
RACF database
  AIM 180
Redbooks Web site 444
  Contact us xvii
registration support
  OMVS shutdown 155
Report classes 88
RESET command 103
Resource groups 88
RESTFS job 25
RESTRICTED attribute 123
restricted attribute 123
RLS 332
RMF
  CF-to-CF Activity report 305
RNL exclusion list 404, 424
rolling IPL 28
RSVNONR 6

S
SAF registry 239
Scheduling environments 89
SEARCH command 180
sendmail utility 227
Service classes 88

Archived

service definition ID 95
Service policy 88
SET IKJTSO=xx command 5
set ikjtso=xx command 34
setfacl command 125–126, 132–133
setfacl shell command 126
SETGRS command 407, 428
SETOMVS command 166
setomvs command 177
SETPROG EXIT 408, 428
SHARED keyword 183
SHARED.IDS profile 182–183, 213
SIGDANGER signal 155
SIGKILL signal 158
SIGTERM signal 158
SINGSAMP data set 26
SMDUPLEX keyword 256
SMF Unload utility 209
SSL 244
starting duplexing 292
stopping duplexing 295
SYNCHRES option 407, 428
SYS1.BRODCAST data set 32
SYS1.SAMPLIB
  IWMINSTL 91, 94
  IWMSSDEF 92
System Logger
  extended HLQ support 255
System Logger logstream attributes 253
System SSL 245
System SSL enhancements 243
system symbols
  &SYSNAME 164
  &SYSPLEX 164
System-Managed CF Structure Duplexing 290
  CFRM changes 315
  CPU overhead 300
  error recovery 296
  hardware requirements 308
  link speeds 302
  LOGR CDS changes 315
  performance considerations 298
  RMF support 314
  software requirements 309
System-Managed CF Structure Duplexing HCD definitions 314
System-Managed Rebuild 290
system-managed structure duplexing 256
SYSVTOC 409, 430
SYSZVVDS 409, 430

T

tar command 143
temporal affinities
  SDSF DA panel 117
  SMF type 79 118
  vary command 118
Tivoli Policy Director 208
TLS 246
TSO/E broadcast data set 4
TSO/E logon 5
TSO/E MOUNT command 166
TSO/E parmlib list command 5
TSOEXEC program 217
TVS application considerations 270
TVTMAINA 29
TVTMAINJ 29

U

UID/GID
  enhancements 179
UKPT 242
unique key per transaction 242
UNIX System Services 190
UNIXMAP class 180, 212
UNIXPRIV class 127, 182, 188, 190, 193
  SUPERUSER.FILESYS.ACLOVERRIDE profile 124
  SUPERUSER.FILESYS.CHANGEPERMS profile 125
UNIXPRIV profiles (new)
  RESTRICTED.FILESYS.ACCESS 123
  SUPERUSER.FILESYS.ACLOVERRIDE 124
  SUPERUSER.FILESYS.CHANGEPERMS 124
User-Managed Duplexing 293
User-Managed Rebuild 290
USS 190

V

VLF 180
VPN applications 220
VSAM record level sharing 332
VSAM RLS 334
  CF caching enhancements 264
VSAM RLS CF lock structure 266

W

warm start 59
WebSEAL 209
Webserver certificates 223
WLM compatibility mode 3, 88
WLM enqueue management 105–106
WLM goal mode 88
WLMZOS tool 119
Workloads 88

X

XES
  CF requests 294

Z

z/OS DFSMS Transactional VSAM Services (DFSMStvs) 331
z/OS loader 3
z/OS service policy 2
z/OS UNIX ACLs 190
z/OS UNIX couple data set 178
z/OS.e

Archived

release cycle 23
z/OS.e HCD 23
z800 22
z800 server 22
zFS 190
zFS aggregate
formatting 198
zFS commands
RACF authorization 191
zfsadm attach command 195
zfsadm commands 191
zfsadm config 192
zfsadm config command 192, 195–197
zfsadm configquery 193
zfsadm configquery command 197
zfsadm format command 199
zfsadm grow 195
ZOSExxxx 23
zSeries File System 190

Archived

Archived

z/OS Version 1 Release 3 and 4 Implementation

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Archived

Archived

Archived

z/OS Version 1 Release 3 and 4 Implementation

z/OS, z/OS.e, msys for Operations, JES2, JES3, WLM

UNIX System Services, zSeries File System, Parallel Sysplex

RACF, PKI, LDAP, Crypto, CF duplexing

In this IBM Redbook, we highlight many enhancements made in z/OS Version 1 Release 3 and Release 4, and show how to use this document to help you install, tailor, and configure these releases.

First we provide a broad overview of z/OS Version 1 Release 3 and Release 4, and then we discuss how to install and tailor z/OS and the many components that have been enhanced: the z/OS base control program (BCP), Parallel Sysplex and System-Managed CF Structure Duplexing, msys for Operations, Workload Manager (WLM), and zSeries File System (zFS). Security functions such as Security Server (RACF), PKI Services, LDAP server, Network Authentication service and Cryptographic services are also covered.

This redbook is intended for systems programmers and administrators responsible for customizing, installing, and migrating to these newest levels of z/OS.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6581-00 ISBN 0738425680