# Dell EMC GDDR for SRDF/SQAR with AutoSwap

Version 5.2

## Product Guide

REV 11

DELL EMC

Copyright © 2013-2020 Dell Inc. or its subsidiaries. All Rights Reserved.

Published February 2020

Dell believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS-IS”. DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. Published in the USA.

Dell EMC
Hopkinton, Massachusetts 01748-9103
1-508-1000 In North America 1-866-464-7381
www.DellEMC.com

# CONTENTS


Preface

Chapter 1 Introduction to GDDR

What is GDDR? ..... 22
Major GDDR capabilities ..... 23
Types of environment ..... 24
Business continuity configurations ..... 25
SRDF/SQAR with AutoSwap configuration ..... 26
SQAR operational scenarios ..... 33
GDDR fundamentals ..... 41
GDDR sites and regions ..... 41
GDDR-plex ..... 42
GDDR systems ..... 42
GDDR components ..... 43
GDDR user interface ..... 43
GDDRMAIN ..... 43
GDDR scripts ..... 43
GDDR parameters ..... 43
GDDR events ..... 44
GDDR monitors ..... 44
GDDR workers ..... 44
Message interception rules ..... 44
GDDR utilities ..... 44

Chapter 2 GDDR Features and Concepts

GDDR system types ..... 46
C-systems ..... 46
Managed systems ..... 47
GDDR C-system multi-tenancy ..... 51
Key concepts ..... 51
Message affinity ..... 53
GDDR Parameter Wizard ..... 54
Parameters set with GDDR Parameter Wizard ..... 54
GDDR Parameter Wizard prerequisites ..... 54
Intended master C-system ..... 55
Auto-Discovery ..... 55
GDDR Parameter Wizard workflow ..... 55
GDDR environment monitoring ..... 62
GDDR monitors ..... 62
What is a GDDR event ..... 63
Monitored events ..... 63
Message interception ..... 66
Exception notification ..... 66
Expected events ..... 69
Degraded mode ..... 70

GDDR licensing options ..... 71
Supported license types ..... 71

License validation ..... 71
DLm integration using DLMDR ..... 72
SRDF/SQAR High Availability (HA) ..... 74
Support for MSC THAW processing ..... 74
TimeFinder device management ..... 75
What is a BCV in GDDR context ..... 75
GOLD and TEST sets of BCVs ..... 75
TimeFinder device states ..... 76
SnapVX softlinking support ..... 77
Targetless SnapVX support ..... 77
zDP integration ..... 78
GDDR multi-tenancy for TimeFinder ..... 79
GDDR support for external devices ..... 81
BCPii support ..... 82
Introduction to BCPii ..... 82
BCPii HMC networking capabilities and requirements ..... 82
Additional routing capabilities ..... 83
B-systems ..... 83
CPC Recovery—LPAR Recovery ..... 84
GDDR script call overrides ..... 85
GDDR user message automation ..... 89
GDDR user exits ..... 90
GDDRUX01 ..... 90
GDDRUX02 ..... 91
GDDRUX03 ..... 91
GDDRUX04 ..... 92
Built-in routines available to exits ..... 92

Chapter 3 Installing GDDR

Hardware and software requirements ..... 96
Mainframe environment requirements ..... 96
Mainframe hardware and software requirements ..... 97
Dell EMC software requirements ..... 97
DASD requirements ..... 98
Licensing requirements ..... 99
Required installation information ..... 100
GDDR installation sequence ..... 101
Steps to install GDDR ..... 102
Overview ..... 102
Load GDDR*vrm*.XMITFILE to disk ..... 102
Run GDDR*vrm*.XMITLIB(#EXTRACT) ..... 104
Customize RIMLIB JCL ..... 104
Run installation jobs ..... 107
Perform cleanup ..... 108
Apply maintenance updates ..... 108
Post-installation tasks ..... 108

Chapter 4 Integrating GDDR

Introduction ..... 110
Update system parameter files ..... 110
Customize SYS1.PARMLIB(IKJTSOxx) ..... 111
Customize TSO logon ..... 112
APF-authorize LINKLIB ..... 112

Customize LINKLIB and REXX parameter files ..... 112
Configure BCPii ..... 114
BCPii requirements ..... 114
Set up security for BCPii ..... 114
Customize SYS1.PARMLIB for BCPii ..... 116
Specify CPC parameters ..... 116
Create SRDF parameter members (SITxxxxx) ..... 119
Set SCF initialization parameters ..... 121
Authorize ConGroup to use Trip API ..... 122
Configure ConGroup ..... 123
Set up ConGroup automated startup ..... 124
Specify ConGroup STC parameter member ..... 124
Determine which command string to use ..... 125
Configure MSC ..... 126
MSC statement definitions ..... 126
MSC SWITCH command ..... 128
MSC Cleanup utility ..... 128
MSC, ConGroup, and AutoSwap integration ..... 128
Set up GDDR security ..... 130
Define GDDR RACF functional groups ..... 130
Summary of RACF permissions ..... 131
Define GDDR ISPF interface security ..... 133
Authorize RACF for HMC LPAR actions ..... 134
Authorize EMCSAFI security interface ..... 135
Verify module and RACF-protected resource authorization ..... 136
Update ACF2 TSO command list for command limiting ..... 136
Define GDDR datasets ..... 139
Define global variable datasets ..... 139
Define parameter management datasets ..... 139
Define GDDRPARM file ..... 140
Install GDDR started procedures ..... 141
Running GDDR without STEPLIB ..... 142
Dynamic exits ..... 143
Running GDDR with JES3 ..... 143
Install GDDR Licensed Feature Code ..... 144
Customize GDDRMAIN parameters ..... 145
Install GDDRPARM file ..... 145
Verify GDDRPARM file consistency ..... 148
Customize PROCLIB member GDDRPROC ..... 150
Customize GDDR ISPF interface invocation REXX exec ..... 151
Update GDDR ISPF profile ..... 151
Configure GDDR ..... 152
Perform initial parameter activation ..... 152
Customize GDDR parameters ..... 154
Configure optional features ..... 155
Configure GDDR C-System Multi-Tenancy ..... 155
Configure GDDR Multi-Tenancy for TimeFinder ..... 158
Configure GDDR support for external devices ..... 160
Set SRDF recovery preference ..... 160
Configure SRDF/SQAR High Availability (HA) ..... 160
Configure DLm support ..... 160
Modify GDDR user exits ..... 161

## Chapter 5 Running GDDRMAIN


What is GDDRMAIN? ..... 164
Remote command processing ..... 164
GDDRMAIN subtasks ..... 164
GDDRMAIN dependent address spaces ..... 165
Worker task management ..... 165
GDDRMAIN sample JCL ..... 169
GDDR and SCF connectors ..... 169
GDDRMAIN EXEC parameters ..... 171
DeBuG ..... 171
GVB ..... 171
NOEVM ..... 172
NOHBM ..... 172
NOMCS ..... 172
TCPIP ..... 172
VERBose ..... 172
GDDRMAIN console commands ..... 173
Stop command (P) ..... 173
Modify commands (F) summary ..... 173
BC and BR ..... 179
CANCEL ..... 181
CHECKUP ..... 182
COMM ..... 193
CONFIG ..... 194
DLM ..... 198
ENQ ..... 199
GATEK ..... 201
FIXPERT ..... 205
GVB ..... 206
LICENSE ..... 207
LOCK ..... 210
MAINTENANCE ..... 211
MPARM ..... 216
MSGS ..... 217
PARM_REFRESH ..... 218
RDFREFR ..... 219
REGION ..... 220
RELOAD ..... 222
RESTART ..... 224
SCRIPT ..... 225
SET ..... 232
START ..... 234
STOP ..... 234
SUBSYS ..... 235
SUMMARY ..... 237
SVCDUMP ..... 239
SYSTEMS ..... 240
TASKS ..... 242
TOPOLOGY ..... 243
TRACE RESET ..... 245
UMA ..... 246
WORKER ..... 247
GDDRMAIN startup command sequence ..... 249
GDDRMAIN validation checks ..... 251
Cross-address space validation of maintenance level ..... 251

Cross-system validation ..... 253
Validation of multi-instance environment ..... 254
GDDRMAIN supported configuration types ..... 255
GDDRPARM statements ..... 256
Overview ..... 256
CMDQMAXT ..... 258
COMM ..... 259
CPC ..... 261
CSYSSITE ..... 263
DRTCOMM ..... 264
GVDIVDSN ..... 265
MSCGROUP ..... 266
MSG ENABLE|DISABLE ..... 267
MSG ADD|MODIFY ..... 268
MULTI ..... 271
SITE ..... 272
SYMM ..... 273
VCPC ..... 275
WORKER ..... 276
GDDR locks ..... 277
Index lock ..... 277
Update lock ..... 278

Chapter 6 Using GDDR ISPF Interface

Overview ..... 280
Checking local maintenance ..... 280
Selecting GDDR subsystem ..... 280
Checking GDDR maintenance level ..... 281
Primary options ..... 281
ISPF menu path convention ..... 282
ISPF panel titles ..... 282
GDDR dashboard ..... 282
Update GDDR ISPF profile (P) ..... 284
Updating personal GDDR ISPF profile ..... 284
Set up and maintain GDDR environment (M) ..... 286
Changing GDDR automation state ..... 286
Manage GDDR parameters (M,P)—GDDR Parameter Wizard ..... 287
Set message, debug and trace options (M,D) ..... 358
Manage GDDR internal command queue (M,Q) ..... 359
Perform HMC discovery (M,H) ..... 361
Refresh GDDR message table (M,R) ..... 362
Manage GDDR system variables (M,S) ..... 362
Transfer master C-system (M,T) ..... 364
View GDDR configuration (G) ..... 365
Perform GDDR health check (C) ..... 366
Viewing system details ..... 372
Run GDDR scripts (S) ..... 375
View GDDR script statistics (T) ..... 381
Perform GDDR actions (A) ..... 382
Perform HMC discovery (A,H) ..... 382
Perform HMC LPAR actions (A,L) ..... 383
Perform HMC CBU actions (A,CBU) ..... 391
Manage coupled datasets (A,S) ..... 392
Manage CF structures (A,CF) ..... 393

Run Dell EMC Started Task Execution Manager (ST) ..... 395
Adding a new procedure ..... 395
Adding tasks to the procedure ..... 396
Executing a procedure locally ..... 399
Executing a procedure remotely ..... 400
Executing a procedure within a procedure ..... 403

Chapter 7 Using GDDR Utilities

GDDR Command Processor (GDDRMCMD) ..... 410
GDDRMCMD batch interface ..... 410
GDDR Automated Configuration Discovery for DASD (GDDRACDD) ..... 412
GDDR Environment Check utility (GDDRECHK) ..... 442
GDDR SRDF Director Overview utility (GDDRDIRS) ..... 447
GDDR MSC Configuration Validation and Cleanup utility (GDDMSCFX) 450
Requirements ..... 450
Configuration identification ..... 450
Operation modes ..... 451
GDDMSCFX reports ..... 454
Sample JCL ..... 456
EXEC parameters ..... 457
Sample output ..... 459
Return codes ..... 467
GDDR SDDF Session Verification utility (GDDRSDDF) ..... 469
GDDR SRDF Device Status Check Utility (GDDRSC06) ..... 476
GDDR TimeFinder Management utility (GDDRTF20) ..... 478
GDDR BCV Reporting utility (GDDRTF00) ..... 483
BCVGROUP Validation utility (GDDRBCVG) ..... 485
GDDR Invalid Track Monitor utility (GDDRMIN0) ..... 486
GDDR DLm Serial Number utility (GDDRDLSN) ..... 491
GDDR IPL Parameter Swap utility (GDDRISWP) ..... 492
GDDR IPL Assist Monitor utility (GDDRGIAM) ..... 495
Intercepted messages ..... 497
Intercepted WTORs ..... 497
GDDR Workload Assist Monitor utility (GDDRHCMD) ..... 501
GDDR HMC Actions utility (GDDRHMCA) ..... 503
GDDR BCPii Connectivity Test utility (GDDRBCPI) ..... 505
GDDR Load Profile Management utility (GDDRLPRF) ..... 508
GDDR ECGUTIL Driver utility (GDDRECG0) ..... 510
GDDR Expected Events utility (GDDREE00) ..... 511
GDDR DIV Management utility (GDDRGVX) ..... 513
GDDR Command Queue utility (GDDRXCMD) ..... 514
Print current queue ..... 514
Clear current queue ..... 514
GDDRMAIN Trace Print utility (GDDRTRCP) ..... 515

Chapter 8 Running GDDR Scripts

Overview ..... 518
What is a GDDR script ..... 518
Site and region designations ..... 519
Scripts by category ..... 519
Pre-script environment checks ..... 520
Rerunning scripts ..... 523
Running scripts via GDDR ISPF interface ..... 524

Selecting managed systems for test scripts ..... 526
Recovering from script errors..... 527
Rerunning a script ..... 528
WTOR messages ..... 529
Planned scripts ..... 530
Automated Configuration Check - DASD (GDDRPCCD)..... 530
Abandon Secondary/Tertiary/Quaternary Site (DC2/DC3/DC4) (GDDRPA60) ..... 531
Abandon Region RG1 (site swap) (GDD2P17A)..... 532
Restart production at DC3/DC4 after site swap (GDD2P18A)..... 533
Swap production from DC1 to DC2 (GDDRPA42)..... 534
Swap DASD from DC1 to DC2 (GDDRPA21) ..... 536
Test scripts ..... 538
Perform test IPL from BCVs at DC2/DC3/DC4 (GDD2P01A) ..... 538
Resume after test IPL from BCVs at DC2/DC3/DC4 (GDD2P02A).. 538
Perform test IPL from R2s at DC2 (GDD2P03A)..... 539
Perform test IPL from R2s at DC3/DC4 (GDDRPA27)..... 539
Resume after test IPL at DC3/DC4 (GDDRPA28)..... 540
Resume after test IPL from R2s at DC2 (GDD2P16A) ..... 541
Unplanned scripts ..... 542
Recover after unplanned swap (GDDRUP31)..... 542
Recover after loss of DC1 (LDR) (GDD2U13A) ..... 543
Resume replication after unplanned swap (GDDRPA0A) ..... 545
Resume replication after loss of DC1 (GDDRPA0A)..... 546
Resumption scripts ..... 547
Resume SRDF/S replication after ConGroup trip (GDDRPA23) ..... 547
Resume SRDF/A in MSC mode to DC3/DC4 (GDDRPM29) ..... 548
Resume SRDF/A in SQAR mode (GDDRPF29)..... 549
Reclaim Secondary/Tertiary/Quaternary site (DC2/DC3/DC4) (GDDRPA65) ..... 550
RDR scripts..... 552
Options for recovery after loss of primary region ..... 552
Recover at DC3/DC4 after RDR in primary region/Recover at DC3/DC4 with SRDF/S to DC4/DC3 (GDDRPA05) ..... 552
Restart production at DC3/DC4 with cascaded/concurrent SRDF (GDDRPA06) ..... 553
Recover at DC3 with cascaded SRDF DC3->DC4, DC4->DC2/Recover at DC4 with concurrent SRDF DC4->DC3, DC4->DC2 (GDDRPA07)..... 555
Special scripts..... 557
Planned STAR-HA Takeover (GDDRPA36) ..... 557
Unplanned STAR-HA Takeover (GDDRUP35) ..... 557
Transfer AutoSwap Owner to DCn (GDDRPXAS) ..... 557
Restart primary MSC server at DCn (GDDRMMSC) ..... 557
Restart secondary MSC server at DCn (GDDRPA36)..... 558
Transfer Master C System to DCn (GDDRPXMC) ..... 558
Global variable backup (GDDRPGVB)..... 558
Move systems to alternate CPC (GDDRMCPC)..... 559

Chapter 9 Handling Unplanned Events

Overview..... 562
Consistency group trip ..... 562
Unplanned swap..... 562
GDDR AutoSwap and restart of Enhanced Catalog Sharing ..... 563

Local and regional disaster ..... 564
Confirm loss of primary region ..... 564
Confirm ready for recovery at secondary region ..... 565
Loss of AutoSwap owner ..... 566
Lost owner policy ..... 566
Lost owner policy operations ..... 566
System failure ..... 567
C-system failure ..... 567
Managed system failure ..... 567
GDDR master function transfer ..... 570

Chapter 10 Maintaining GDDR Environment

Updating GDDR licenses ..... 572
Updating GDDRPARM file statements ..... 573
Changing SYMM parameters ..... 574
Setting up a new C-system ..... 575
Renaming an existing C-system ..... 576
Changing C-system or managed system IP address ..... 577
Changing C-system or managed system IP port ..... 578
Adding a new system or sysplex to GDDR ..... 579
Changing the consistency group name ..... 580
Adding new SRDF groups to GDDR ..... 581
Adding new devices to GDDR ..... 583
Removing an SRDF group from GDDR control ..... 585
Removing devices from GDDR control ..... 586
Removing a system or a sysplex from GDDR ..... 587
Changing global variable DIV dataset or WORKER parameters ..... 588
Increasing GDDR DIV size ..... 589
Restoring normal operations after MSC THAW ..... 590
Handling special types of datasets ..... 591
Page datasets ..... 591
Coupled datasets ..... 591
Standalone dump considerations ..... 591

Chapter 11 Troubleshooting

Detecting and resolving problems ..... 594
Tracing and debugging ..... 594
Using GDDRXCMD batch utility ..... 594
Verifying maintenance level ..... 595

# FIGURES


1 SRDF/SQAR with AutoSwap environment ..... 27
2 Physical designations in SRDF/SQAR topology ..... 29
3 Logical designations in SRDF/SQAR topology ..... 29
4 Storage system and SRDF group asymmetries ..... 30
5 SRDF/SQAR operations and SDDF session management ..... 31
6 SRDF/SQAR initial state ..... 34
7 SRDF/SQAR inter-region swap with primary site at DC3 ..... 35
8 SRDF/SQAR inter-region swap with primary site at DC4 ..... 35
9 SRDF/SQAR intra-region swap ..... 36
10 SRDF/SQAR unplanned swap from Site A to Site B ..... 37
11 SRDF/SQAR Site A failure with recovery at Site B ..... 38
12 SRDF/SQAR Site A failure with recovery at Site C ..... 38
13 SRDF/SQAR Site A failure with recovery at Site D ..... 39
14 SRDF/SQAR primary region failure ..... 39
15 SRDF/SQAR region 1 workload failure ..... 40
16 GDDR parameter management ..... 57
17 GDDR Parameter Wizard workflow—step 1 ..... 58
18 GDDR Parameter Wizard workflow—step 2 ..... 58
19 GDDR Parameter Wizard workflow—step 3 ..... 59
20 GDDR Parameter Wizard workflow—step 4 ..... 59
21 GDDR Parameter Wizard workflow—step 5 ..... 60
22 GDDR Parameter Wizard workflow—step 6 ..... 61
23 DLm integration using DLMDR ..... 72
24 Member selection list ..... 105
25 Dell EMC JCL Customization Utility ..... 105
26 Dell EMC JCL Customization Utility completed panel ..... 107
27 MSC servers in SRDF/SQAR mode ..... 126
28 Select Parameter Input Dataset panel during initial parameter activation ..... 153
29 Prepare Work Dataset - Status panel, input variables ..... 154
30 Prepare Work Dataset - Status panel, work members ..... 154
31 Refreshing local Maintenance info popup ..... 280
32 Select GDDR Subsystem panel ..... 280
33 Maintenance level mismatch detected ..... 281
34 Primary Options Menu panel ..... 281
35 GDDR dashboard ..... 282
36 Change GDDR ISPF Profile Variable Values panel (P) ..... 284
37 Setup and Maintenance Menu panel (M) ..... 286
38 Parameter Management Options Menu panel (M,P) ..... 287
39 Manage GDDR Parameter Backups panel (M,P,B) ..... 290
40 Select Dataset for GDDR Parameter Backup panel ..... 292
41 GDDRACDD - Automated Configuration Discovery for DASD panel (M,P,G) ..... 293
42 Dataset name to Backup panel ..... 294
43 Select Parameter Input Dataset panel (M,P,I) ..... 295
44 Prepare Work Dataset for Parameter Load confirmation panel ..... 297
45 Prepare Work Dataset status panels ..... 297
46 GDDI010I message - User 1 ..... 298
47 Parameter Edit Warning panel - User 2 ..... 298
48 Message GDDI010E in GDDR - Define Configuration Basics panel after FORCE of Edit-in-Progress serialization lock - User 1 ..... 299
49 Message GDDI010E in Parameter Management Options Menu panel after FORCE of Ed-

| | | |
|---|---|---|
| | it-in-Progress serialization lock - User 1 | 299 |
| 50 | Define Configuration Basics panel (M,P,C) | 300 |
| 51 | Prompt to save unsaved changes | 301 |
| 52 | Define GDDR Configuration features panel (M,P,C,F) | 302 |
| 53 | Define C-Systems panel (M,P,C,C) | 304 |
| 54 | Define GDDR Datasets panel (M,P,C,D) | 307 |
| 55 | Define Site roles and groups panel (M,P,C,R) | 308 |
| 56 | Define Data Storage Objects panel (M,P,D) | 309 |
| 57 | Define SRDF Device Ranges panel (M,P,D,S) | 310 |
| 58 | Define TimeFinder Device Ranges panel (M,P,D,T) | 312 |
| 59 | Define GDDR Snapshot Names panel (M,P,D,N) | 313 |
| 60 | Define GDDR Managed VDG Names panel (M,P,D,V) | 313 |
| 61 | Define SDDF GateKeepers panel (M,P,D,G) | 314 |
| 62 | Define DLm Systems panel (M,P,D,D) | 314 |
| 63 | ACP Details panel | 316 |
| 64 | DLm QUIESCE and VARY panel | 316 |
| 65 | Define Host Objects panel (M,P,H) | 317 |
| 66 | Define Managed Systems panel (M,P,H,S) | 318 |
| 67 | Define System IPL Priorities panel (M,P,H,SP) | 320 |
| 68 | Define Managed LPARs panel (M,P,H,L) | 323 |
| 69 | Define System Recovery Attributes panel (M,P,H,R) | 325 |
| 70 | Define Managed CPCs panel (M,P,H,P) | 326 |
| 71 | Define IPL Parameters panel (M,P,H,I) | 328 |
| 72 | Define HMC Load Activation Profiles panel (M,P,H,A) | 331 |
| 73 | Define Managed Couple Datasets (1/2) panel (M,P,H,D) | 332 |
| 74 | Define Managed Couple Datasets (2/2) panel (M,P,H,D) | 333 |
| 75 | Define Managed CF Structures panel (M,P,H,CF) | 335 |
| 76 | Define External Workloads panel (M,P,H,W) | 336 |
| 77 | Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E) | 337 |
| 78 | Specify GDDR Options panel (M,P,O) | 339 |
| 79 | Specify Default Script Call Overrides panel (M,P,O,O) | 340 |
| 80 | Script Sysplex Options panel (M,P,O,S) | 341 |
| 81 | Script JCL Parameters panel (M,P,O,J) | 344 |
| 82 | Utility Parameters panel (M,P,O,U) | 346 |
| 83 | Specify GDDR Tuning Values panel (M,P,O,T) | 347 |
| 84 | Define GDDR User Labels panel (M,P,O,L) | 351 |
| 85 | Validate GDDR Parameter Set panel (M,P,V) | 353 |
| 86 | Activate GDDR Parameter Set panel (M,P,A) | 355 |
| 87 | Parameter Load Activation status panel 1 of 2 | 357 |
| 88 | Parameter Load Activation status panel 2 of 2 | 357 |
| 89 | Set Output Message Levels by Program panel (M,D) | 358 |
| 90 | Add Program to MsgLevel/Debug/Trace List panel | 359 |
| 91 | Manage Internal Command Queue panel (M,Q) when no script is running | 359 |
| 92 | Manage Internal Command Queue panel (M,Q) when a script is running | 360 |
| 93 | Discovering HMC Objects panel | 361 |
| 94 | HMC Discovery Results panel (M,H) | 361 |
| 95 | Manage GDDR System Variables panel (M,S) | 363 |
| 96 | Transfer Master C-System panel (M,T) | 364 |
| 97 | View GDDR Configuration panel (G) | 365 |
| 98 | Perform Health Check panel (C) | 366 |
| 99 | GDDRMAIN System Details panel | 373 |
| 100 | Select Script to Run panel (S) | 375 |
| 101 | Select Script to Run panel (S), master C-system at Secondary Region, 1 of 3 | 376 |

102 Select Script to Run panel (S), master C-system at Secondary Region, 2 of 3 ..... 377
103 Select Script to Run panel (S), master C-system at Secondary Region, 3 of 3 ..... 377
104 Script Details panel ..... 380
105 GDDR Operation - Script Selection for Status panel (T) ..... 381
106 View Script Step Statistics panel ..... 381
107 Actions Menu panel (A) ..... 382
108 Discovering HMC Objects panel ..... 382
109 HMC Discovery Results panel (A,H) ..... 383
110 Perform HMC LPAR Actions panel (A,L) ..... 383
111 Perform CBU Actions panel (A,CBU) ..... 391
112 Manage Couple Datasets panel (A,S) ..... 392
113 Manage CF Structures panel (A,CF) ..... 393
114 Dell EMC Started Task Execution Manager panel (ST)—Procedure Member List . 395
115 Dell EMC Started Task Execution Manager—Procedure Add/Update panel ..... 395
116 EMC Started Task Execution Manager—Procedure Member List panel ..... 396
117 Dell EMC Started Task Execution Manager—blank task list ..... 396
118 Dell EMC Started Task Execution Manager—Task Add/Update panel, blank ..... 397
119 Dell EMC Started Task Execution Manager—Task Add/Update panel, completed 398
120 Dell EMC Started Task Execution Manager—task list, completed ..... 399
121 Dell EMC Started Task Execution Manager—procedure JFPSCFD selected ..... 399
122 Dell EMC Started Task Execution Manager—procedure ADUP selected for remote execution ..... 401
123 Dell EMC Started Task Execution Manager—Host List for procedure panel ..... 402
124 Dell EMC Started Task Execution Manager—Remote Execution panel ..... 402
125 Enter Remote User Information pop-up ..... 403
126 SRDF/SQAR GDDRACDD parameter example ..... 431
127 GDDR SRDF directors overview for all SRDF groups ..... 448
128 SRDF groups by director by storage system by site, Site: DC1 ..... 448
129 SRDF groups by director by storage system by site, Site UNK ..... 449
130 Specify Parameters for Initial Script Run panel ..... 524
131 Specify Call Overrides for Script panel ..... 525
132 Job submission confirmation panel ..... 526
133 Select System for DCn Restart panel ..... 526
134 Select System for DCn Reset panel ..... 527

# TABLES


| | | |
|---|---|---|
| 1 | Monitored events | 63 |
| 2 | Software state analysis messages | 67 |
| 3 | Expected events | 70 |
| 4 | Supported TimeFinder device states | 76 |
| 5 | GDDR call overrides | 85 |
| 6 | Mainframe hardware and software requirements | 97 |
| 7 | Dell EMC software requirements | 97 |
| 8 | DASD requirements | 98 |
| 9 | RIMLIB library contents | 104 |
| 10 | SRDF parameter members | 119 |
| 11 | RACF functional groups | 130 |
| 12 | RACF permissions | 131 |
| 13 | RACF permissions, OPERCMDS class | 132 |
| 14 | Summary of GDDR ISPF RACF permissions | 133 |
| 15 | SAMPLIB security members | 136 |
| 16 | GDDRMAIN subtasks | 164 |
| 17 | GDDRMAIN dependent address spaces | 165 |
| 18 | Worker task names | 165 |
| 19 | Default MIN and MAX limits | 166 |
| 20 | Subtask management commands | 173 |
| 21 | Miscellaneous GDDRMAIN console commands | 174 |
| 22 | GDDRMCMD commands: configuration information | 176 |
| 23 | GDDRMCMD commands: GDDR-plex information | 177 |
| 24 | GDDRMCMD commands: local system information | 177 |
| 25 | GDDRMCMD commands: address space information | 178 |
| 26 | Utility GDDRMCMD commands | 178 |
| 27 | Reread and processed GDDRPARM statements | 224 |
| 28 | GDDRMAIN supported configuration types | 255 |
| 29 | GDDRPARM statements | 256 |
| 30 | Possible lock states | 277 |
| 31 | TimeFinder and zDP field value combinations | 304 |
| 32 | Defining managed coupled datasets | 334 |
| 33 | Mainframe Enablers STC requirements for managed systems | 338 |
| 34 | Mainframe Enablers STC requirements for C-systems | 339 |
| 35 | Script management options | 343 |
| 36 | Script generation status messages | 378 |
| 37 | GDDRMCMD return codes | 411 |
| 38 | GDDRTF20 operations and TimeFinder commands | 479 |
| 39 | GDDRBCVG return codes | 485 |
| 40 | GDDRMIN0 return codes | 490 |
| 41 | GDDRISWP return codes | 494 |
| 42 | GDDRHCMD return codes | 502 |
| 43 | GDDRHMCA return codes | 504 |
| 44 | GDDREE00 return codes | 512 |
| 45 | Resiliency Expert global variables | 527 |
| 46 | Actions to refresh GDDRPARM settings | 573 |

# PREFACE

*As part of an effort to improve its product lines, Dell EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.*

*Contact your Dell EMC representative if a product does not function properly or does not function as described in this document.*

---

**Note:** This document was accurate at publication time. New versions of this document might be released in Dell EMC Online Support. Check Dell EMC Online Support to ensure that you are using the latest version of this document.

---

## Purpose

This guide describes the basic concepts of Dell EMC Geographically Dispersed Disaster Restart (GDDR), how to install it, and how to implement its major features and facilities.

## Coverage

This guide describes the product features available with the latest maintenance applied.

## Audience

This document is part of the GDDR documentation set, and is intended for use by GDDR systems administrators and computer operators.

Readers of this document are expected to be familiar with the following topics:

- IBM z/OS operating environments
- IBM parallel sysplex
- IBM DFSMS
- The Dell EMC Mainframe Enablers software suite, including ResourcePak Base, SRDF Host Component, Multi-Session Consistency (MSC), AutoSwap, Consistency Groups (ConGroup)
- Dell EMC Disk Library for mainframe (DLm)

## Related documentation

The following publications provide additional information:

- *GDDR Release Notes*
- *GDDR Message Guide*
- *Mainframe Enablers Installation and Customization Guide*
- *ResourcePak Base for z/OS Product Guide*

- *SRDF Host Component for z/OS Product Guide*
- *AutoSwap for z/OS Product Guide*
- *Consistency Groups for z/OS Product Guide*
- *TimeFinder SnapVX and zDP Product Guide*
- *TimeFinder/Clone Mainframe Snap Facility Product Guide*
- *TimeFinder/Mirror for z/OS Product Guide*
- *Disk Library for mainframe User Guide*

## Conventions used in this document

Dell EMC uses the following type style conventions in this document:

| | |
|---|---|
| *Italic* | Used for:<br>• Titles of publications referenced in text<br>• Emphasis, for example, a new term |
| `Courier` | Used for:<br>• Command syntax and parameters<br>• System output, such as messages |
| **`Courier bold`** | Used for user input, for example: Reply **CONT**. |
| *`Courier italic`* | Used for variables in command/parameter syntax and messages, for example: `DISPLAY ccuu` |
| `Courier underline` | Underline indicates the default value, for example: `YES\|NO` |
| < > | Angle brackets enclose variables or explanatory text when it includes multiple words, for example: *<list of device numbers>* |
| [ ] | Square brackets enclose optional values, for example: `DISPLAY [DETAIL]` |
| \| | Vertical bar indicates alternate selections (the bar means “or”), for example: `RUN\|NORUN` |
| { } | Braces are used together with the vertical bar (\|) to indicate the start and end of alternate selections, for example: `{DEV symdv#\|CUU ccuu}` |
| ... | Ellipses indicate nonessential information omitted from the example |

## Where to get help

### Product information

For information about Dell EMC products, licensing, and service, go to www.dellemc.com (registration required).

### Technical support

To access the Dell EMC Online Support web site, go to www.dell.com/support and search for your product. You will be redirected to the product support page, offering quick links to Documentation, Downloads, Advisories, and Knowledgebase for your

product. The product support page also provides a link to the Service Center where you can create a service request, manage your service requests, and contact Dell EMC Customer Support either through Dell EMC Live Chat or using other options.

**Note:** To open a service request through Dell EMC Online Support, you must have a valid support agreement. Contact the Dell EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

### eLicensing support

To activate your entitlements and obtain your license files, visit the Service Center as directed on your License Authorization Code (LAC) letter emailed to you.

For help with missing or incorrect entitlements after activation (that is, expected functionality remains unavailable because it is not licensed), contact your Dell EMC Account Representative or Authorized Reseller.

For help with any errors applying license files through Solutions Enabler, contact Dell EMC Customer Support.

If you are missing a LAC letter, or require further instructions on activating your licenses through the Online Support site, contact Dell EMC worldwide licensing team at licensing@emc.com or call:

- North America, Latin America, APJK, Australia, New Zealand: SVC4EMC (800-782-4362) and follow the voice prompts.
- EMEA: +353 (0) 21 4879862 and follow the voice prompts.

## Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to:

VMAXContentFeedback@emc.com

# CHAPTER 1
# Introduction to GDDR

This chapter includes the following topics:


- What is GDDR? ..... 22
- Business continuity configurations ..... 25
- GDDR fundamentals ..... 41
- GDDR components ..... 43

# What is GDDR?

Dell EMC Geographically Dispersed Disaster Restart (GDDR) automates business recovery following both planned outages and disaster situations, including the total loss of a data center. GDDR achieves this goal by providing monitoring, automation, and quality controls to many Dell EMC and third-party hardware and software products required for business restart.

Because GDDR restarts managed systems following disasters, it does not reside on the same z/OS systems that it is seeking to protect. GDDR resides in separate logical partitions (LPARs) from the host z/OS systems that run application workloads.

GDDR is installed on a control z/OS system at each site. Each GDDR node is aware of the other GDDR nodes through network connections between the sites. This awareness allows GDDR to:

- Detect disasters
- Identify survivors
- Nominate the leader
- Recover business at one of the surviving sites

To achieve the task of business restart, GDDR automation extends well beyond the disk level and into the host operating system level. At this level controls and access to third-party software and hardware products are implemented to enable Dell EMC to perform automated recovery.

GDDR main activities include:

- Managing planned site swaps (workload DASD and tape) between the primary and secondary sites and recovering the SRDF/SQAR with AutoSwap environment
- Managing planned site swaps (DASD and tape) between the primary and secondary sites and recovering the SRDF/SQAR with AutoSwap environment
- Managing recovery from a single or dual planned or unplanned site outage in one region, with local SRDF/S protection established differentially between the recovery sites in another region. Out of region recovery is supported with cascaded or concurrent SRDF to an available out of region site.
- Managing the recovery of the SRDF environment and restarting SRDF/A (asynchronous remote replication) in the event of an unplanned site swap
- Active monitoring of the managed environment and responding to exception conditions
- Resetting/IPLing z/OS systems at the remote site
- Testing disaster recovery from BCVs at the remote site
- Testing disaster recovery from R2 at the remote site

## Major GDDR capabilities

GDDR provides the following major capabilities:

- Situational awareness
- Survivor recognition
- Leadership nomination
- Restart coordination
- Additional capabilities

### Situational awareness

GDDR can distinguish normal operational disruptions, such as network outages (SRDF link drop), from real disasters and respond accordingly. This awareness is achieved by periodic exchange of dual-direction heartbeats between the GDDR control systems.

### Survivor recognition

GDDR can determine which sites and systems have survived a disaster. GDDR has built-in intelligence to monitor other GDDR systems. GDDR continually checks for disaster situations and ensures that other GDDR systems are "healthy." This allows GDDR to recognize and act on potential disaster situations, even if only one GDDR system survives.

"Split brain" problems associated with cluster technologies are avoided through operator prompts. Upon the initial recognition stage, GDDR issues messages to the operator console seeking confirmation of the event and required restart actions.

### Leadership nomination

If a local or regional disaster occurs, GDDR can determine which of the surviving sites will execute the recovery.

If the site of the master control system is destroyed or the master control system fails, one of the surviving control systems assumes the role of the master control system.

### Restart coordination

Restart procedures following disasters are coordinated from the master control system. GDDR coordinates and executes predetermined processes to:

- Restart the business at the desired surviving site in the event of a disaster
- Automate a planned site swap

### Additional capabilities

As part of the planned site swap process and the recovery process after an unplanned site swap, GDDR can optionally perform the following tasks:

- Trigger stop or start of distributed workloads
- Trigger stop or start of z/OS workloads in multiple sysplexes in parallel
- Manage DLm configurations using DLMDR

## Types of environment

GDDR can manage environments with the following elements:

- Multiple z/OS systems
- Multiple sysplexes

  GDDR is able to control multiple sysplexes from a single GDDR control system.
- Multiple PowerMax/VMAX systems
- Intermix of CKD and FBA/FBAM DASD and BCVs, as well as DLm

# Business continuity configurations

GDDR is available in the following configurations:

**Note:** "GDDR sites and regions" on page 41 explains the DC1, DC2, DC3, and DC4 abbreviations.

- SRDF/S with ConGroup

  The 2-site SRDF/S with ConGroup configuration provides disaster restart capabilities at site DC2.

- SRDF/S with AutoSwap

  The 2-site SRDF/S with AutoSwap configuration provides for near-continuous availability through device failover between DC1 and DC2.

- SRDF/A

  The 2-site SRDF/A configuration provides disaster restart capabilities at site DC3.

- SRDF/Star

  The 3-site SRDF/Star configuration provides disaster restart capabilities at either DC2 or DC3. Concurrent and cascaded SRDF support further minimize the DC3 recovery time objective. The 2-site SRDF/Star configuration has DASD at 3 sites in a concurrent topology, but has no control system at DC2, and provides disaster restart at DC3 only.

  **Note:** All SRDF/Star configurations must have R22 devices at DC3.

- SRDF/Star with AutoSwap

  The 3-site SRDF/Star with AutoSwap configuration provides for near-continuous availability through device failover between DC1 and DC2 as well as disaster restart capabilities at DC3. Concurrent and cascaded SRDF support further minimizes the DC3 recovery time objective.

  **Note:** All 3-site SRDF/Star configurations, with or without AutoSwap, can be dynamically reconfigured back and forth between concurrent and cascaded data flow.

- SRDF/Star-A

  The 3-site SRDF/Star-A (Asynchronous) configuration provides replication for two asynchronous SRDF groups from an R11 source device, with the ability to perform differential resynchronization between the two SRDF/A target devices.

  GDDR provides recovery after loss of Site A. The recovery includes automation to restart the business at either of the surviving sites, preservation of the most recent data available at those sites, and differential resynchronization between the sites.

- SRDF/SQAR with AutoSwap

  The 4-site SRDF/SQAR with AutoSwap configuration provides for near-continuous availability through device failover between DC1 and DC2, within region 1; as well as disaster restart capabilities at region 2 with DC3 and DC4 located at an

extended geographical distance away from region 1. Concurrent or cascaded SRDF replication protects data originating from the recovery site after a primary region outage.

GDDR can be customized to operate in any of the supported configurations. During GDDR implementation, the parameter library that controls the GDDR functionality is customized to reflect the prerequisite software stack and the required data center topology. The configuration choice is made by selecting site names that are used in GDDRPARM and is further refined using the GDDR Parameter Wizard.

This document discusses the GDDR SRDF/SQAR with AutoSwap configuration. Documentation for other GDDR configurations is available at the Dell EMC Online Support website.

## SRDF/SQAR with AutoSwap configuration

SRDF/SQAR (Symmetrix Quadrilateral Asynchronous Replication) is a four-site implementation of SRDF/S and SRDF/A that enables differential resynchronization between sites along the perimeter of a 'square' multi-site SRDF topology. Its principle objective is to meet the requirement for a multi-region high availability and disaster restart solution.

GDDR support for the SRDF/SQAR configuration provides the ability to recover from a single or dual unplanned site outage in one region, with local SRDF/S protection established differentially between the recovery sites in another region. This enables you to meet the requirement of quickly resuming a workload with SRDF/S and AutoSwap protection in another region. In certain failure scenarios it also provides 'zero data loss' disaster recovery across regions.

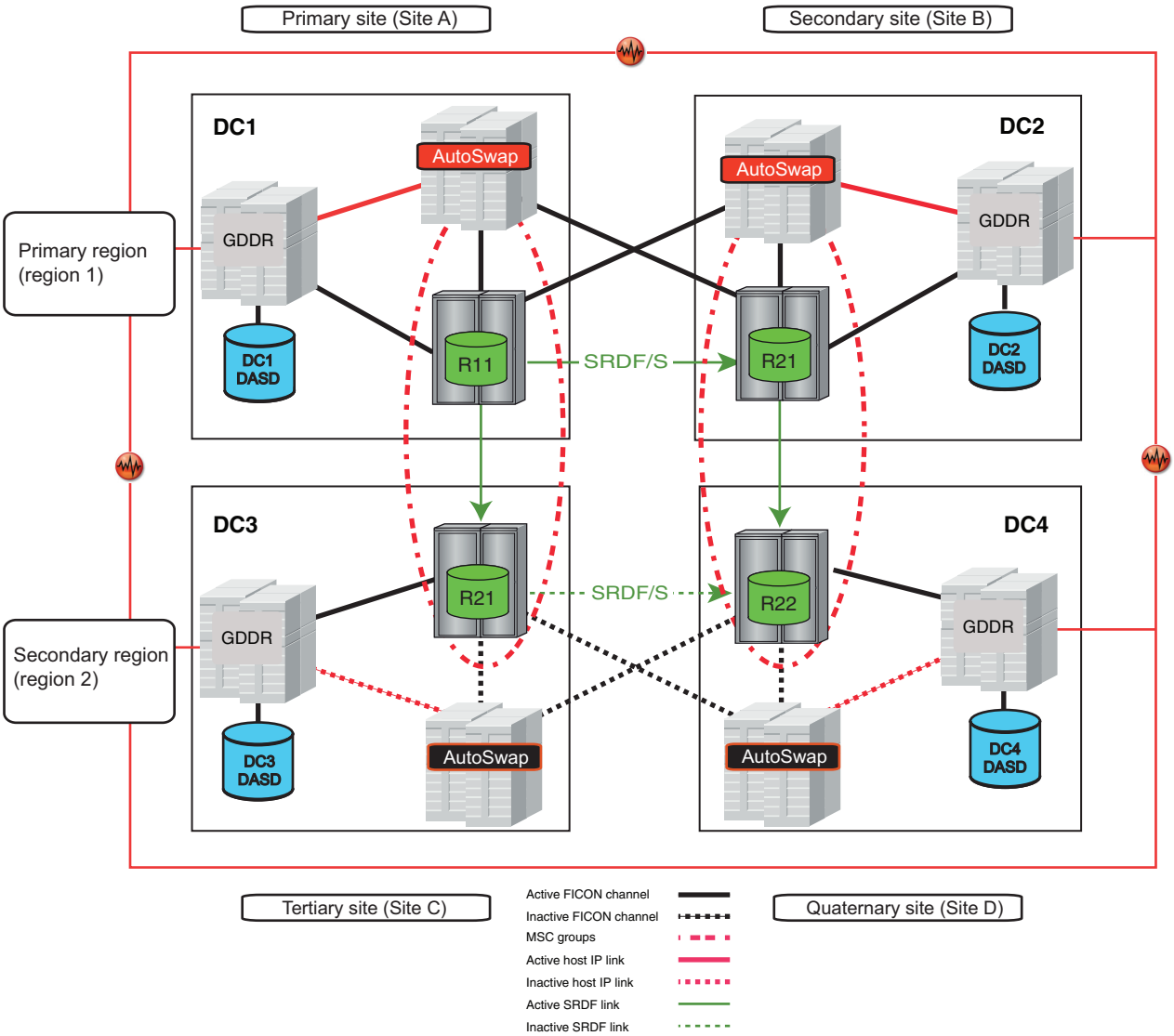



**Figure 1** SRDF/SQAR with AutoSwap environment

**Note:** The SRDF/SQAR with AutoSwap environment can optionally include DLm, as described in "DLm integration using DLMDR" on page 72. The GDDR environment with DLm and PowerMax or VMAX backend is shown in Figure 23 on page 72.

Figure 1 shows the four control systems with their independent heartbeat communication paths, separate from the production disk and computer facilities. Each of the managed z/OS systems has Dell EMC AutoSwap and Dell EMC Consistency Groups (ConGroup) installed.

Each GDDR SRDF/SQAR environment manages two consistency groups (one active, one defined) and two Multi-Session Consistency (MSC) groups (both active). A consistency group is a named group of source (R1) volumes managed by the ConGroup application as a unit. An MSC group is a named group, consisting of multiple SRDF groups operating in SRDF/A mode, managed by the Dell EMC MSC control software feature as a single unit. The relationship between Site A (DC1) and Site B (DC2) is maintained through SRDF/S (synchronous) replication of primary disk images at DC1 to DC2, while SRDF/A (asynchronous) replication maintains out of region mirrored data at Site C (DC3) and Site D (DC4).

## Requirements and restrictions

- SRDF/SQAR is required to be configured with the MSC High Availability feature with a second SCF instance and MSC configured using weight factor =2.
- SRDF Host Component actions which change the devices defined to SQAR MSC groups require the MSC tasks to be down at the time of the change.
- Connectivity is provided only along the perimeter of the SQAR topology. Cross-site connectivity (A>D, B>C) is not supported.
- Differential resynchronization is only allowed along the perimeter of the SQAR topology. A>D and B>C connectivity is not supported. Therefore, traditional 3-site SRDF/Star as a recovery configuration is not supported.
- For a single site failure, it is important to know which SRDF/A site is more current. The existing SRDF/A secondary time-of-day value is used to determine which site is ahead.

## Physical and logical designations

Figure 2 illustrates the designations used to describe the physical configuration of an SRDF/SQAR topology. These designations do not change as a result of planned or unplanned events. The SRDF/SQAR configurationfeatures two *regions* which are interconnected using SRDF/A and are typically separated by a large distance. Within each region, there are two sites that are connected via SRDF/S intra-region.

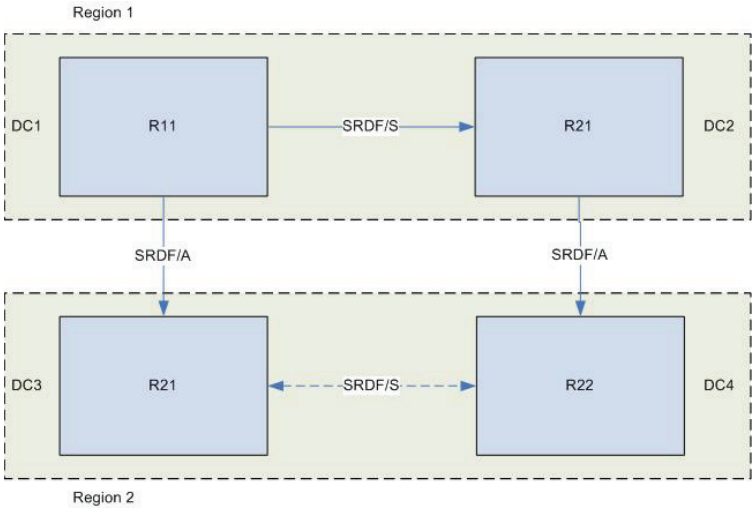



**Figure 2** Physical designations in SRDF/SQAR topology

Figure 3 illustrates the designations that are used to describe the SRDF/SQAR logical environment. Sites can change roles during planned or unplanned site and /or region swaps. The role of a site is described as *primary*, *secondary*, *tertiary*, or *quaternary*, referenced using the corresponding letters A, B, C, and D respectively.

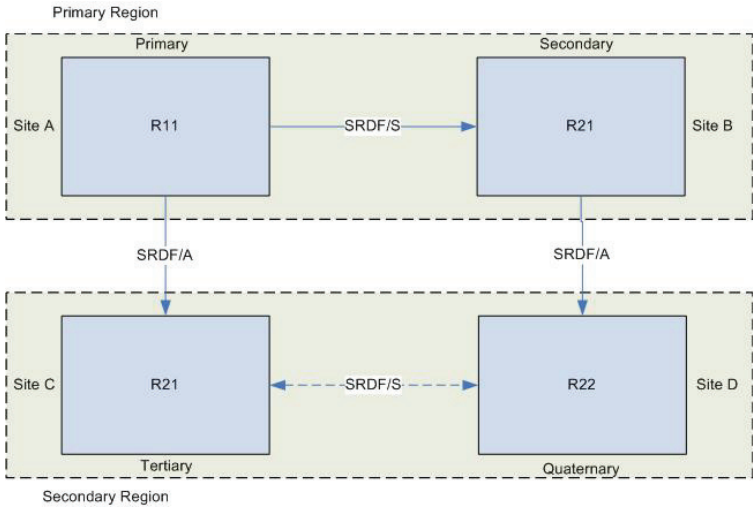



**Figure 3** Logical designations in SRDF/SQAR topology

For example, if Site A experienced an outage, recovery can occur at Site B within the primary region or at either Site C or Site D in the secondary region, using data current to the point of disaster from Site B. If a regional outage occurred, recovery can occur at Site C or D. In this example, the primary site moves to one of the Region 2 sites and becomes Site A.

## Configuration rules

### Storage systems and SRDF groups

As shown in Figure 4, it is permissible to have varying numbers of storage systems connecting the sites. However, the number of asynchronous (SRDF/A) groups configured for the two SQAR groups must be equal. Also, the number of SYNC and recovery groups cannot be more than the number of asynchronous groups.

Only one SQAR configuration (two related MSC SQAR groups) is allowed in a single MSC environment.

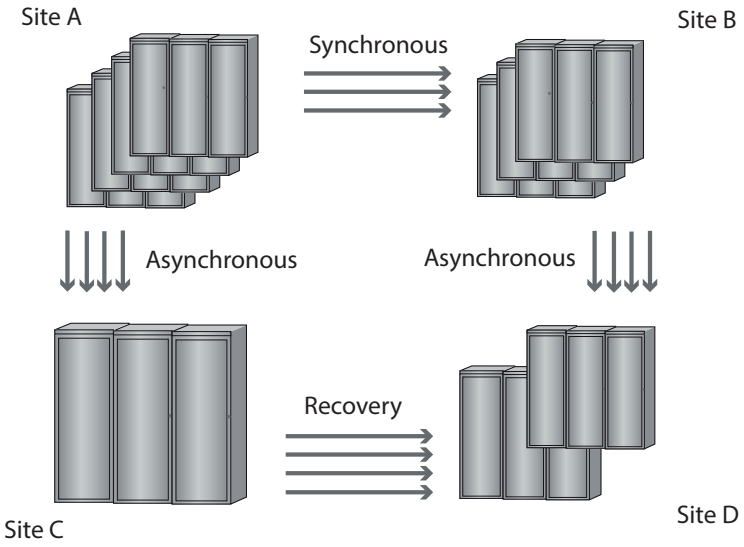



**Figure 4** Storage system and SRDF group asymmetries

### Devices

All devices protected in SQAR mode must be replicated DC1->DC2, DC1->DC3, DC2->DC4, and DC3->DC4. You cannot have more devices replicated with SRDF/S than with SRDF/A, or more devices replicated in SRDF/A than with SRDF/S.

GDDR supports additional devices replicating on ADCOPY-DISK mode on the DC1-DC3 and DC2-DC4 legs of the configuration. These devices must be configured in separate SRDF groups.

## Normal operation

Figure 5 illustrates SRDF/SQAR operations.

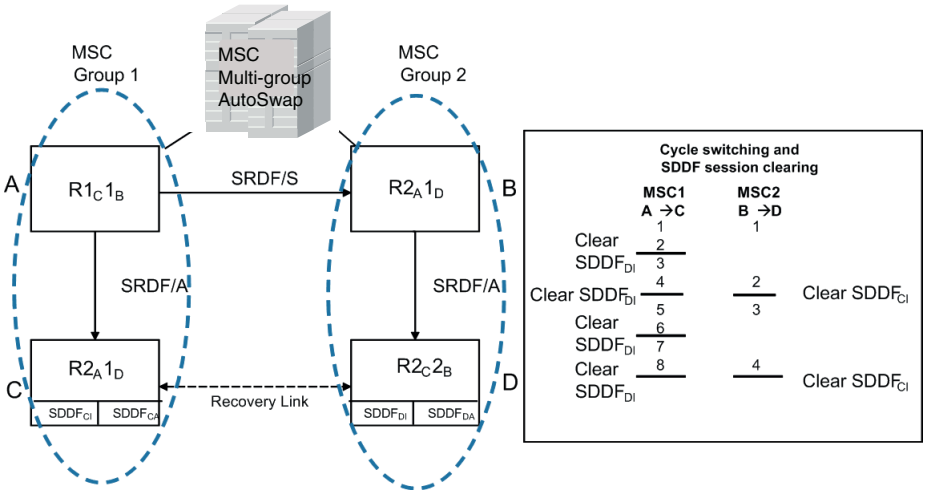



**Figure 5** SRDF/SQAR operations and SDDF session management

Establishing and sustaining normal SQAR operations with consistency requires the following:

- An active synchronous SRDF relationship between Site A and Site B with zero invalid tracks
- Two active SRDF/A relationships, A>C and B>D, each with zero invalid tracks and each exporting a positive SRDF/A consistency indication

The SRDF/S mirror (A->B) of the R11 devices for the MSC group A is ConGroup-protected. If a ConGroup trip occurs, MSC group B is dropped. Cycle switching continues for MSC group A after MSC has exported the C-ahead flag, but SDDF management is suspended (due to the dropping of B>D). SQAR processing returns to normal if MSC group B is restarted and ConGroup is resumed.

Two SDDF sessions each at Site C and Site D are used to track changes that are owed to other sites to ensure that the necessary amount of data is transferred during resynchronization operations. These bitmaps are periodically rotated and cleared after two cycle switches have occurred in each SRDF/A leg.

## SDDF session management

The objective of SDDF session management is to maintain the knowledge in each site of the data that the opposite site may not have. The periodic clearing and rotating of SDDF sessions at both sites optimizes the amount of data moved during a differential resynchronization by only moving tracks that may have changed at one site and not the other. This allows for a differential resynchronization that moves the least amount of data.

As shown in Figure 5 on page 31, there are two MSC host subtasks (one per MSC group) independently monitoring and driving the cycle switch processing of the A>C and B>D MSC groups. These tasks run in the same SCF address space. Each MSC subtask has two main functions:

- Perform SRDF/A cycle switching on one leg of the SQAR
- Manage the SDDF sessions on the opposite leg's SRDF/A R2 volumes

There are two SDDF sessions each at Site C and Site D. Each session is in one of two states (active or inactive) and toggles between these states under the control of one MSC subtask. The SDDF sessions serve as SQAR's 'memory' of what data a site has received that may not be at the opposite site.

The meaning of the SDDF session's bits is as follows:

- Site C: Data that Site C has that may not be at Site D
- Site D: Data that Site D has that may not be at Site C

An MSC subtask, for example, MSCGA, is responsible for performing SRDF/A cycle switching for one leg of the SQAR (MSCGA), and for ensuring that SDDF sessions on the SRDF/A R2 of the other leg (cycle switched by the MSCGB subtask) reflect that data has successfully arrived at the SRDF/A R2 under the control of the MSCGA subtask. MSCGA does this by clearing bits in the inactive session on MSCGB SRDF/A R2 after MSCGA has completed at least two cycle switches on its own leg. Following the clearing of the inactive session, the SDDF sessions are rotated making the active session 'inactive' and the inactive session 'active'.

After executing at least two cycle switches, the MSCGA subtask is aware of what data has reached Site C. Based on this knowledge, the task clears the inactive SDDF session in D because it knows whatever data is in the inactive SDDF session at Site D has already arrived at Site C.

As part of the session rotations, the inactive SDDF session is now cleared and made active. If a subsequent disaster occurs, any data that is tracked in both of Site D SDDF sessions must be data that Site D has which may not be at Site C. These bits represent tracks that must become part of a differential resynchronization to make devices at Site C and Site D equivalent.

Conversely, the MSCGB subtask rotates and clears the inactive SDDF session in Site C after it has completed at least two cycle switches because it knows the data is in Site D.

The SRDF/A cycle switching is only loosely coupled with the SDDF session management, that is, SDDF session management only occurs after at least two cycle switches, but does not need to occur every two cycle switches.

When SDDF processing is fully enabled for both SQAR groups, message SCF1630I (SQAR Recovery is now available) is issued. SQAR mode is indicated by the x'02' flag in the first byte of the cycle tag, which is set when SQAR recovery becomes available. The x'80' and x'40' bits indicating R2 consistency are common to MSC and are a prerequisite to recovery. Therefore, the first byte of the cycle tag will be x'C2' when SQAR recovery is available.

### Recovery and resynchronization

In the event of an unplanned outage requiring the restart of the production workload at another region, you must make two independent choices:

- Which site is to be the new primary site, and
- Which site has the most current data.

Data currency is determined by the presence or absence of the C-ahead flag, and by the SRDF/A secondary time-of-day value.

GDDR interrogates this time and delivers the information to guide the decision on resynchronization direction.

The data to be resynchronized is determined by performing an inclusive OR of all the SDDF bitmaps among the arrays and by instructing the array with the most current data to mark the resulting set of tracks as remotely invalid. All other sites mark them as locally invalid. Normal SRDF differential resynchronization follows.

### High availability support

GDDR SRDF/SQAR support provides the high availability takeover feature for the MSC task with the MSC TAKEOVER command initiating a takeover operation for both SQAR groups. If the primary MSC task is lost, a secondary MSC task takes over the cycle switch and SDDF session management of the SQAR configuration, becoming the primary MSC task.

## SQAR operational scenarios

In the following description of planned and unplanned scenarios supported in SRDF/SQAR, the link states are shown for each link and the 'LB' indicates a link block state on one mirror of an R22 device.

Only a subset of scenarios in planned and unplanned outage cases are presented here to illustrate capabilities included in SRDF/SQAR.

The starting point for normal operation is as follows:

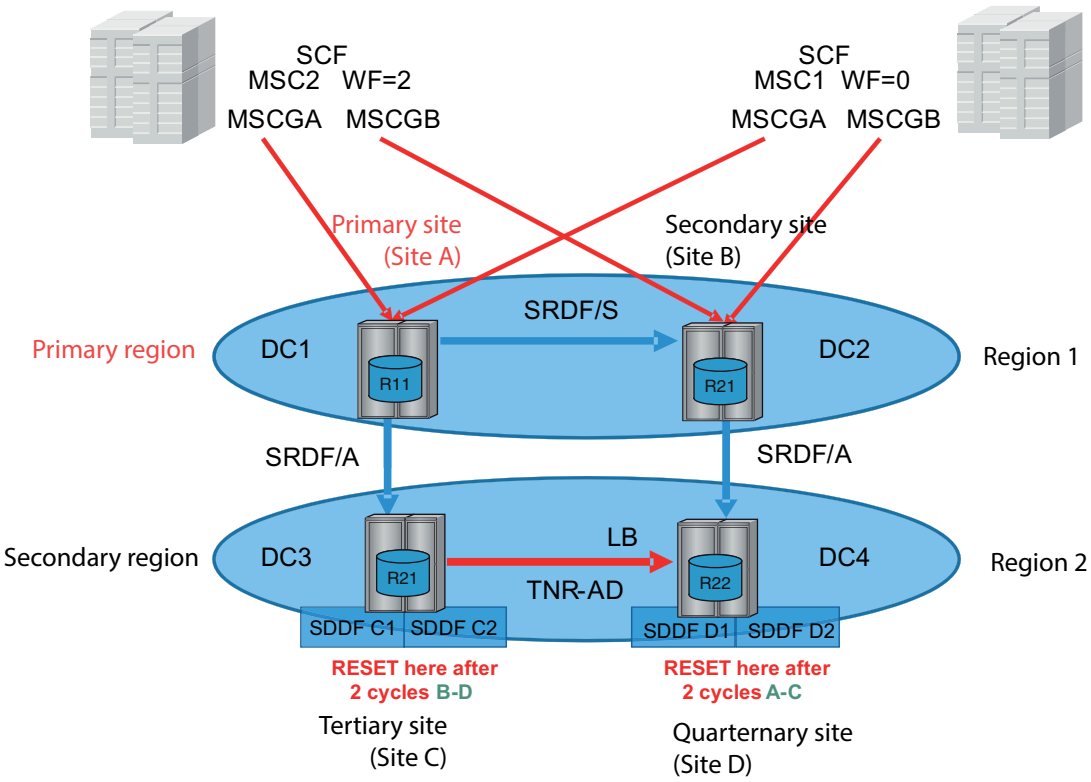



**Figure 6** SRDF/SQAR initial state

The term 'primary site' refers to the one site in a SQAR configuration that contains only R1 mirrors (see Figure 3 on page 29). Note how the logical site and region roles change in the scenarios but the physical site and region names remain constant.

## Planned scenarios

Planned scenarios include:

- Case 1: Inter-region swap
- Case 2: Intra-region swap

### Case 1: Inter-region swap

Primary/secondary and tertiary/quaternary sites swap regions (configuration rotates on horizontal axis). Two cases are shown with the primary site at DC3 or DC4. See Figure 6 on page 34 for the initial state.

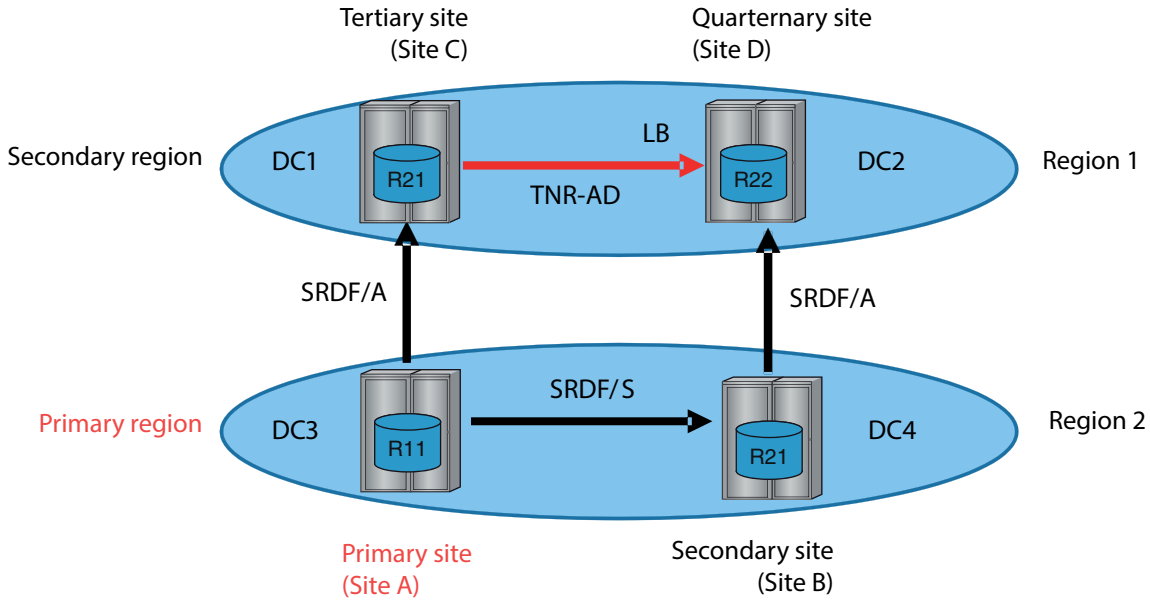



**Figure 7** SRDF/SQAR inter-region swap with primary site at DC3

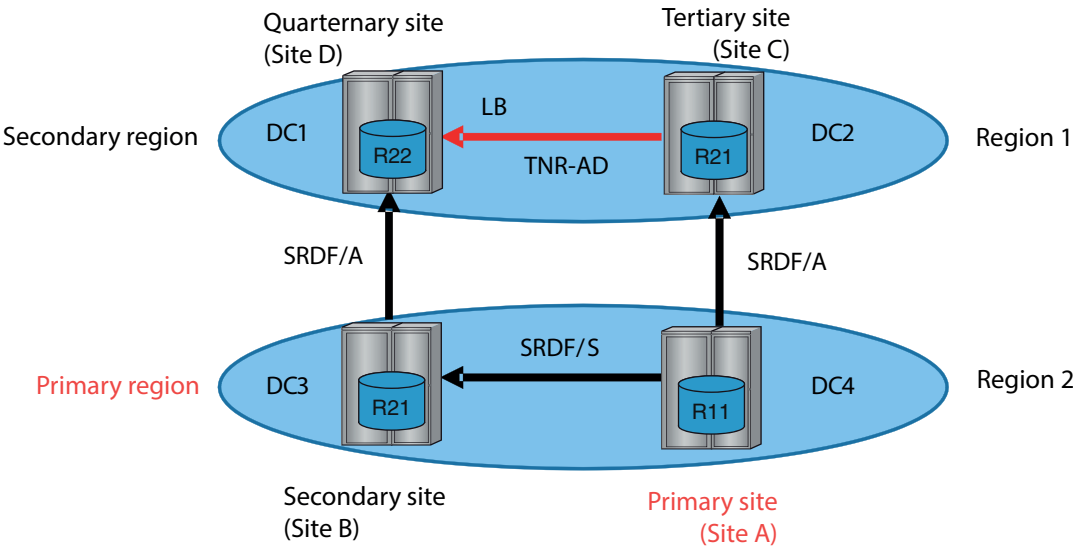



**Figure 8** SRDF/SQAR inter-region swap with primary site at DC4

## Case 2: Intra-region swap

Primary/secondary and tertiary/quaternary sites swap (the configuration rotates on vertical axis)

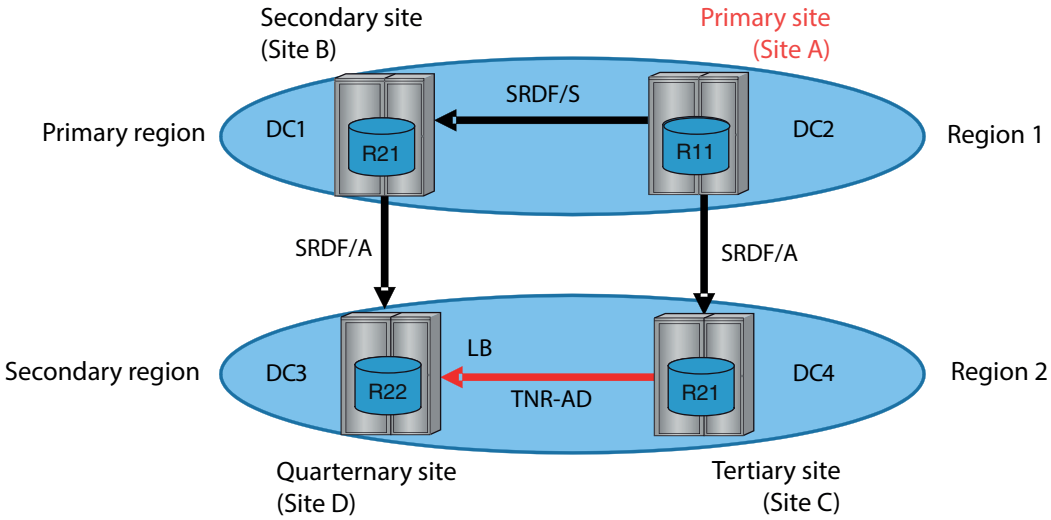



**Figure 9** SRDF/SQAR intra-region swap

## Unplanned scenarios

Unplanned scenarios include:

- Case 3a: Unplanned swap from Site A to Site B
- Case 3b: Unplanned swap from Site A to Site B
- Case 3d: Site A failure with recovery at Site C
- Case 3e: Site A failure with recovery at Site D
- Case 4a: Primary region failure
- Case 4b: Region 1 workload failure

### Case 3a: Unplanned swap from Site A to Site B

GDDR is configured with the default Unplanned Protection Preference of SRDF/S. GDDR validates the configuration and finds out that a full intra-region swap is possible.

The result is identical to the planned intra-region swap shown in Figure 9 on page 36.

### Case 3b: Unplanned swap from Site A to Site B

GDDR is configured with the Unplanned Protection Preference of SRDF/A, or GDDR validates the configuration and finds out that a full intra-region swap is not possible.

GDDR performs an intra-region site swap with remote replication from Site B to Site D only.

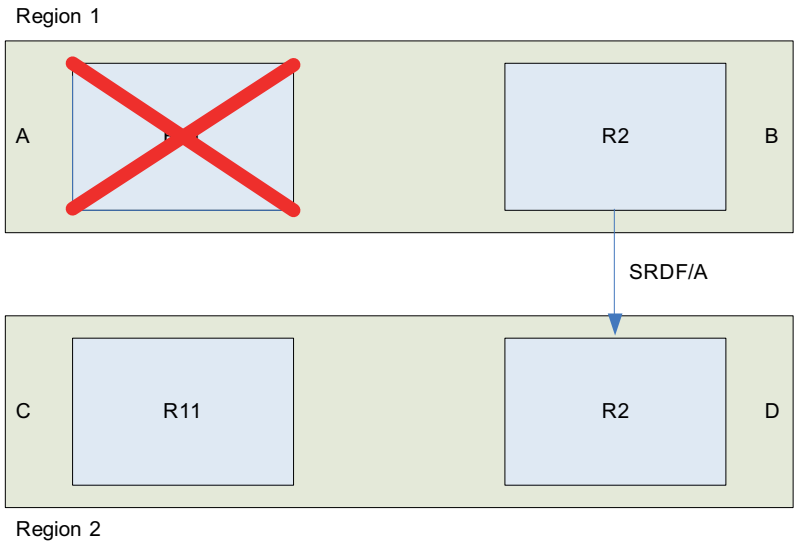



**Figure 10** SRDF/SQAR unplanned swap from Site A to Site B

## Case 3c: Loss of Site A with recovery at Site B

A ConGroup trip occurs. Then Site A is lost. GDDR ensures recovery at Site D (using Site B data drained to Site D) and protection to C (synchronous) and B (asynchronous).

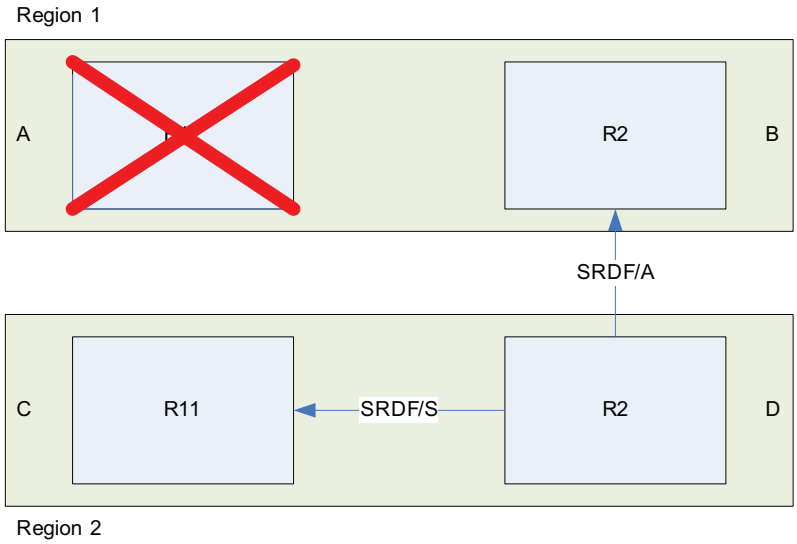



**Figure 11** SRDF/SQAR Site A failure with recovery at Site B

**Note:** GDDR blocks this scenario if Site C has more recent data than Site B.

## Case 3d: Site A failure with recovery at Site C

GDDR ensures recovery at Site C and protection to C>D (synchronous) and D>B (asynchronous).

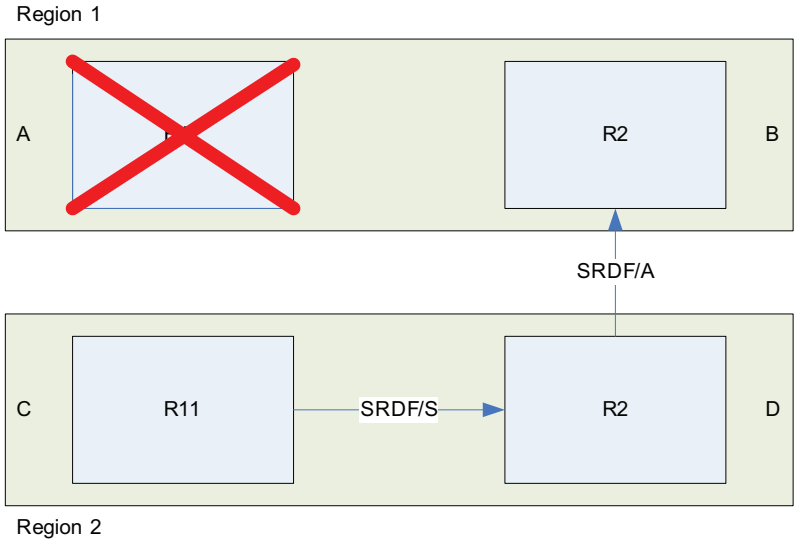



**Figure 12** SRDF/SQAR Site A failure with recovery at Site C

**Note:** GDDR blocks this recovery scenario if Site B has more recent data than Site C.

## Case 3e: Site A failure with recovery at Site D

GDDR ensures recovery at Site D (using Site B data drained to Site D) and protection to C (synchronous) and B (asynchronous).

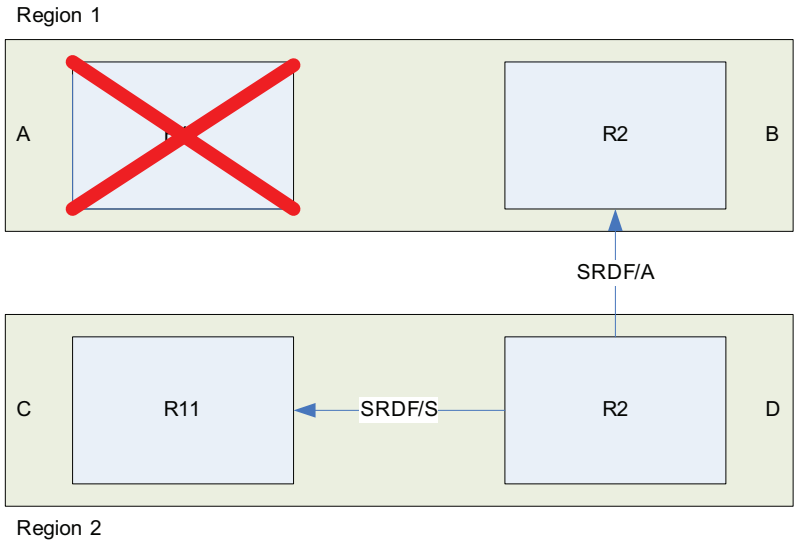



**Figure 13** SRDF/SQAR Site A failure with recovery at Site D

## Case 4a: Primary region failure

GDDR ensures recovery at Site C and protection to Site D using the latest copy of data.

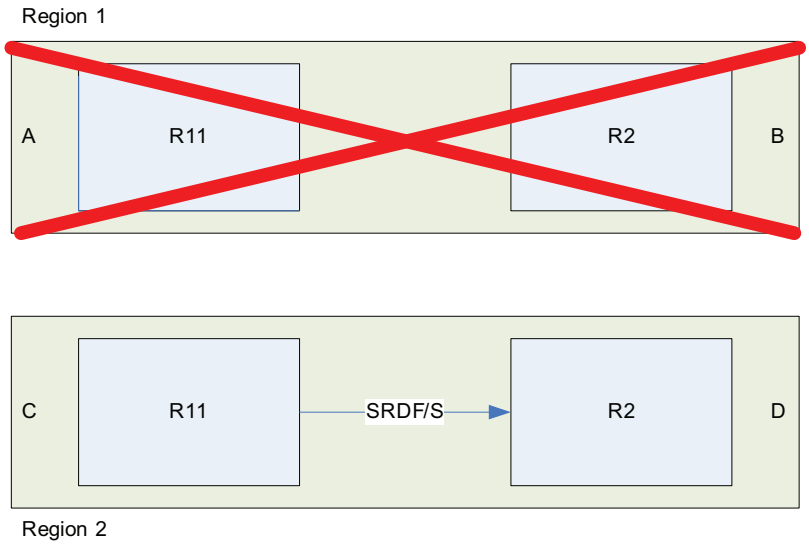



**Figure 14** SRDF/SQAR primary region failure

## Case 4b: Region 1 workload failure

GDDR ensures recovery at Site D and protection to Site C using the latest copy of data.

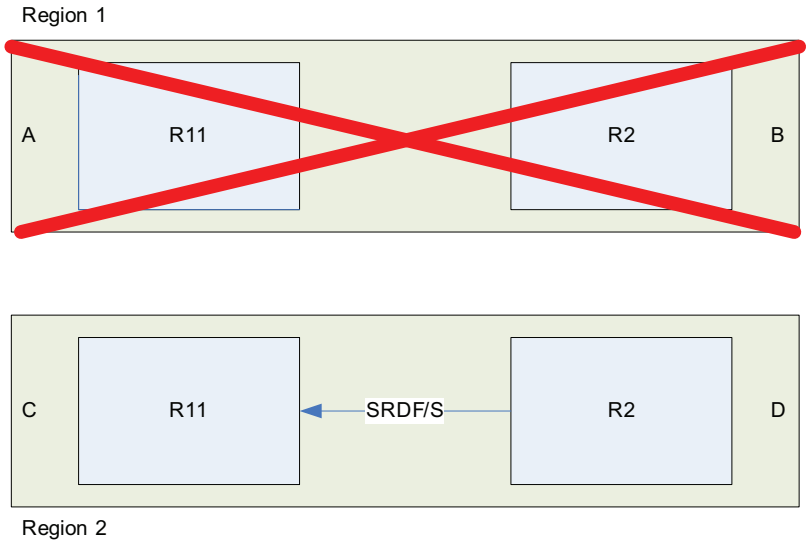



**Figure 15** SRDF/SQAR region 1 workload failure

# GDDR fundamentals

This section discusses the following GDDR fundamentals:

- GDDR sites and regions
- GDDR-plex
- GDDR systems

## GDDR sites and regions

A physical site is a real-world location where various components such as DASD, tape storage and CPU are co-located and connected to form a *data center*. For the SRDF/SQAR configuration, data centers are abbreviated as DC1, DC2, DC3, and DC4.

A physical region is a set of one or more physical sites located within SRDF/Synchronous distance from each other. For the SRDF/SQAR configuration, regions are abbreviated as RG1 and RG2. Physical sites that are at SRDF/Asynchronous distances from each other, by definition are in different physical regions.

IMPORTANT

Physical site and region designations do not change during the lifespan of a GDDR configuration.

Logical site and region designations reflect the role of the given site or region in the current GDDR environment. For the SRDF/SQAR configuration, the site role can be described as *primary*, *secondary*, *tertiary*, or *quaternary*, referenced using the corresponding letters A, B, C, and D respectively. The region role can be described as *primary* or *secondary*.

Sites and regions can change roles due to planned or unplanned site and/or region swaps.

An initial SRDF/SQAR configuration has the following physical-to-logical correspondence:

- DC1 is the primary site (Site A) located in primary region RG1.
- DC2 is the secondary site (Site B) located in primary region RG1. DC2 is connected to DC1 with SRDF/S.
- DC3 is the tertiary site (Site C) located in secondary region RG2. DC3 is connected to DC1 with SRDF/A.
- DC4 is the quaternary site (Site D) located in secondary region RG2. DC4 is actively connected to DC2 with SRDF/A, and to DC3 as a recovery connection.

**Note:** The SITE statement in the GDDRPARM file determines the sites and regions used in a GDDR configuration.

## GDDR-plex

The collection of GDDR control systems and managed (production or contingency) systems together with the managed hardware they share, is known as a *GDDR-plex*. A single GDDR-plex can span across multiple sysplexes. On the other hand, a single set of control systems can run multiple GDDR-plexes, each implemented using a dedicated set of GDDR started tasks, each governed by its own set of parameters.

In a GDDR-plex managing an SRDF/SQAR configuration, the business or production workload can run as either:

- A single-site workload

  All production workload runs at a single site only; that is, one side of the sysplex. This is normally the same location as the primary DASD site.

- A multi-site workload

  Production workload runs at both the primary and secondary sites.

## GDDR systems

A GDDR system is a z/OS system that acts as either a control system (C-system) or a managed system (P-system) in the GDDR-plex.

"GDDR system types" on page 46 provides detailed information on each system type.

# GDDR components

GDDR consists of the following components:

- GDDR user interface
- GDDRMAIN
- GDDR scripts
- GDDR parameters
- GDDR events
- GDDR monitors
- GDDR workers
- Message interception rules
- GDDR utilities

## GDDR user interface

The GDDR user interface is an ISPF application that you can use to access all features and capabilities of GDDR.

Chapter 6, "Using GDDR ISPF Interface," presents the GDDR ISPF interface.

## GDDRMAIN

GDDRMAIN is the heart of GDDR. The GDDRMAIN address space controls GDDR global variable management, message interception, and communication between C-systems and managed systems.

Chapter 5, "Running GDDRMAIN," provides a detailed description of GDDRMAIN, including GDDRMAIN parameters ("GDDRPARM statements" on page 256) and console commands ("GDDRMAIN console commands" on page 173).

## GDDR scripts

GDDR scripts are used to perform the following actions in the GDDR environment:

- Initiate planned events, such as swapping the production workload to another site because of scheduled system maintenance at the original workload site
- Respond to unplanned events, such as losing a site
- Perform testing and maintenance functions, such as verifying the GDDR configuration

Chapter 8, "Running GDDR Scripts," provides detailed information about GDDR scripts.

## GDDR parameters

GDDR parameters define the managed environment and configuration.

You set GDDR parameters using the GDDR Parameter Wizard, which is part of the GDDR ISPF interface. "GDDR Parameter Wizard" on page 54 provides an overview of the parameters you can set with the wizard and the workflow. "Manage GDDR parameters (M,P)—GDDR Parameter Wizard" on page 287 instructs on how to complete all the panels available in the GDDR Parameter Wizard.

## GDDR events

GDDR monitors the managed environment for changes in state of its components, such as link failures, missing heartbeat, and so on. GDDR responds to events by taking certain actions, for example, preventing a planned process from running.

**Note:** "GDDR environment monitoring" on page 62 provides more information and a list of GDDR events.

### Expected events

GDDR recognizes *expected* events. When GDDR expects that the current GDDR action would cause a certain event to occur, such event is listed as an expected event. Expected events require no user action.

**Note:** "Expected events" on page 69 provides more information and a list of expected events.

## GDDR monitors

GDDR includes a number of purpose-specific monitors that check the GDDR-managed environment continually or at a user-defined interval:

- GDDR Event Monitor
- GDDR Heartbeat Monitor
- GDDR DLm Heartbeat Monitor

## GDDR workers

GDDR includes a number of worker tasks that process work on behalf of GDDR scripts and processes. See "Worker task management" on page 165 for a complete list and description of GDDR workers.

## Message interception rules

GDDR is supplied with message interception rules that help automate response to events occurring in the GDDR-managed environment.

"Message interception" on page 66 further explains message interception.

## GDDR utilities

GDDR utilities enable you to perform various administration, configuration, and maintenance tasks.

Chapter 7, "Using GDDR Utilities," lists the GDDR utilities.

# CHAPTER 2
# GDDR Features and Concepts

This chapter includes the following topics:


- GDDR system types ..... 46
- GDDR C-system multi-tenancy ..... 51
- GDDR Parameter Wizard ..... 54
- GDDR environment monitoring ..... 62
- GDDR licensing options ..... 71
- DLm integration using DLMDR ..... 72
- SRDF/SQAR High Availability (HA) ..... 74
- TimeFinder device management ..... 75
- GDDR support for external devices ..... 81
- BCPii support ..... 82
- CPC Recovery—LPAR Recovery ..... 84
- GDDR script call overrides ..... 85
- GDDR user message automation ..... 89
- GDDR user exits ..... 90

# GDDR system types

GDDR system types are as follows:

- C-systems
  - Master C-system
- Managed systems
  - Production systems
  - Contingency systems
  - Excluded systems
  - HMC-only systems
  - HMC Bypass systems
  - External workload systems
  - Recovery LPARs
  - Coupling facility LPARs

## C-systems

The GDDR control systems are more commonly referred to as *C-systems*. One C-system is located at each site (DC1, DC2, DC3, and DC4).

The site for a given C-system is defined using the CSYSSITE statement in the GDDRPARM file.

C-systems must be configured as standalone systems by specifying either XCFLOCAL or MONOPLEX in the PARMLIB IEASYS PLEXCFG parameter (XCFLOCAL is recommended). Each C-system runs as a standalone z/OS system from local DASD. However, multiple GDDR instances can run on a single C-system, using the GDDR Multi-Tenancy feature described in "GDDR C-system multi-tenancy" on page 51.

---

**Note:** It is recommended that the C-system DASD are located on separate controllers from the production DASD. Because the Dell EMC software applications run from local C-system volumes, this separation ensures that the C-systems are not affected by events that may impact the availability of the managed systems.

---

C-systems do not run any production workload. The main functions of a C-system are as follows:

- Control the recovery after an outage
- Control a planned site swap
- Provide the GDDR ISPF interface (described in Chapter 6, "Using GDDR ISPF Interface")
- Run the GDDR Event Monitor, GDDR Heartbeat Monitor, and GDDR DLm Heartbeat Monitor (described in "GDDR monitors" on page 62)

### Master C-system

One of the C-systems is the *master C-system*. During normal operations, the master C-system is the central point of control for all GDDR activities. Most GDDR actions are allowed only when logged in to the master C-system.

The master C-system is located at the secondary DASD site. This ensures that, in the event of losing the primary DASD site, the master C-system survives to coordinate recovery.

The master C-system is located at the opposite site from the business applications.

If the master C-system becomes unavailable for some reason, a C-system at another location/site assumes the GDDR master function ownership. When the original master C-system becomes available, the master function ownership automatically transfers back to the correct location.

Some GDDR functions can only be carried out by the master C-system, for example:

- Run planned processes and GDDR scripts
- Update GDDR parameters

All C-systems are potential candidates to take over as the master C-system. GDDR automatically detects an outage of the master C-system function and presents the appropriate recovery options, including the takeover of the master C-system role by another surviving C-system.

## Managed systems

In addition to C-systems, the following types of systems can be defined to GDDR:

- Production or test systems and their optional contingency systems
- Excluded systems
- HMC-only systems
- External workload systems

These systems are referred to as *managed systems*.

The site for a given managed system is defined using the Define Managed Systems panel (M,P,H,S).

### Production systems

The GDDR production systems are more commonly referred to as *P-systems*. A P-system is a managed system that normally runs the site's production workload and updates the primary DASD.

Production systems are typically located at the same location as the primary DASD.

GDDR can trigger stop and restart of production workloads on z/OS systems and distributed systems.

Production systems require that GDDRMAIN is running on these systems at all times to take full advantage of the following GDDR system management capabilities:

- Production systems run from GDDR-managed DASD.

- GDDR triggers start and stop of business workload during execution of GDDR scripts.
- GDDR triggers stop or start of ConGroup on production systems if Call_Override bytes 08/09 are true.
- Production systems can be IPLed, reset, activated, or deactivated during execution of certain scripts.
- Production systems can be protected using the LPAR Recovery feature and can participate in CPC (Central Processor Complex) swaps.
- Production systems can be managed from the Perform HMC LPAR Actions panel (A,L).
- GDDR can perform CBU actions (initiated using scripts or ISPF panels) for the CPCs on which production systems run.
- GDDR manages or simulates management of coupled datasets and CF (Couple Facility) structures for production systems, if they are in a sysplex.
- GDDR performs VARY ONLINE/OFFLINE processing for tape devices using DLm.

Note the following requirements:

- Production systems must be specified in COMM parameters in the GDDRPARM file.
- The Define Managed Systems panel (M,P,H,S) must have an entry for each system.
- Production systems must have a CPC and LPAR defined for them.
- Production systems must have IPL parameters defined for them.
- Production systems must have Mainframe Enablers started tasks defined for them.

## Contingency systems

A *contingency* or *standby* system is a managed system that normally provides hot backup to a production system. A contingency system:

- May be in a geographically distanced region that is equipped with processor capacity reserved for support of business workload restarts.

Multiple locations containing contingency or standby systems may be used to increase availability and provide disaster restart options. Regional contingency systems in an SRDF/SQAR configuration are typically in the same location as the secondary DASD, while out-of-region standby systems provide protection from geographic and infrastructure exposures that may negatively impact the primary and secondary sites.

Contingency systems run from GDDR-managed DASD.

Contingency systems require that GDDRMAIN is running on these systems at all times to take full advantage of the following GDDR system management capabilities:

- GDDR triggers start and stop of business workload during execution of GDDR scripts.
- GDDR triggers stop or start of ConGroup on contingency systems if Call_Override bytes 08/09 are true.

- Contingency systems can be IPLed, reset, activated, or deactivated during execution of certain scripts.
- Contingency systems can be protected using the LPAR Recovery feature and can participate in CPC (Central Processor Complex) swaps.
- Contingency systems can be managed from the Perform HMC LPAR Actions panel (A,L).
- GDDR can perform CBU actions (initiated using scripts or ISPF panels) for the CPCs on which contingency systems run.
- GDDR manages or simulates the management of coupled datasets and CF structures for contingency systems, if they are in a sysplex.

Note the following requirements:

- Contingency systems must be specified in COMM parameters in the GDDRPARM file.
- The Define Managed Systems panel (M,P,H,S) must have an entry for each system.
- Contingency systems must have a CPC and LPAR defined for them.
- Contingency systems must have IPL parameters defined for them.
- Contingency systems must have Mainframe Enablers started tasks defined for them.

## Excluded systems

GDDR can be configured to exclude certain systems from workload management, although these systems have their DASD in the managed storage systems.

Excluded systems are mainframe systems running from the GDDR-managed DASD.

GDDR capabilities and restrictions are as follows:

- GDDR triggers stop or start of ConGroup on excluded systems if Call_Override bytes 08/09 are true.
- GDDR does not trigger start or stop of workload for excluded systems.
- GDDR does not perform HMC LPAR actions for excluded systems during script execution, but they are listed on the Perform HMC LPAR Actions panel (A,L).
- Excluded systems can be specified in COMM parameters in the GDDRPARM file.
- Excluded systems can have a CPC and LPAR defined for them.
- Excluded systems can have Mainframe Enablers started tasks defined for them.

**Note:** The last three items are required if you intend to use the ConGroup Stop/Start method (Call_Override bytes 08/09=1).

Excluded systems are displayed in the Define Managed Systems panel (M,P,H,S) as "Manage Workload=NO".

You can add or delete excluded systems from the Define Managed Systems panel (M,P,H,S), unless they are specified in COMM parameters in the GDDRPARM file.

## HMC-only systems

GDDR can be configured to limit IPL and CBU actions for certain systems to the online interface. No other actions or automation are performed for these systems.

The only functionality for HMC-only systems is ACT/DEACT/LOAD/RESET actions performed from within the Perform HMC LPAR Actions panel (A,L). To facilitate automatic discovery, it is recommended to define HMC-only systems in the GDDRPARM file.

GDDR capabilities and requirements are as follows:

- HMC-only systems must have a CPC and LPAR defined.
- HMC-only systems must have IPL parameters defined.
- You can add or delete HMC-only systems from the Define Managed Systems panel (M,P,H,S), unless they are specified in COMM parameters in the GDDRPARM file.

## HMC Bypass systems

If the site where GDDR is running is under management of a third-party facility provider, GDDR offers the HMC Bypass feature, by site and by LPAR, to prevent GDDR HMC interaction with all or selected LPARs at a specific site.

## External workload systems

External workloads run on mainframe systems which do not have their DASD in the managed storage systems.

GDDR can coordinate stop and start of workload on these non-managed mainframe systems with the workload stop and start for managed systems.

This category of systems is displayed in the Define External Workloads panel (M,P,H,W). External workload systems have very limited support. These are mainframe systems; they are not running from the GDDR-managed DASD. However, when GDDR triggers stop or start of workload for GDDR-managed systems, the external workload systems are included as well.

## Recovery LPARs

Recovery LPARs provide a way to run the same workload in two different locations at different times. A recovery LPAR is located on the same CPC, or on a different CPC, at the same site, or at a different site. In AutoSwap configurations, the recovery LPAR is located at a different site than the “home” LPAR of the system. When a system defined with a recovery LPAR is lost, an additional recovery option is presented to the operators. Managed systems can have a contingency system as well as a recovery LPAR.

## Coupling facility LPARs

The only functionality for these systems is ACT/DEACT actions performed from within scripts and the Perform HMC LPAR Actions panel (A,L). During activation, coupling facility (CF) LPARs are processed first. During deactivation, they are processed last, after all other LPARs.

GDDR restrictions are as follows:

- CF LPARs cannot have a recovery LPAR defined at another site.
- There are no IPL parameters associated with CF LPARs.
- There is no COMM statement defined for CF LPARs in the GDDRPARM file.

# GDDR C-system multi-tenancy

The C-System Multi-Tenancy feature virtualizes the concept of a C-system. It allows multiple instances of the C-system function to run on a single z/OS system, without interfering with each other.

**Note:** C-System Multi-Tenancy can also be referred to as Multi-GDDR.

With C-System Multi-Tenancy, multiple GDDR-plexes that manage different storage configurations can share the same set of C-systems. This helps decrease the number of mainframe LPARs dedicated to GDDR and reduce hardware costs.

Multi-tenancy is supported for C-systems only. A managed system (P-system) can have only one active GDDR instance.

There is no limit to the number of C-instances that can run on a single z/OS-system, barring resource limitations. However, there is a limit of 8 ConGroup started tasks per system, limiting the number of C-instances requiring ConGroup to 8.

"Key concepts" on page 51 introduces the concepts that are used to implement C-System Multi-Tenancy. "Configure GDDR C-System Multi-Tenancy" on page 155 explains how to configure your environment to use C-System Multi-Tenancy.

## Key concepts

The following terms are used in the context of C-System Multi-Tenancy:

- GDDRMAIN instance
- GDDR instance
- C-instance
- P-instance
- Mainframe Enablers instance
- Affinity group

### GDDRMAIN instance

A single GDDRMAIN address space running on a z/OS system.

### GDDR instance

The combination of the GDDRMAIN instance and its dependent address spaces running on a z/OS system.

The related GDDR address spaces are tied together by the same GDDR subsystem name. They include the GDDRMAIN address space, a variable number of GDDRWORK address spaces, and, for the C-system function only, the GDDREVM, GDDR scripts, and GDDR ISPF user address spaces.

A GDDR instance can be a *C-instance* or a *P-instance*.

### C-instance

A GDDR instance running on a z/OS system that is defined to GDDR as a C-system.

A C-instance consists of a single GDDRMAIN address space and its dependent address spaces (GDDREVM, GDDRWORK, GDDR scripts) running on a specific z/OS system.

A C-system can have multiple C-instances running on it at the same time.

Different C-instances running on a single z/OS system are completely disjointed. C-instances cannot share MFE address spaces, and a single C-instance cannot be part of multiple *GDDR-plexes*.

Each C-instance operates exclusively within its own set of address spaces. A particular C-instance does not react to messages nor reply to WTORs issued by an address space outside of its *affinity group*.

### P-instance

A GDDR instance running on a z/OS system that is defined to GDDR as a managed system (P-system).

A managed system can have only one active P-instance.

### Mainframe Enablers instance

The combination of a single SCF address space and other Mainframe Enablers address spaces using the same SCF subsystem name. Each GDDR instance requires a dedicated MFE instance.

### Affinity group

The combination of a GDDR instance and its dedicated set of Mainframe Enablers started tasks.

### GDDR-plex

---

**Note:** The following definition applies when C-System Multi-Tenancy is enabled. Otherwise, see "GDDR-plex" on page 42.

---

A set of C-instances and their managed P-systems connected via TCP/IP. Each GDDR-plex uses its own dedicated TCP/IP port.

One GDDR-plex manages a single set of consistent data.

Different GDDR-plexes operate independently of each other. Each GDDR-plex operates exclusively within its own set of GDDR instances. Messages are not forwarded to a C-instance or P-instance outside a specific GDDR-plex.

C-System Multi-Tenancy allows multiple GDDR-plexes managing different storage configurations to share the same set of z/OS systems for the C-system function. This decreases the number of mainframe LPARs that must be dedicated to GDDR and therefore reduces hardware costs.

## Message affinity

Support for multiple C-instances on the same z/OS system is made possible primarily by a mechanism called message affinity.

Message affinity exploits the requirements that each GDDR instance must have a unique GDDR subsystem name and a unique SCF subsystem name, specified using the GDD$*nnnn* and SCF$*nnnn* connectors in the JCL. One GDDR subsystem connects to only one SCF subsystem.

**Note:** "GDDR and SCF connectors" on page 169 discusses the GDD$*nnnn* and SCF$*nnnn* connectors.

Each GDDRMAIN instance monitors all message traffic on that given system. When a message is intercepted, GDDR checks whether the message has affinity to a given GDDR instance using the following rules:

- GDDR messages have affinity with the GDDR instance only if the issuer's GDD$*nnnn* connector matches the GDD$*nnnn* connector defined for the GDDR instance.
- MFE messages have affinity with the GDDR instance only if the issuer's SCF$*nnnn* connector matches the SCF$*nnnn* connector set for the GDDR instance.
- Non-Dell EMC messages are not subject to affinity checking.
- Messages forwarded from remote systems or issued on a different system in the same sysplex are not subject to affinity checking.

If it is determined that the message has affinity with the GDDR instance and message automation is enabled for that message ID, the message is processed or forwarded as indicated in the GDDR message intercept table. Otherwise, the message is ignored.

Message affinity applies to both C-instances and P-instances.

Message affinity checking helps keep different GDDR-plexes fully isolated from each other and prevents a GDDR instance from reacting to a message that does not belong to it, either by processing it locally or by forwarding it to another system in the GDDR-plex. It also prevents a GDDR instance from reacting to messages issued by any Mainframe Enablers started tasks running outside of a GDDR-managed configuration.

# GDDR Parameter Wizard

The GDDR Parameter Wizard is a series of ISPF panels used to initially define, and later maintain, GDDR parameters. Each panel is backed by a member of a PDS.

To access the GDDR Parameter Wizard, choose option **M** in the GDDR ISPF Primary Options Menu. In the Setup and Maintenance Menu panel that is displayed, choose option **P**. Proceed as described in "Manage GDDR parameters (M,P)—GDDR Parameter Wizard" on page 287.

## Parameters set with GDDR Parameter Wizard

GDDR parameters define the GDDR managed environment and configuration. The parameters can modify the sequence of function calls that constitute a GDDR script.

The environment that GDDR manages is described to GDDR through a collection of common variables as follows:

- Configuration-defining variables

  These variables define the managed configuration type, the C-systems, the initial role for each site, the consistency group names and the MSC group names.

- Storage object variables

  These variables define the SRDF and TimeFinder devices, SRDF groups, SnapVX snapshot names, DLm devices, and gatekeeper devices that form the GDDR-managed configuration.

- Host object variables

  These variables define the managed systems, external systems, and HMC-only systems, as well as their LPARs, CF LPARs, system recovery attributes, IPL parameters, IPL priorities, and CPCs. Host object variables also define HMC consoles, sysplex objects, and Mainframe Enablers started tasks.

- GDDR option variables

  These variables define user-selectable values for a variety of actions taken during GDDR automation sequences. GDDR option variables also define site defaults for JCL and utilities used by GDDR, and GDDR tuning values.

## GDDR Parameter Wizard prerequisites

The following items are required to run the GDDR Parameter Wizard:

- The GDDRPARM file has been defined with C-system and managed system COMM statements and is consistent on all systems.

  **Note:** "Define GDDRPARM file" on page 140 describes this process.

- GDDRMAIN tasks are active and communicating on all C-systems and optionally also on managed systems (recommended).

- Parameter backup and parameter work datasets are allocated on all C-systems.

  **Note:** "Define GDDR datasets" on page 139 describes this process.

  Parameter backup datasets are populated by the GDDR Heartbeat Monitor on all C-systems. Parameter work datasets are populated when you run the GDDR Parameter Wizard.

- One *last activated parameter dataset* is pre-allocated on all C-systems.

  **Note:** "Define GDDR datasets" on page 139 describes this process.

## Intended master C-system

Parameter validation/activation is performed on the C-system which is the intended master C-system for the configuration being defined (with the exception of missing C-system communication). The master C-system must be on the secondary DASD site.The Validate operation queries the storage environment using the site, system, gatekeeper, SRDF group, and device ranges presented as GDDR Parameter Wizard input. The presence of R1 devices determines the current primary DASD site. If the query is unable to find R1 devices, the primary DASD site found in the parameter input deck is used during validation.

## Auto-Discovery

The GDDR Parameter Wizard helps you define and maintain the parameters that describe your particular disaster restart topography and SRDF, and optional TimeFinder, DLm and host configuration by populating values available through the Auto-Discovery feature.

System parameter changes that are required because of changes made to the definitions of C-systems or managed systems are initiated by modifying the CSYSSITE and COMM parameters in the GDDRPARM file according to instructions in "Updating GDDRPARM file statements" on page 573. After modifying the parameters in the GDDRPARM file, the changes are propagated to the active GDDR parameter set using the GDDR Parameter Wizard.

The Auto-Discovery feature requires that the GDDRMAIN tasks have been started on each C-system and managed system before you start the GDDR Parameter Wizard in the Select Parameter Input Dataset panel (M,P,I).

## GDDR Parameter Wizard workflow

GDDR Parameter Wizard is based on an iterative change process. Changes are applied initially to a consistent work copy of the active GDDR global variables. These changes are then applied to the active GDDR global variables in an activation step (preceded by an implicit or explicit validation).

To start a new GDDR Parameter Wizard session, choose a backup of GDDR global variables to be used as input for the process. In most cases, a new backup is to be created, so the process starts from the current set of global variables. Alternatively, you can select any backup of GDDR global variables created earlier, to restore the configuration to an earlier point in time.

The GDDR Parameter Wizard further requires a GDDR Parameter Wizard work dataset. The process of selecting a suitable backup as input for a new GDDR Parameter Wizard session overwrites the content of the GDDR Parameter Wizard work dataset with the global variables as found in the selected backup.

The GDDR Parameter Wizard offers a sequence of panels allowing you to perform selective updates to the GDDR Parameter Wizard work dataset.

On completion of the GDDR Parameter Wizard work dataset updates using the GDDR Parameter Wizard panels, you can perform validation to ensure that the new set of global variables meets criteria of configuration consistency.

This validation can be done up-front, and is also implicitly included in the activation process. The activation process updates the active GDDR global variables with the contents of the GDDR Parameter Wizard work dataset.

Figure 16 illustrates the parameter backup, edit serialization, validation and activation process, and the relationship with GDDRMAIN and its dependent address spaces.

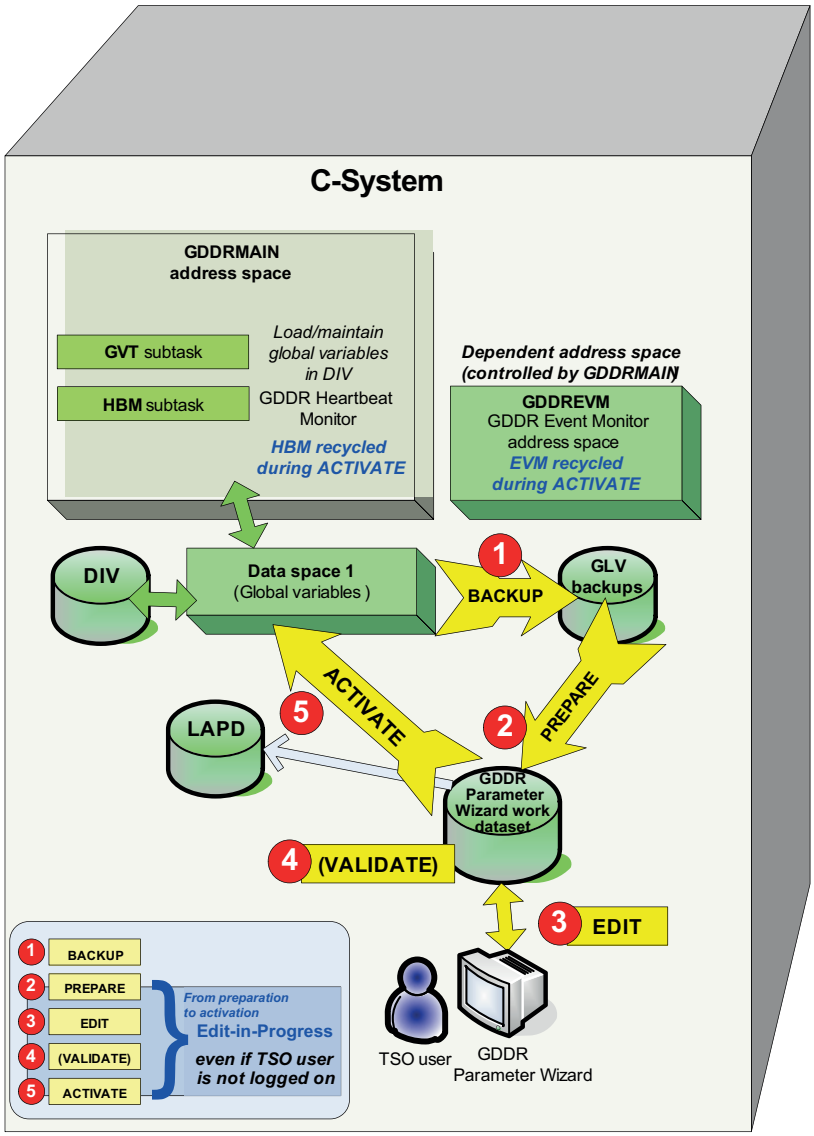



**Figure 16** GDDR parameter management

The GDDR Parameter Wizard workflow includes the following steps:

- Step 1: Backup
- Step 2: Prepare
- Step 3: Update
- Step 4: Populate
- Step 5: Validate
- Step 6: Activate

## Step 1: Backup

Initial definition of global variables is performed using no prior backup, with automation which populates system parameters available from the GDDRPARM file installed during GDDRMAIN installation.

Maintenance of global variables is applied using a current global variable backup as the base.

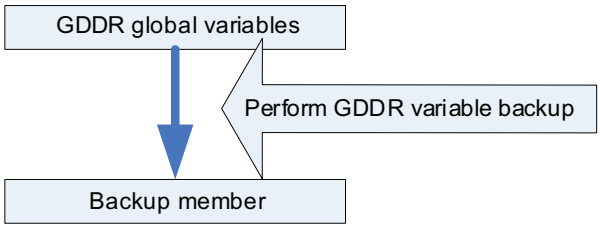



**Figure 17** GDDR Parameter Wizard workflow—step 1

## Step 2: Prepare

GDDR Parameter Wizard usage is limited to one active session at a time.

When you signal the intent to update parameters by performing the Prepare command, access to the GDDR Parameter Wizard panels by other users is prevented. You can review the GDDR configuration parameters at any time using the Review command.

Serialization of the panels is held until you have completed parameter activation. (An emergency procedure is available in case this serialization creates a problem.)

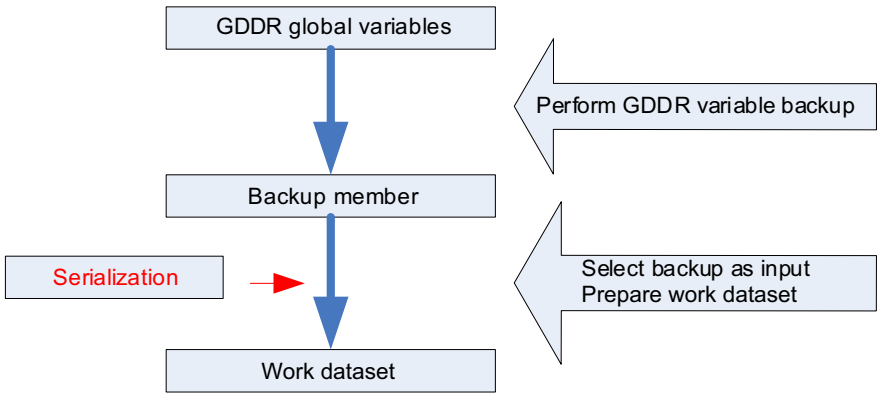



**Figure 18** GDDR Parameter Wizard workflow—step 2

## Step 3: Update

The GDDR Parameter Wizard updates the parameter work dataset which was specified at the time the Prepare (or Review) command was issued. It does this by storing each global variable found in the selected input dataset to the appropriate work dataset member.

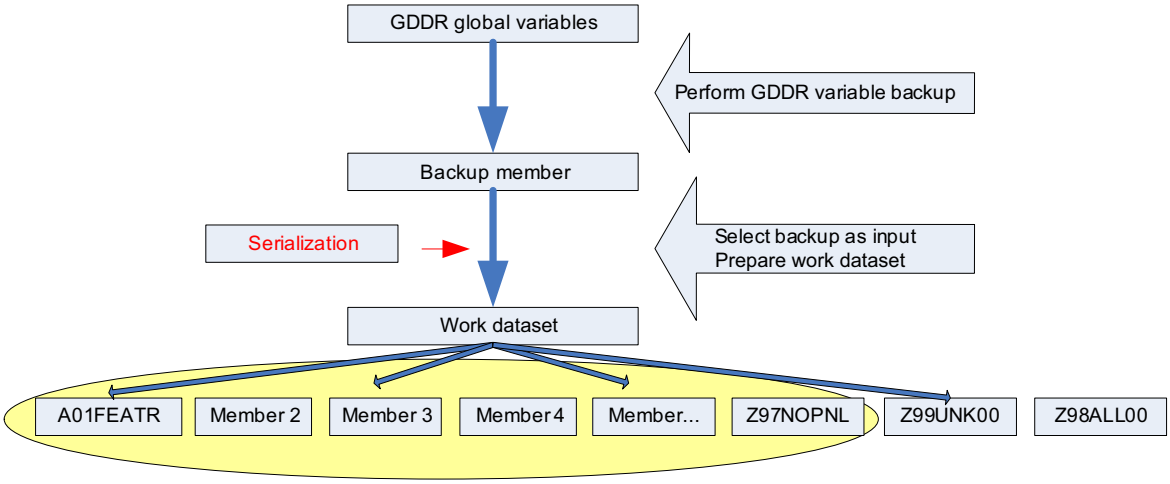



**Figure 19** GDDR Parameter Wizard workflow—step 3

## Step 4: Populate

Each GDDR Parameter Wizard panel lets you input values for the GDDR global variables found in the corresponding work dataset member. Selected GDDR Parameter Wizard panels are populated with data from GDDRPARM parameters, either directly or using discovery functions.

Where discrepancies exist between GDDRPARM derived data and data which is provided from the 'base' global variable backup, the difference is indicated in the panels by displaying this data in yellow font.

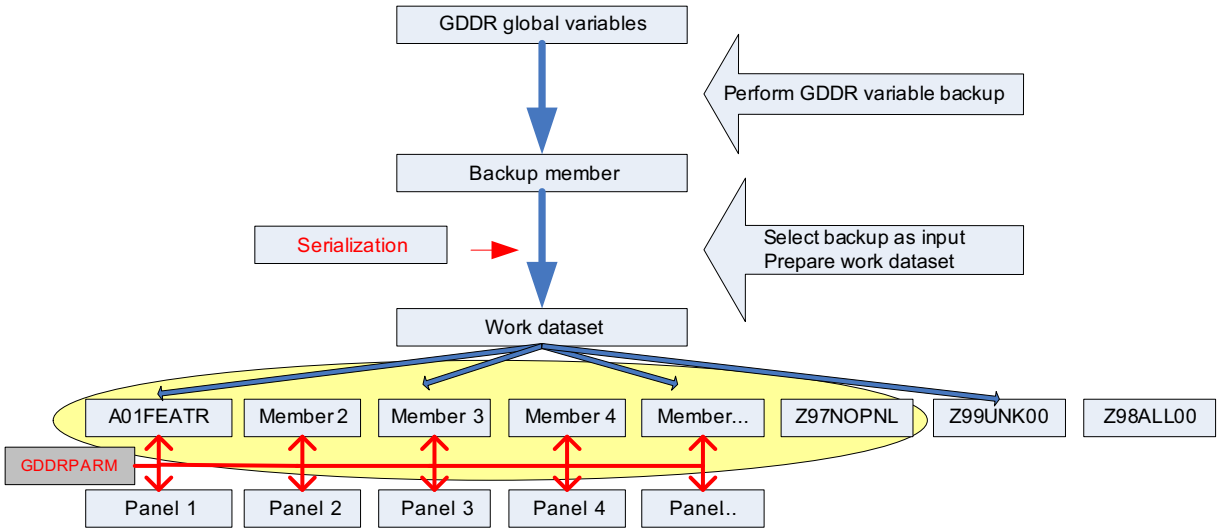



**Figure 20** GDDR Parameter Wizard workflow—step 4

## Step 5: Validate

The Validate function assembles panel-bound work dataset members as well as the Z97NOPNL member into the Z00PARMS member and presents it to the validation engine.

When using the FULL option, all panel-bound members are included. When using the PARTIAL option, only updated members are included. When using the EVENT(RESET) option, the Z97NOPNL member is ignored.

The Validate function performs live queries against the storage environment. Error messages are reported in the Z00VOUT member of the parameter work dataset. The VIEW command on the Validate panel can be used to review these messages.

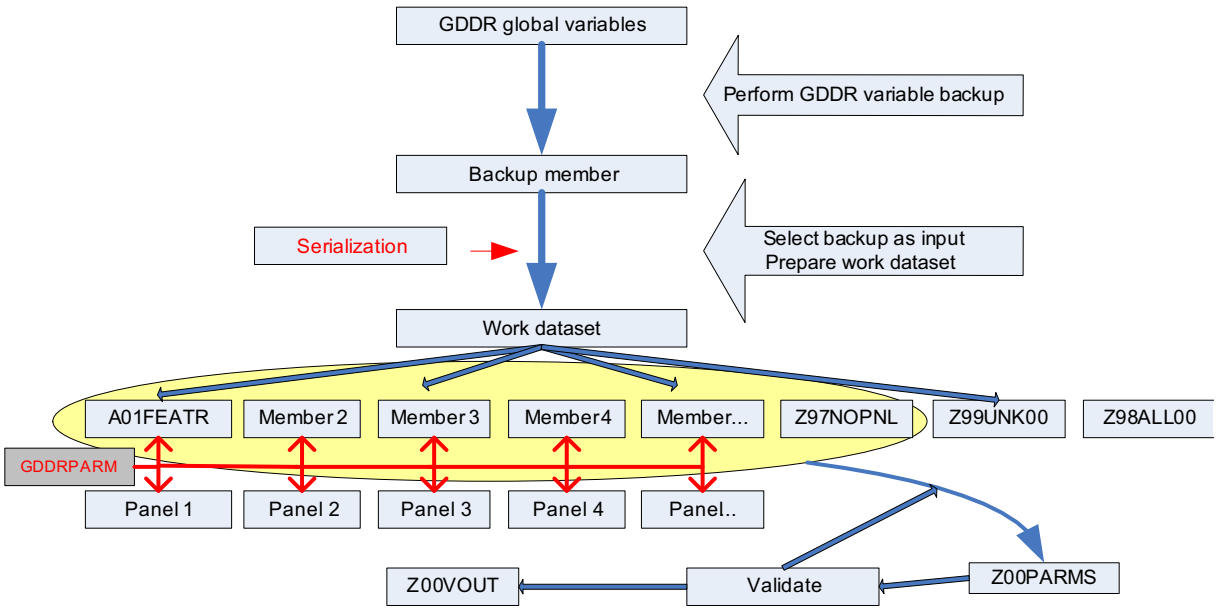



**Figure 21** GDDR Parameter Wizard workflow—step 5

## Step 6: Activate

The Activate function assembles panel-bound work dataset members as well as the Z97NOPNL member into the Z00PARMS member and presents it to the validation engine. If this implicit validation succeeds, it proceeds to the activation steps, which include taking a global variable backup both before and after the activation.

When using the FULL option, all panel-bound members are included. When using the PARTIAL option, only updated members are included. When using the EVENT(RESET) option, the Z97NOPNL member is ignored.

The implicit validation made during activation performs live queries against the storage environment.

Error messages are reported in the Z00AOUT member of the parameter work dataset.

The VIEW command on the Activate panel can be used to review these messages.

The Activate function copies your parameter work dataset to the last activated parameter dataset and makes the new parameter set active on all C-systems. Activate provides options to reset GDDR event variables and clear the GDDR command queue both locally and on other C-systems.

If the GDDR Heartbeat Monitor and GDDR Event Monitor are running, Activate automatically stops them, and restarts them when activation is complete.

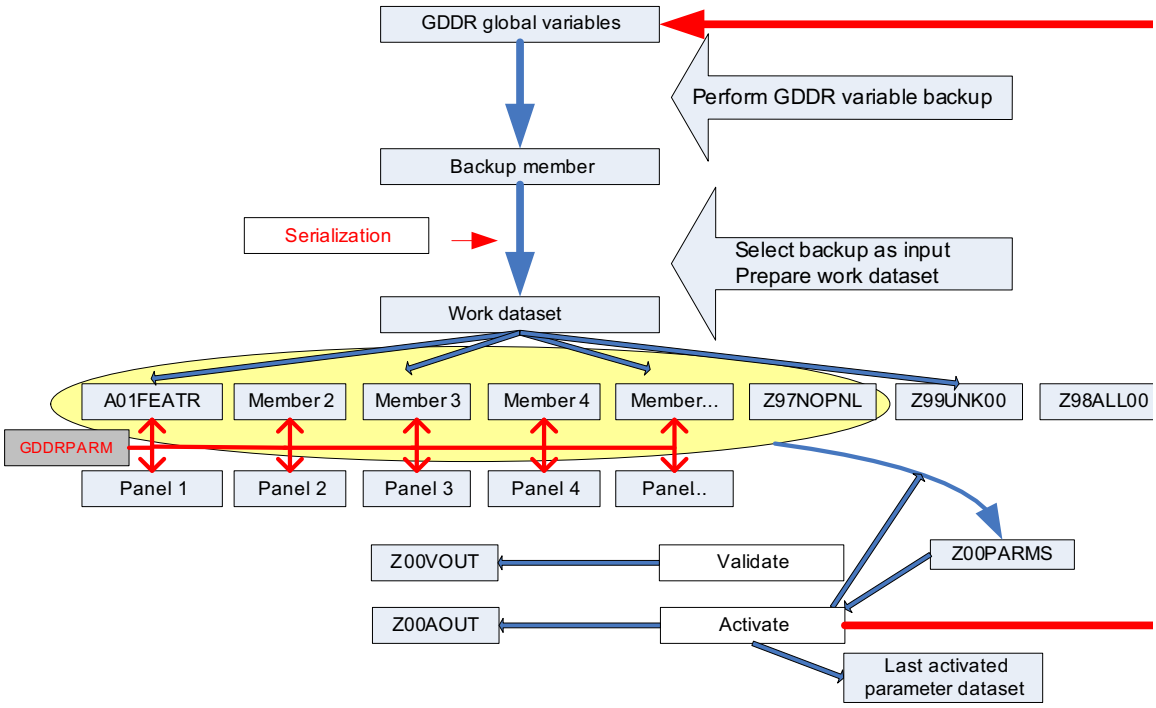



**Figure 22** GDDR Parameter Wizard workflow—step 6

# GDDR environment monitoring

## GDDR monitors

The following monitors run on each C-system:

- GDDR Event Monitor
- GDDR Heartbeat Monitor
- GDDR DLm Heartbeat Monitor

### GDDR Event Monitor

The GDDR Event Monitor (EVM) runs on each C-system to analyze event state changes in which GDDR is interested. On detecting the occurrence of selected events, the GDDR Event Monitor determines what action to take and prompts operators with the appropriate choices.

The GDDR Event Monitor verifies the status of SRDF, ConGroup, and MSC operation at a user-defined interval. GDDR produces messages for integration with user automation that indicate when a GDDR event changes state.

**Note:** "What is a GDDR event" on page 63 explains GDDR event state changes. Table 1, "Monitored events," on page 63 lists GDDR-monitored events.

Certain software operating states are monitored and communicated solely through messages. Message rules enable certain messages of interest to be forwarded to managed systems where user automation can then react to the problem.

**Note:** Table 2, "Software state analysis messages," on page 67 lists state analysis messages.

Messages for monitored operating states can be used for the following purposes:

- MSC and ConGroup analysis
- SDDF analysis
- SRDF/A analysis
- SRDF group and link analysis
- Loss of DASD access
- Loss of site

### GDDR Heartbeat Monitor

The GDDR Heartbeat Monitor (HBM) aids the GDDR Event Monitor in determining the status of the GDDR-managed environment. The lack of a heartbeat from a particular C-system is used to determine the state of the C-system and the site.

### GDDR DLm Heartbeat Monitor

The GDDR DLm Heartbeat Monitor regularly verifies that all managed DLm systems are responsive over TCP/IP, in good health and replicating as expected in the current state of the configuration.

## What is a GDDR event

A GDDR event is a change in state of a component that is part of the environment that GDDR is actively monitoring.

A GDDR event can have a state of either TRUE or FALSE.

- If the event has a state of TRUE, it has occurred or is currently occurring. An event that is TRUE is considered an exception.
- If the event has a state of FALSE, it no longer occurs.

The GDDR Event Monitor and GDDR scripts use GDDR events to determine the environment state. A change in state can then:

- Trigger unplanned/takeover processes
- Prevent a planned process from running
- Allow an unplanned process to run

## Monitored events

The GDDR Event Monitor checks for the following storage and system environment events:

**Table 1** Monitored events

| Event | Description |
|---|---|
| **C>>**: C-Ahead<br>**C>>**: DC3 ahead<br>**C>>**: Site C ahead of site B | SRDF/A updates are ahead of SRDF/S. This often occurs as a result of a ConGroup trip. |
| **CAX**: AutoSwap Event | An unplanned event has occurred for one or more devices in an AutoSwap swap group. As a result, AutoSwap has performed, or is performing, a device swap for the swap group. |
| **CFG**: SRDF Device Reconfiguration<br>**CFG**: Defined vs. discovered DASD discrepancy | This is a transient event. An SRDF dynamic device reconfiguration script has added or removed devices from GDDR-managed SRDF groups. Synchronization of GDDR RDF.DEVICES parameters will follow immediately, which will turn off the CFG event.<br>This also occurs whenever the GDDRPCCD script detects a discrepancy between the defined DASD configuration and the discovered DASD configuration.<br>If the CFG event is seen after an SRDF dynamic device reconfiguration script has completed, take the following actions:<br>• Review the joblog of the script, step GDDRACCD, for error messages showing discrepancies between GDDR parameters and the discovered SRDF devices.<br>• Decide whether the GDDR parameters are correct or the existing device configuration is correct.<br>• If the GDDR parameters are correct, run a new SRDF dynamic device reconfiguration script to bring the configuration in line with the parameters. If the configuration is correct, perform GDDR parameter activation, using the GDDRACDD utility to ensure the RDF.DEVICES parameters are in line with the configuration.<br>As long as the CFG event is present, no GDDR scripts are allowed, except for the SRDF dynamic device reconfiguration script. |
| **CGD**: ConGroup Disable | ConGroup has been disabled. |
| **CGT**: ConGroup Trip | ConGroup trip has occurred. |

**Table 1** Monitored events

| Event | Description |
|---|---|
| **CGV**: ConGroup Verify | ConGroup Verify is in progress. |
| **DLM**: Missing DLm Heartbeat<br>**DC*n*.DLM** | Where DC*n* is one of the configured sites. The DLm event signals a problem in TCP/IP communications between GDDR and a DLMDR interface at the site. One or more DLm ACP systems failed to respond to a heartbeat request during a certain time window. Issue the GDDRMAIN DLM command described in "DLM" on page 198 for troubleshooting. |
| **LDR**: Local Disaster | A single site outage has occurred. |
| **LNK.DC*m*VDC*n***: Missing Propagated GDDR Heartbeat | The C-system reporting this event has detected that the C-system at DC*n* did not propagate the heartbeat of the C-system at DC*m*.<br>Investigate the TCP/IP communication status between DC*m* and DC*n*. |
| **MHB**: Missing Heartbeat<br>**MHB.DC*n*** | One or more C-systems failed to issue a heartbeat during a certain time window. |
| **MIM**: Missing Heartbeat MIM<br>**MIM.DC*n*** | |
| **MSC**: SRDF/A Multi-Session Consistency<br>**MSC.DC*n***<br>**DC*n*.DC*m*.MSC** | During SRDF/A check, the GDDR Event Monitor found MSC inactive or not consistent, or found SRDF/A inactive, or an MSC group is in an unexpected status as reported by message GDDR818I (for the specified site pair). It is recommended to review the status of SRDF/A groups. Activate SRDF/A if necessary, then activate MSC.<br>The GDDR Event Monitor monitors for successful MSC initialization, which then clears the MSC event. If the above action does not resolve the issue, check GDDR Event Monitor SYSTSPRT, SCF log, and SRDF Host Component log for additional messages. |
| **MSF**: GDDRMAIN COMM Link Down<br>**MSF**: Unable to communicate between C-systems<br>**MSF.DC*n*** | Communication to the system is not available. |
| **MST**: Master C-system conflict | A master C-system conflict was detected. There is a discrepancy or disagreement between two or more sites regarding the name of the current master C-system. |

**Table 1** Monitored events

| Event | Description |
|---|---|
| **MXS**: Unexpected loss of a system<br>**MXS**: CF Signal Connectivity lost<br>**MXS.DC*n*** | The system name shown after 'MXS' in the event name has been unregistered from CSC[a] communication.<br>Possible causes are as follows:<br>For C-systems and managed systems:<br>• Message SCF0696W has occurred indicating that the named system has been unregistered from CSC communication. This event is cleared when message SCF0690I occurs signaling the system is being registered with CSC communication again.<br>For C-systems:<br>• The GDDR Heartbeat Monitor has found that GDDRMAIN communication to the indicated C-system was not possible.<br>• During a script where an MSC,PENDDROP or MSC,REFRESH was done, GDDRMAIN communication to the indicated system was not possible.<br>For managed systems:<br>• During a GDDR script which starts mainframe workload, GDDRMAIN communication to the named system was not possible. This event is cleared when the GDDR Event Monitor finds that the indicated system has done a GDDRMAIN communication to the C-system where the GDDR Event Monitor is running. This event is also cleared for managed systems in a SYSPLEX when message IXC466I, signaling outbound signal connectivity is established with the named system, is intercepted by GDDRMAIN running on a sysplex partner system and forwarded to a C-system.<br>This event will be cleared once the GDDR Heartbeat Monitor finds GDDRMAIN communication to this C-system is again possible. |
| **NCX**: CAX protection lost<br>**NCX**: ConGroup lost AutoSwap protection | There are no swappable devices in an AutoSwap swap group; that is, the swap group is not armed. |
| **RDF.DC*m*.DC*n***: RDF Directors offline | A GDDR-managed SRDF group defined by site pair DC*m*.DC*n* has been found to have all directors offline. Look for the following messages to determine which directors or SRDF groups are affected:<br>• Message GDDR650W means that the indicated director is offline, inoperable, or has no ports.<br>• Message GDDR649E means that all directors for the indicated SRDF group are either inoperable or offline or disconnected. |
| **RDR**: Regional Disaster | Primary region outage has occurred. |
| **SRA**: SRDF/A Link Down<br>**SRA**: SRDF/A disabled<br>**DC*n*.DC*m*.SRA** | SRDF/A has been disabled (for the specified site pair). |
| **STR**: SQAR Recovery not available<br>**STR**: STAR Recovery not available | SRDF/SQAR recovery is not available. |

a. CSC (Cross-System Communication) is a component of SCF.

When the GDDR Event Monitor detects any of the events listed in Table 1, "Monitored events," and a GDDR planned script is not running, it issues message GDDS027E, for example:

```
GDDS027E GDDR Error: Event GLOBAL.GDDR.DCN.Unplanned.C>> detected,
   value 1
```

Message GDDS027E is issued every GDDR Event Monitor cycle for each unplanned event which was detected during this or any previous GDDR Event Monitor cycle, for each previously set event which has not been cleared. When a GDDR script is running, message GDDS027E is changed to GDDS027W.

A Split Brain condition is declared when the primary and secondary DASD site C-systems have differing CGT and/or CAX event statuses. Message GDDS221E is issued giving the names of the sites and the event reported by one site but not reported by the second site.

```
GDDS221E site-pair SPLIT BRAIN detected: Site site-id reports event
   but Site site-id does not.
```

For the duration of the Split Brain event, GDDR issues message GDDS081I when the master C-system detects events which normally trigger an unplanned script. This continues until the Split Brain condition is resolved.

```
GDDS081I GDDR Bypassing script script-name submit, SPLIT BRAIN
   condition detected
```

Message GDDS222E is issued when the Split Brain condition is resolved.

```
GDDS222E site-pair SPLIT BRAIN condition cleared
```

Corrective action for a Split Brain condition typically involves re-establishing connectivity between the sites reported to have the different event statuses.

## Message interception

GDDR message interception rules are installed on the C-systems and managed systems.

The message interception rules have two primary functions:

- Detect events that GDDR is interested in and set the appropriate GDDR event TRUE or FALSE.
- Detect events that GDDR processes have to wait for (WTOR), and reply as to the success or failure of such event. This determines if a GDDR process proceeds or terminates.

GDDR uses the z/OS MCSOPER facility to monitor the managed systems for messages of interest. The GDDRMAIN tasks which are installed on the C-systems and the managed systems perform the communication function to route message traffic to or from production systems. The arrival of a message at the target production system can be used to trigger an automation rule. Automation rules can be implemented using, for example, Computer Associates OPS/MVS® Event Management and Automation, IBM Tivoli NetView®, or BMC Control-M®. Such rules can be used to start or shut down workloads on the appropriate systems.

## Exception notification

Notification of GDDR Event Monitor or GDDRMAIN status exceptions can easily be implemented using third-party system management software products which perform event notification tasks.

Software state analysis messages that are candidates for user system automation triage or alert actions are described in Table 2. For information about message cause and action, see the corresponding message ID in the *GDDR Message Guide*.

**Table 2** Software state analysis messages (1 of 3)

| Usage | Message | Cause |
|---|---|---|
| Event is Set | GDDS027E | The GDDR Event Monitor has found an event set during this or any previous EVM cycle which has not been cleared yet. |
| | GDDR786E | This message is issued by the GDDR Event Monitor, each time a GDDR event changes state. (OFF to ON, or ON to OFF). It is forwarded by GDDR to managed systems to facilitate custom automation on detection of events. |
| GDDRMAIN analysis | GDDM014W | The storage used for global variables has exceeded the threshold (80%) of the capacity of the DIV. |
| | GDDM015E | An error has occurred writing to the DIV dataset. The message gives the return code and reason code from the DIV SAVE operation. |
| | GDDM031E | The named Z/OS service returned an error. The message lists the return and reason codes. |
| | GDDM058W | The named lock has been set for over 5 seconds. The lock may be “stuck”. |
| | GDDM153I | During HBM initialization, a conflict in master C-system definition has been discovered. HBM stops EVM since it is also impacted by this condition. |
| | GDDM161W | Degraded mode has been set on or off at the indicated system (which is the C-system at the named site). Previously issued messages should indicate why this action was taken. |
| | GDDM162E | An attempt was made to set Degraded mode on or off on the named system (which is the C-system for the named site).<br>However, Degraded mode could not be changed. This is generally because communication could not be accomplished with the named system because it is down or GDDRMAIN is not running. |
| GDDRMAIN analysis | GDDM195E | During MISC processing, a GDDRPARM CPC statement was found that specified BCPII connectivity for a C-system to a CPC. An attempt to verify that connectivity failed with the specified return code information. Issued by GDDRMTIM. |
| SDDF analysis | GDDP441I | The GDDR SDDF Session utility (GDDRSDDF) has completed a run of verifications with the RC indicated in the message.<br>Investigate non-zero return codes. |
| | GDDP448E | The GDDR SDDF Session utility (GDDRSDDF) found devices with invalid SDDF sessions. The message identifies the reason why a session is considered invalid.<br>Alternatively, the utility found devices at risk for a Full PUSH or an unusually large differential at the next CreatePair or ResumePair differential. |
| | GDDP449W | The GDDR SDDF Session utility (GDDRSDDF) found devices for which the SDDF Session time stamps did not change compared to the first run. |
| | GDDP453E | The GDDR SDDF Session utility (GDDRSDDF) lists the number of devices that are considered to be at risk for a Full PUSH on the next CreatePair or ResumePair Differential operation. |

**Table 2** Software state analysis messages (2 of 3)

| Usage | Message | Cause |
|---|---|---|
| MSC analysis in the GDDR Event Monitor | GDDS202E | The ConGroup owner, MSC primary server, or MSC secondary server (as indicated in the message) is not the same on the C-systems in the primary region. |
| | GDDS203E | The ConGroup name reported by MSC does not match the ConGroup name which GDDR believes to be active. |
| | GDDS204W | The MSC server is in an unexpected state (not PRIMARY or SECONDARY). The message shows the current role. |
| | GDDS205E | The current SRDF state is not SRDF/Star with AutoSwap. |
| | GDDS207E | Neither $DC1 nor $DC2 are aware of the primary or secondary (as indicated) MSC server. |
| | GDDS221E<br>GDDS222E | A Split Brain condition has been detected between sites in the primary region, based on the detection of the named event at one site but not at the other.<br>If the condition is resolved, message GDDS222E will be issued. |
| MSC analysis in the GDDR Event Monitor | GDDS225E | The GDDR Event Monitor set the MSC and/or STR event, because when reviewing MSC configuration data, it found an unexpected condition.<br>GDDR and/or Mainframe Enablers software setup and configuration errors are most frequently the cause of this. Examples are:<br>• The ConGroup name defined to MSC Star does not match the one defined to GDDR<br>• The user defined a Concurrent SRDF/Star configuration to GDDR, but a Cascaded SRDF/Star configuration in SRDF Host Component parameters<br>• ConGroup is disabled in a Star configuration<br>• A Star configuration is defined to GDDR, but MSC reports MSC-mode<br>• In a pair of SQAR-associated groups, MSC has different associated group names than those defined to GDDR<br>• In a pair of SQAR-associated groups, the SQARA and SQARB groups report a different primary MSC server SMFID<br>Another common cause is that the MSC group has not been initialized sufficiently, or took an error during initialization. |
| | GDDS227E<br>GDDS228E | The GDDR Event Monitor reports a delay in MSC SW activity for the named MSC group. The reported MSC Post Timestamp is older than 2 times MSC_CYCLE_TARGET seconds. Message GDDS228E reports a return to normal. |
| | GDDS229E<br>GDDS230E | The GDDR Event Monitor reports a delay in MSC cycle switching for the named MSC group. The reported MSC Cycle TOD is older than X times MSC_CYCLE_TARGET seconds.<br>Message GDDS230E reports a return to normal. |
| | GDDS231E<br>GDDS232E | The GDDR Event Monitor reports a delay in MSC SDDF resets for the named MSC group. The reported MSC SDDF RESET TOD is more than X MSC cycles ago. Message GDDS232E reports a return to normal. |
| | GDDS233E<br>GDDS234E | The GDDR Event Monitor reports a delay in MSC SDDF resets for the named MSC group. The reported MSC SDDF RESET TOD is more than X seconds ago. Message GDDS234E reports a return to normal. |
| | GDDR818I | This message indicates that the state of the named MSC group has changed from an earlier recorded state. It is also issued the first time the MSC group state is recorded in the life of a GDDR address space. |

**Table 2** Software state analysis messages (3 of 3)

| Usage | Message | Cause |
|---|---|---|
| Loss of DASD access or loss of site | GDDR850I | The GDDR Event Monitor has detected an unplanned swap in progress. It does not take action until the swap is complete. |
| | GDDR852I | The GDDR Event Monitor has detected that the current secondary site is dead, which indicates a local disaster (LDR) occurred. |
| | GDDR853I | The GDDR Event Monitor at the primary site has detected a local disaster (LDR), which indicates that recovery at the secondary site is no longer available. |
| | GDDR854I | The GDDR Event Monitor at a site other than the primary site has detected a local disaster (LDR), which indicates that recovery at the secondary site is no longer available. |
| | GDDR855I | An RDR event (regional disaster) has been detected. If this message occurs in the primary region, then the secondary region has suffered the disaster; if this message occurs in the secondary region, then the primary region has suffered the disaster and further messages and action should be expected to deal with the situation. |
| SRDF/A monitoring | GDDR592I | A query (local or remote) has been done to determine if SRDF/A is active or not. The message shows the path and SRDF group which were interrogated, as well as other SRDF information. The status (active or not active) is shown. If the status is not active, the Cycle TOD shows the last time that SRDF/A was active on this SRDF group. If MSC is not active on this SRDF group, the Rcv Tag is shown as all dashes (-), and C-Ahead and Host Intervention Required are shown as N/A. |
| | GDDR863W | SRDF/A was found in the named state (Active, Inactive or Suspended) for the named gatekeeper and SRDF group. |
| SRDF links failure | GDDS047E | An SRDF Links failure event has been detected. The GDDR Event Monitor continues to check if other error conditions exist. |
| SRDF group analysis | GDDR649E | The named SRDF group on the indicated storage system is either offline or has no functioning director. All directors associated with the SRDF group are either dead or offline or disconnected. |
| | GDDR650W | The named director on the indicated storage system is not operational for the reason given. Reasons could be Dead, No ports, or Offline. |
| | GDDR652E | The C-system at the named site is found to have no usable gatekeeper devices for the specified storage system, which is expected to be directly channel attached at the named site. The site ID of the storage system can be the same site as that of the C-system or in AutoSwap configurations, a different site in the same region. |
| | GDDR653W | The C-system at the named site is found to have no usable connections to the specified storage system at a different site, which is not expected to be directly channel attached. |
| | GDDR655E | The C-system at the named site was unable to update the SRDF topology map. The most likely cause for this is that the C-system at the named site has no usable gatekeepers for one or more storage systems at its own site. If that is the case, there will be additional symptoms:<br>• The SRDF event will be ON for each site pair involving the named site.<br>• GDDREVM at the master C-system will set to Degraded mode because there are no device workers started at the named site. |

## Expected events

GDDR recognizes *expected* events. An event that is expected to occur as a result of a GDDR action is listed as an expected event. Expected events require no user action.

GDDR adds or removes expected events automatically over the course of the GDDR action. When the action completes or shortly after it, this list would be empty.

Table 3 lists possible expected events.

**Table 3** Expected events

| Event | Description |
|---|---|
| **CGD**: ConGroup Disable | ConGroup has been disabled. |
| **CGT**: ConGroup Trip | ConGroup trip has occurred. |
| **ECA**: Enginuity Consistency Assist | ConGroup has cleared ECA flags. |
| **MSC**: SRDF/A Multi-Session Consistency<br>**DC*n*.DC*m*.MSC** | MSC is inactive or not consistent, or SRDF/A is inactive, or an MSC group is in an unexpected status as reported by GDDR818I (for the specified site pair). |
| **SRA**: SRDF/A Link Down<br>**DC*n*.DC*m*.SRA** | SRDF/A has been disabled (for the specified site pair). |

To view expected events, open the Perform Health Check panel (C) and find the Expected Events field, or run the GDDREE00 utility as described in "GDDR Expected Events utility (GDDREE00)" on page 511.

You can clear the Expected Events list or remove one or more events from the list using the GDDREE00 utility.

## Degraded mode

Degraded mode is set when one or more of the following conditions are met:

- A C-system has a Missing Heartbeat event (MHB). This is indicated in the "Active Events" field of the Perform Health Check panel (C) or GDDRMAIN CHECKUP command output.
- GDDRMAIN is not running on a C-system. The C-system is shown as Inactive in the GDDRMAIN Status field of the Perform Health Check panel (C) or GDDRMAIN CHECKUP command output.
- A GDDRMAIN subtask or worker is not running on a C-system. The C-system is shown as Degraded in the GDDRMAIN Status field of the Perform Health Check panel (C) or GDDRMAIN CHECKUP command output.
- The MPARM results are inconsistent.

If none of the above conditions explain the Degraded mode, review GDDREVM SYSTSPRT. Message GDDR6444I is issued explaining the reason for Degraded mode.

---

**Note:** The Degraded Mode field on the Perform Health Check panel (C) or in the GDDRMAIN CHECKUP command output may indicate No, but the System Communication Status shows GDDRMAIN Status as Inactive or Degraded for one or more managed systems (PSYS).

---

# GDDR licensing options

This section describes the supported license types, licensing controls, and license validation. For instructions on how to install your license, see "Install GDDR Licensed Feature Code" on page 144.

## Supported license types

For the SRDF/SQAR configuration, the following license type is supported:

- GDDR—Provides GDDR management of Dell EMC PowerMax and VMAX systems, including use of the storage systems as a backend for DLm using the DLMDR control interface.

  This license is referred to as the *GDDR for DASD*, *GDDR PowerMax,* or *GDDR VMAX license*.

  The granularity of GDDR operations is all-or-nothing at the site level.

## License validation

During GDDRMAIN initialization, a check is performed to ensure the local system has a valid GDDR license in the SCF initialization file. If no license is detected, GDDRMAIN shuts down with an error message.

# DLm integration using DLMDR

GDDR can manage DLm configurations using the DLMDR control interface. GDDR supports the NASRDF implementation of DLMDR for the SRDF/SQAR configuration. The tape storage uses PowerMax or VMAX as the backend, and this data is part of the Enterprise Consistency Group managed by GDDR, thus providing Universal Data Consistency (UDC). GDDR adds DLMDR script steps to ensure coordinated planned and unplanned actions across DASD and tape data.

Figure 23 illustrates DLm integration using DLMDR at a GDDR site.

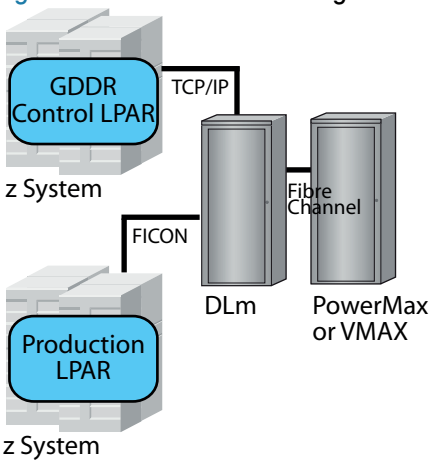



**Figure 23** DLm integration using DLMDR

The following commands are issued from GDDR:

- VG8HALT (null or SEC)
- VG8ACT (null or SKIP_SITEA_SHUTDOWN)
- VG8REST (null or PRI or SEC or SEC,SKIP_SITEA_SHUTDOWN)
- VTECFGSWAP (DUMMY or PROD or DRTEST)
- DEVSTATUS (null or QUIESCE)
- Heartbeat (null)
- VG8BCV (MOUNT or UMOUNT)
- VG8RESET (null)

The following responses are returned from DLm:

- Passed (null or continue or quiesced). Note that GDDR always continues on passed responses.
- Failed (RETRY or ABORT). Note that GDDR always halts the step at that point. This is true for any step that did not succeed (normal GDDR processing).

Note the following when restarting a script after any failure:

- If RETRY, the scripts starts at the step that failed and the operator can either skip that step or restart on that step. All following DLM steps will be executed.

- If ABORT, the Global GDDR ABORT Flag is set on. When operator continues, all DLM steps will be skipped other than VG8RESET. If the operator generates a new script and the GDDR ABORT Flag is on, the DLM steps will not be included in the new script.

Follow the instructions in "Configure DLm support" on page 160 to configure GDDR support for DLMDR.

# SRDF/SQAR High Availability (HA)

SRDF/SQAR HA operation ensures continuous operation of SRDF/SQAR in the event of loss or removal of the primary MSC server.

The SRDF/SQAR HA option is set in the Define GDDR Configuration features panel (M,P,C,F).

When SRDF/SQAR HA is enabled, GDDR planned, unplanned, and resumption scripts which normally achieve SRDF/SQAR mode on completion waits for SRDF/SQAR HA. Planned and unplanned site swap scripts also swap the primary and secondary MSC server roles. The Planned Star-HA takeover (GDDRPA36) script, which is selected from the Special Scripts section of the Select Script to Run on GDDRPLEX panel (S), performs a planned swap of the primary and secondary MSC server roles. GDDREVM submits the Unplanned Star-HA Takeover (GDDRUP35) script in case the primary MSC server is lost unexpectedly.

**Note:** GDDR support for SRDF/SQAR HA is limited to SRDF/SQAR with AutoSwap configurations.

The SRDF/SQAR HA value of '2' indicates that SRDF/SQAR High Availability is supported for planned and unplanned actions.

## Support for MSC THAW processing

MSC THAW processing prevents possible loss of SRDF/A protection as a result of a ConGroup trip.

**Note:** This feature applies to the SRDF/SQAR configuration with GDDR configured for High Availability (HA).

A ConGroup trip causes SRDF/A sessions to freeze. This means that they stop cycle switching, and will soon drop, running out of cache on the R1 side. The primary MSC server must “thaw” the frozen sessions after it has activated the SDDF sessions at Site C and exported the C-ahead flag. However, the cause of the ConGroup trip (that is, loss of inter-site links between Site A and Site B or total loss of Site B) can also cause the primary MSC server running at Site B, to be unable to reach the system at Site A, making it impossible to thaw the frozen SRDF/A sessions.

With MSC THAW processing enabled, GDDR issues the MSC,THAW command at the secondary MSC server wherever appropriate to activate the SDDF sessions and export the C-ahead flag, thus preventing loss of SRDF/A.

You can enable or disable MSC THAW processing using the EVM MSC THAW Processing field in the Specify GDDR Tuning Values panel (M,P,O,T) of the GDDR Parameter Wizard.

To restore normal operations after MSC THAW processing, perform the procedure provided in “Restoring normal operations after MSC THAW” on page 590.

# TimeFinder device management

The GDDR-managed configuration can include TimeFinder devices.

GDDR supports the following local replication technologies:

- SnapVX
- TimeFinder/Clone
- TimeFinder/Mirror

In addition, GDDR provides management of configurations that are protected with zDP Versioned Data Groups (VDG).

For information about each of the local replication technologies, see the *Dell EMC Mainframe Enablers TimeFinder SnapVX and zDP Product Guide, TimeFinder/Clone Mainframe Snap Facility Product Guide*, or *TimeFinder/Mirror for z/OS Product Guide*.

By default GDDR uses legacy TF/Mirror with Clone emulation across all generations of PowerMax and VMAX systems present in the configuration.

**Note:** In an environment where both GDDR and SnapVX are present, non-GDDR users of SnapVX can have a maximum of 254 or 255 snapshot names. GDDR uses its reserved snapshot names for the GOLD and optional TEST set it manages, and these should be subtracted from the SnapVX limit of 256 snapshots per volume.

## What is a BCV in GDDR context

In GDDR documentation, the term 'BCV' references any PowerMax or VMAX device that provides a Business Continuance Volume function. This is not limited to BCV devices in the context of legacy TimeFinder/Mirror replication technology. With support for SnapVX and TF/Clone, GDDR considers any TimeFinder target device as a BCV in the broader sense.

GDDR BCV management is controlled using call overrides to program GDDRKF20.

## GOLD and TEST sets of BCVs

GDDR-managed devices can have multiple BCV devices associated with them. The GDDR Automatic Configuration Discovery utility (GDDRACDD) discovers up to 2 sets of BCVs. To reflect their different GDDR use cases, these sets are named the GOLD set and the TEST set.

The BCV GOLD set is used during GDDR scripts to preserve a golden copy of the data from GDDR-managed R2 devices, prior to an SRDF R1>R2 resynchronization.

The BCV TEST set is intended for test-IPL purposes. When running a GDDR Test-IPL from a BCV script, if a TEST set of BCVs is defined to GDDR for the selected site and you want to exploit this feature, the script splits off this TEST set and does the IPL off of those devices. This leaves the GOLD set of BCVs intact, so it can be used immediately if disaster strikes during a test-IPL scenario.

## TimeFinder device states

Table 4 lists supported device states of source and target devices for the three supported local replication technologies.

**Table 4** Supported TimeFinder device states

| Local replication | Device state | Applies to source or target devices | Comments |
|---|---|---|---|
| SnapVX | CREATE<br>ACTIVATE<br>RESTORE | Source | CREATE is rarely expected in a GDDR environment because all CREATE commands are activated (by ACTIVATE) in the same step.<br>RESTORE means that the source device is linked to its own snapshot. |
| | LINK<br>UNLINK | Target | |
| TF/Clone | ESTABLISH<br>SPLIT<br>RESTORE | Source and target | ESTABLISH is rarely expected in a GDDR environment because all GDDR SNAP VOLUME commands are activated (by ACTIVATE) in the same step.<br>SPLIT is the most common state, meaning that the snap was activated.<br>RESTORE means that a snap was performed where a GDDR-defined BCV is the source and a GDDR-defined SRDF device is the target. |
| TF/Mirror | ESTABLISH<br>SPLIT<br>RESTORE | Target | TF/Mirror target devices must have the Legacy BCV flag. |

Because you can mix TimeFinder technologies at the same site (within the supported limits), multiple states may be required to perform a complete site query.

For example, when you are using SnapVX on a PowerMax, VMAX All Flash, or VMAX3 system and TimeFinder/Clone on a VMAX 10K, 20K, or 40K system:

- STATE(SPLIT,ACTIVATE) queries TimeFinder/Clone source and target devices and SnapVX source devices.
- STATE(SPLIT,ACTIVATE,LINK) queries TimeFinder/Clone and SnapVX source and target devices.

If you are using both TimeFinder/Clone and SnapVX on a PowerMax, VMAX All Flash, or VMAX3 system:

- STATE(ACTIVATE,LINK) queries SnapVX source and target devices.
- STATE(SPLIT) queries TimeFinder/Mirror target devices. In this case, only one state is required.

**Note:** This example assumes that you did not request SnapVX softlinking support in the Define GDDR Configuration features panel (M,P,C,F) and specified TimeFinder/Mirror as the legacy TimeFinder method.

## SnapVX softlinking support

GDDR supports SnapVX configurations where a snapshot can be linked and unlinked to multiple target devices.

The following settings are used to enable this feature:

- The SOFTLINK parameter of the GDDR Automatic Configuration Discovery for DASD (GDDRACDD) utility

  By specifying a GDDRACDD SOFTLINK parameter with value YES for a site, you can instruct GDDR to use SnapVX with softlinked targets on PowerMax, VMAX All Flash, or VMAX3 systems for that site. GDDR then uses TF/Clone Mainframe Snap Facility on VMAX 10K, 20K, or 40K systems on that same site.

- The SNAP-VX SOFTLINK Support field in the Define GDDR Configuration features panel (M,P,C,F) of GDDR Parameter Wizard, as described in "Define configuration features (M,P,C,F)" on page 302.

This feature requires Mainframe Enablers 8.0 or later.

## Targetless SnapVX support

GDDR supports SnapVX configurations where no target devices are defined for GDDR-managed snapshots.

You can select sites for GDDR to discover and configure SnapVX targetless devices using the SNAP-VX is TARGETLESS field in the Define GDDR Configuration features panel (M,P,C,F) of the GDDR Parameter Wizard, as described in "Define configuration features (M,P,C,F)" on page 302.

During configuration discovery, the Automatic Configuration Discovery for DASD utility (GDDRACDD) discovers GDDR-managed SRDF devices that are protected by a GDDR-managed snapshot, and define those to GDDR using STDBCV parameters where the "STD" and "BCV" device ranges are identical.

---

**Note:** To enable discovery of targetless configuration by GDDRACDD, specify the TGTLESS keyword on the SOFTLINK parameter of the utility.

---

At script runtime, the GDDR BCV management program GDDRKF20 performs SnapVX CREATE and ACTIVATE commands for the SnapVX targetless devices. No SnapVX LINK commands are performed, since no target devices are defined.

If a site is configured with targetless SnapVX support, scripts GDD2P01A (Perform Test IPL from BCVs) and GDD2P02A (Resume after TEST IPL from BCVs) are blocked since they would require linked target devices to IPL from.

# zDP integration

Integration between GDDR and zDP includes script steps to prevent zDP from causing SC VOL command failures in GDDR scripts and ensure that zDP activity is restored at the end of GDDR scripts.

zDP obtains device locks when a Versioned Data Group (VDG) is started. These device locks cause #SC VOL commands that are issued against the same devices to fail. They are released when the VDG is terminated or paused. A zDP RELDLOCK command is provided to ensure that locks are released in unplanned scenarios.

At the start of scripts, script actions are added to PAUSE or RELDLOCK VDGs defined against GDDR-managed SRDF devices at one or more sites where the script issues SC VOL commands, as well as the sites paired with them using SRDF. These active or stopped VDGs are automatically discovered at runtime.

VDGs in an active state are PAUSED so they can later be RESUMED. The script issues a RELDLOCK command against VDGs in a stopped state to ensure that no residual device locks are left.

The discovery and the resulting actions are done at all C-systems in the same region as the site where the devices are located, as needed based on run-time discovery.

Towards the end of the script, GDDR automatically ensures the VDGs for which it issued the PAUSE command are brought back to the Active state, using either a RESUME command or a START command, depending on the state discovered at runtime. Any VDGs defined to GDDR (in the Define GDDR Managed VDG names ISPF panel of the GDDR Parameter Wizard) are also brought to the Active state.

You can enable or disable GDDR zDP management using the GDDR zDP Management field in the Define GDDR Configuration features panel (M,P,C,F).

You define the GDDR-managed VDGs using the Define GDDR Managed VDG names ISPF panel as described in "Define GDDR-managed VDGs (M,P,D,V)" on page 313.

---

**Note:** GDDR does not support a VDG that is defined to zDP on multiple sites.

---

## zDP call overrides

The following call overrides control the status of zDP processes during script execution:

- GDDRZDP0 Perform zDP PAUSE and RELDLOCK Actions
- GDDRZDP0 Perform zDP RESUME and START Actions

Similar control is provided on the GDDRMAIN,SCRIPT command using the ZDPPAUSE and ZDPRESUME keywords.

It is recommended to set both options to **Y** by default. However, at runtime, for the first script of a 2-script sequence (GDD2P03A, GDDRPA27, GDD2P17A, GDDRPA60), it may be desirable to override this default for the zDP RESUME and START actions to **No**. These scripts halt SRDF replication, and a resumption of VDGs at the end of those scripts causes the VDG to create new snapsets of the same data image over and over. A possible reason for doing the RESUME at this point could be to preserve snapshots of the data created while workload is activated on the target site.

The following call overrides let you suppress resumption of zDP processes during unplanned scripts based on the site role:

- GDDRZDP0 Perform zDP Unplanned RESUME at SITEA
- GDDRZDP0 Perform zDP Unplanned RESUME at SITEB
- GDDRZDP0 Perform zDP Unplanned RESUME at SITEC
- GDDRZDP0 Perform zDP Unplanned RESUME at SITED

---

**Note:** "GDDR sites and regions" on page 41 explains the site roles and designations. Table 5, "GDDR call overrides," on page 85 provides more details about making the call override settings.

---

These call overrides can be used to prevent resumption of VDGs at a site where the site data is not being updated, or to preserve copies from before an unplanned event.

For example, an unplanned swap event would stop SRDF/A replication from DC1 to DC3, and the unplanned swap script (GDDRUP31 with SRDF/A preference) does not resume this replication. In this case, you do not want to resume VDGs at DC3 at the end of this script, so you set the Perform zDP Unplanned RESUME at SITEC call override to **N**.

Similar control is provided on the GDDRMAIN,SCRIPT command using the ZDPUPLA, ZDPUPLB, ZDPUPLC, or ZDPUPLD keywords.

## GDDR multi-tenancy for TimeFinder

When managing a TimeFinder environment, GDDR ensures that a site-level consistent restartable image of the data exists on the BCVs. GDDR Multi-Tenancy for TimeFinder provides system-level isolation that is needed when an environment, which is normally managed by GDDR at the site level, is shared by multiple tenants.

When performing test IPL from BCVs using the GDD2P01A script, you can select one or more systems for GDDR to SPLIT only BCVs relevant to those systems from the source devices.

However, during the resumption script run after the test IPL, all BCV devices are REESTABLISHed, in order to come back to site-level operations.

It is recommended to use the TEST set of BCVs for test IPLs for a selection of systems, so that data integrity continues to be preserved at the site level on the GOLD set of BCV devices.

"Configure GDDR Multi-Tenancy for TimeFinder" on page 158 explains how to implement GDDR Multi-Tenancy for TimeFinder. The BCVGROUP dataset described in "Allocate BCVGROUP dataset" on page 159 defines a group of BCVs by listing SRDF devices subject to GDDR Multi-Tenancy for TimeFinder processing.

The feature is controlled using the GDDRKF20 'Use BCVGROUP data set' call override.

System selection is accomplished using the following GDDR ISPF panels:

- Select System(s) for DC*n* Restart—Allows selection of one, multiple or all supported systems. The script uses the information about which systems are selected for inclusion in the test IPL to select BCV devices to be split, based on the associations of BCV devices with systems in the BCVGROUP dataset.
- Select System(s) for DC*n* Reset—Allows you to make the GDDR script reset all supported systems, one system, a number of systems, or the exact list of systems that were used during the corresponding test IPL script (by selecting the *IPLed* entry).

  **Note:** The Resume after Test-IPL from BCVs at DC*n* script re-establishes all BCVs at the affected site, regardless of the systems selected for Reset in the Select System(s) for DC*n* Reset panel.

When submitting a script using the GDDRMAIN SCRIPT command, the BCVGROUPS parameter specifies the BCVGROUP dataset and the SELSYS parameter lists the selected systems.

# GDDR support for external devices

GDDR always manages devices under consistency protection. This can be ConGroup protection, MSC protection, or SRDF/SQAR protection. GDDR support for external devices adds appropriate commands to GDDR scripts to manage SRDF groups outside of the GDDR-managed consistency protection. GDDR support for external devices enhances automated restart at a site which is an SRDF/A target by adding management of various types of volumes (for example, page volumes and temp pool volumes) which are not typically replicated with consistency protection, due to various considerations such as bandwidth constraints or the fact that they simply do not have the consistency requirement and should not trigger swap or trip events.

The SRDF device actions to be performed on the external devices are “make R2 R/W” or “make R1 TNR”, depending on the script and the site role.

**Note:** See “Configure GDDR support for external devices” on page 160 for instructions on how to configure GDDR support for external devices.

SRDF/SQAR configurations do not support “external” devices on the SRDF/S leg.

# BCPii support

## Introduction to BCPii

The Base Control Program internal interface (BCPii) is the interface to the IBM Mainframe Service Element (SE) and the Hardware Management Console (HMC) used to communicate with the Central Processing Complex (CPC) SE. BCPii provides for a direct link through the operating system to the SE associated with the local CPC or to a remote CPC using the existing HMC LAN network as described in "BCPii HMC networking capabilities and requirements" on page 82.

The BCPii interface provides a Connect and Disconnect function to objects associated with the mainframe CPC. Various functions can then be performed against those objects across the established connections. The relevant object types are CPC, Image, Capacity Record, and the major HMC profile types of Image, Load, and Reset. The connection type determines the functions available to the caller. GDDR supports CPC, Image, and Load Activation Profile connections.

BCPii also provides a Command function enabling z/OS HMC command functions to be entered against the connected-to object. For example, for a CPC connection, command options such as Activate and Deactivate can be requested. For an Image connection, command options such as Load, Reset Clear, z/OS Operating System Commands, and others can be issued.

BCPii also provides List and Query functionality against the object it is connected to, enabling retrieved information to be used in determining the logic path followed in GDDR. The List function lists all objects of a given type that is associated with the HMC network. A list of CPCs can be obtained that are within the HMC Grouping associated with the local CPC on whose image/LPAR the BCPii List functions are executed. "BCPii HMC networking capabilities and requirements" on page 82 provides more information about this topic. A list of Images associated with a connected-to CPC can also be requested. Similar lists are available for the other connection types. The Query function returns setting and information specific to the connected-to object enabling the caller to query setup and operational settings for the CPC, the Image, the Profiles, and all other object types. Complementing this Query function is the Set function enabling modification of certain settings within the object. Load parameters for an Image can be changed. LPAR weightings within a CPC can be changed. There are many options available for modification.

**Note:** To configure BCPii, follow the instructions provided in "Configure BCPii" on page 114.

## BCPii HMC networking capabilities and requirements

The Service Element (SE) is a specialized laptop in the CPC used to setup, maintain, and control system objects on that CPC. The HMC is an external PC running specialized code that communicates with the SE for all affiliated CPCs enabling, for the most part, the same level of control against the objects associated with any listed CPC but without requiring physical proximity to the CPC. An HMC can control more than a single CPC, thus introducing the concept of CPC grouping. A CPC group is a defined list of CPCs that an HMC can address.

BCPii works with the local SE. All requests that may be resolved locally do not involve any HMC with LIC Enabled status. Each SE affiliates with an HMC which introduces that SE to the CPC group. If more than one HMC has the SE in the CPC group, and all such HMCs have LIC Enabled status, then, for a remote HMC request, the first HMC to get the tap from the SE is the one with which BCPii communicates.

BCPii can only talk to those remote SE/CPCs that are defined in the CPC group on the HMC with which the local SE is affiliated. If multiple HMCs are defined to the data center LAN, it is up to that data center to ensure all SE/CPCs are defined to the CPC group for each HMC. If all CPCs are defined in each CPC group, that is, each HMC knows about all CPCs in the network, then BCPii on one system may reach any SE in the network. Contrary to this, the data center may want to fence certain SE/CPCs from BCPii interaction using separate HMCs with unique CPC groupings. The routing for a remote BCPii request involves sending the request to the local SE, then on to an affiliated HMC and then on to the SE of the remote CPC for which the original BCPii request was created. Routing for a local request is directly to the SE for that local CPC and no further.

As the CPC is affiliated with an HMC network, its fully qualified name is comprised of two network related elements, the NETID and the Network Addressable Unit (NAU). These two elements, separated by a period (.), are required when establishing the communications link between the BCPii address space on the GDDR B-system and the SE affiliated with the target of the BCPii request.

All GDDR interfaces (ISPF screens, control blocks, and so on) use the NETID component of the CPC name. When the time comes to perform an HMC function against the LPAR/CPC, the fully qualified CPC name (NETID.NAU) is used.

**Note:** You can test connectivity to all CPCs using the GDDRBPCI utility described in "GDDR BCPii Connectivity Test utility (GDDRBCPI)" on page 505.

## Additional routing capabilities

In addition to the SE/HMC network routing capabilities, GDDR provides additional routing capabilities.

If a GDDR code element requires a BCPii action against a particular CPC and does not have SE/HMC network access to the target CPC, it can route the request to another system. In GDDR documentation, that other system is referenced as the *B-system* for the target CPC, as described in "B-systems" on page 83.

The GDDRMAIN CPC parameters enable you to inform GDDR which systems have SE/HMC access to the CPCs GDDR is expected to manage.

## B-systems

There is no requirement for GDDR BCPii work to be done on a C-system. It can be done on a C-system or production system that meets certain requirements—consider this to be a *B-system* for BCPii work. A B-system is not a new GDDR system type beyond C-systems or P-systems, but is a functional role that can be assigned to either a C-system or a P-system.

A B-system is any GDDR system (C-system or managed system) which is named on a CPC statement in the GDDRPARM file. The GDDWXH worker and the MISC subtasks of MISCBS*nn* are applicable only to B-systems.

"Specify CPC parameters" on page 116 provides instructions and examples on CPC parameter definition.

# CPC Recovery—LPAR Recovery

CPC Recovery and LPAR Recovery features deliver system recovery capabilities for Central Processor Complexes (CPC), protecting a single system or all systems within the complex. CPC Recovery is enabled for all or selected processor complexes within a site as soon as LPAR Recovery is defined for one of the systems which has an LPAR defined with the CPC. Protection is accomplished through monitoring for system outage trigger messages indicating a system has unregistered from a common communication facility, in this case, Dell EMC Symmetrix Control Facility (SCF).

The CPC Recovery functionality complements the production/contingency protection mechanism and entails the following:

- Contingency systems are no longer required to be defined for managed systems.
- A managed system can have either a contingency system or a recovery LPAR defined, or both.
- A C-system can have a recovery LPAR defined.
- A system that is protected with LPAR Recovery is specified with a “home” and “recovery” LPAR location (“away”).
- An LPAR within a CPC can be either the normal LPAR or a recovery LPAR for a system, or both (normal LPAR for system A, recovery LPAR for system B).
- The recovery LPAR can be in the same CPC hosting the system at home.
- The recovery LPAR can be in a different CPC in the system’s “home” data center.
- If the LPAR location of a system A is the recovery LPAR of a system B, then system A cannot have a recovery LPAR defined.
- If a system A has an LPAR defined for it which is the recovery LPAR for a system B, then system A cannot be a contingency system.
- If a system is protected with a recovery LPAR at an “away” site different from its “home” site, then it cannot have a second LPAR parameter for that “away” site. Once this restriction is honored, LPAR Recovery protected systems can have one LPAR per site in the configuration.
- If a system is not protected with LPAR recovery, then it can have one LPAR parameter defined for each site in the configuration.
- The SITE.<*sysname*> parameters for systems which do not have a recovery LPAR defined change when these systems are IPL'd in an LPAR on a different site when they are being displaced by the recovery of an LPAR Recovery protected system. For systems protected with LPAR Recovery, the site does not change, but a separate DCN.LOCATION.<*sysname*> parameter is switched between values “home” and “away”.

# GDDR script call overrides

Call overrides let you alter behavior when running a GDDR script.

You set call overrides as follows:

- For a single script to run, use the Specify Call Overrides for Script panel (as described in "Running scripts via GDDR ISPF interface" on page 524) or specify the corresponding keywords on the GDDRMAIN,SCRIPT command.
- To set default values of call overrides for all GDDR scripts, use the Specify Default Script Call Overrides panel (M,P,O,O) in the GDDR Parameter Wizard.

Table 5 lists call overrides and their meanings. Note that all call overrides do not apply to all scripts. For any script you run, you can set a subset of the possible call overrides.

**Table 5** GDDR call overrides (1 of 4)

| Program | Function | Description |
|---|---|---|
| GDDRRDF0 | DYNAPI - DeletePair and Half-DeletePair Commands | Call override byte 4<br>Specify **Y** to perform SRDF Host Component SC VOL DELETEPAIR and HDELETEPAIR commands using the GDDR command queue. They will be executed in parallel at the storage system level.<br>Specify **N** to suppress GDDR command queue usage for all DELETEPAIR and HDELETEPAIR commands. They will be executed one command at a time. |
| GDDRCL00 | Perform SDDF Session Cleanup | Enables parallel execution of SDDF cleanup in the background, while other script steps are running. |
| GDDRGF08 | Use ConGroup Shutdown/Startup instead of Refresh | Performs shutdown/startup of Dell EMC Consistency Group STC on all z/OS systems. |
| GDDRGFHS | Allow LOAD CLEAR when Activating LPARs in Scripts | Let GDDR scripts perform LOAD CLEAR when activating LPARs. |
| GDDRGFHS | Allow Primary Site HMC Actions in RDR Scripts | Allows the listed scripts to perform an ACTIVATE and LOAD of LPARs at DC3, for systems currently defined at the primary DASD site.<br>• Recover at DC3 after RDR at DC1 and DC2 (GDDRPA05)<br>• Restart production at DC3 SRDF/A to DC1/DC2 (GDDRPA06)<br>• Recover at DC3 after LDR at DC1 SRDFA to DC2 (GDDRPA07) |
| GDDRGFHS | Allow Primary Site HMC Actions in Test Scripts | Allows the listed scripts to perform an ACTIVATE and LOAD of LPARs at the DC3 site, for systems currently defined at the primary DASD site:<br>• Perform test IPL from BCVs at DC2/DC3/DC4 (GDD2P01A)<br>• Perform test IPL from R2s at DC3/DC4 (GDDRPA27)<br>Allows the listed scripts to perform a RESET and DEACTIVATE of LPARs at the DC2/DC3/DC4 site, for systems currently defined at the primary DASD site:<br>• Resume after test IPL from BCVs at DC2/DC3/DC4 (GDD2P02A)<br>• Resume SRDF/A after test IPL at DC3/DC4 (GDDRPA28) |

**Table 5** GDDR call overrides (2 of 4)

| Program | Function | Description |
|---|---|---|
| GDDRGFHS | Allow Primary Site HMC Actions in Unplanned Scripts | Allows the listed scripts to perform a RESET and DEACTIVATE of LPARs currently defined at the primary DASD site:<br>• Recover after unplanned swap (GDDRUP31)<br>• Recover after loss of *DC1* (LDR) (GDD2U13A) |
| GDDRGFHS | Allow Secondary Site HMC Actions in Planned Scripts | Allows the listed scripts to perform an ACTIVATE and LOAD of LPARs currently defined at the secondary DASD site:<br>• Swap production from *DC1* to *DC2* (GDDRPA42) |
| GDDRGFHS | Allow Secondary Site HMC Actions in RDR Scripts | Allows the listed scripts to perform an ACTIVATE and LOAD of LPARs at DC3, for systems currently defined at the secondary DASD site.<br>• Recover at DC3/DC4 after RDR in primary region (GDDRPA05)<br>• Restart production at DC3 SRDF/A to DC1/DC2 (GDDRPA06)<br>• Recover at DC3 after LDR at DC1 SRDFA to DC2 (GDDRPA07) |
| GDDRGFHS | Allow Secondary Site HMC Actions in Test Scripts | Allows the listed scripts to perform an ACTIVATE and LOAD of LPARs at the DC2/DC3/DC4 site, for systems currently defined at the secondary DASD site.<br>• Perform test IPL from BCVs at DC2/DC3/DC4 (GDD2P01A)<br>• Perform test IPL from R2s at DC3/DC4 (GDDRPA27)<br>Allows the listed scripts to perform a RESET and DEACTIVATE of LPARs at the DC2/DC3/DC4 site, for systems currently defined at the secondary DASD site.<br>• Resume after test IPL from BCVs at DC2/DC3/DC4 (GDD2P02A)<br>• Resume SRDF/A after test IPL at DC3/DC4 (GDDRPA28) |
| GDDRGFHS | Allow Secondary Site HMC Actions in Unplanned Scripts | Allows the listed scripts to perform an ACTIVATE and LOAD of LPARs currently defined at the secondary DASD site.<br>• Recover after unplanned swap (GDDRUP31)<br>• Recover after loss of *DC1* (LDR) (GDD2U13A) |
| GDDRGFHS | Confirm GDDR Script HMC actions by WTOR | When GDDR HMC actions in scripts are allowed per the GDDRGFHS call overrides above, **Y** for this call override causes a WTOR to be issued for each action so the operator can allow or deny the action.<br>Specify **P** for Priority, meaning that WTORs for HMC actions during scripts will be reduced to 1 per priority level. The usual confirmation WTORs for each individual action will then be suppressed.<br>Specify **N** for this call override to suppress these WTORs, thus automatically allowing all GDDR HMC actions in scripts, within the confines determined by the GDDRGFHS call overrides above.<br>It is recommended to specify this call override as **P**. A setting of **Y** causes many more WTORs during scripts. A setting of **N** is only recommended for isolated test environments. |
| GDDRGFHS | Allow Primary Site HMC Actions in Planned Scripts | Allows the listed scripts to perform a RESET and DEACTIVATE of LPARs currently defined at the primary DASD site:<br>• Swap production from *DC1* to *DC2* (GDDRPA42) |
| GDDRHMC2 | Prompt for Alternate SYSRES usage | Determines whether to issue prompts for alternate SYSRES usage. |
| GDDRHMC2 | Use DR-Test IPL parameters | Determines whether to use DR-Test IPL parameters. |

**Table 5** GDDR call overrides (3 of 4)

| Program | Function | Description |
|---|---|---|
| GDDRKF0C | Trigger Production Workload Stop | Enables to stop production workloads during GDDR script execution. |
| GDDRKF0E | Transfer AutoSwap Ownership | Adds a step to site swap scripts to transfer the AutoSwap owner role to the appropriate site. In SRDF/Star configurations, the primary MSC server role will be transferred to the same site. |
| GDDRKF0H | Transfer Master Function Ownership | Add a step to site swap scripts to transfer the master C-system function to the appropriate site. |
| GDDRKF0I | Trigger Production Workload Startup | Enables to start production workloads during GDDR script execution. |
| GDDRKF20 | BACKGROUNDCOPY for TF/Clone | Determines whether to use BACKGROUNDCOPY for TimeFinder/Clone operations. |
| GDDRKF20 | Manage external BCV devices | Enables to manage external BCVs during GDDR script execution. |
| GDDRKF20 | Simulate BCV Actions | Performs actions in simulation mode.<br>For TimeFinder/Mirror actions, the commands are echoed to the script joblog.<br>For TimeFinder/Clone and SnapVX actions, EMCSNAP is invoked with the specification of GLOBAL TYPRUN(SCAN). |
| GDDRKF20 | Use BCVGROUP data set | Specify **Y** to have the GDD2P01A script selectively SPLIT BCVs associated with the systems selected for the script. |
| GDDRKF20 | Manage BCVs at DC1/DC2/DC3/DC4 | When set to **Y**, performs TimeFinder/Mirror, TimeFinder/Clone, and/or SnapVX actions at the indicated site (as applicable). Set this to **N** if you do not have BCVs configured at DC1. |
| GDDRKF20 | Manage TEST BCVs at DC1/DC2/DC3/DC4 | Enables to perform actions against test BCVs at the indicated site (as applicable). |
| GDDRRDF0 | Call DYNAPI Interface | Call override byte 2<br>Specify **Y** to:<br>• Perform SRDF device action commands using the GDDR command queue, enabling parallel command execution.<br>• Perform the SRDF Host Component SC VOL CREATEPAIR, RESUMEPAIR, DELETEPAIR, HDELETEPAIR, SWAP, HSWAP, R22SWTCH, MOVEPAIR, HMOVEPAIR, CASCRE, CASDEL, CASSUSP, CASRSUM, and CASSWAP commands in parallel at the storage system level.<br>Call overrides 3-7 further control GDDR command queue usage by command type.<br>Specify **N** to suppress all GDDR command queue usage. Commands will be issued one at a time. Call overrides 3-7 are ignored. |
| GDDRRDF0 | DYNAPI - CreatePair and ResumePair Commands | Call override byte 5<br>Specify **Y** to perform SRDF Host Component SC VOL CREATEPAIR and RESUMEPAIR commands using the GDDR command queue. They will be executed in parallel at the storage system level.<br>Specify **N** to suppress GDDR command queue usage for all CREATEPAIR and RESUMEPAIR commands. They will be executed one command at a time. |

**Table 5** GDDR call overrides (4 of 4)

| Program | Function | Description |
|---|---|---|
| GDDRRDF0 | DYNAPI - SRDF/S Commands | Call override byte 3<br>Specify **Y** to perform SRDF device actions against SRDF/S devices using the GDDR command queue.<br>Specify **N** to suppress usage of the GDDR command queue for SRDF/S devices.<br>In both cases, call overrides 4 and 7 further control GDDR command queue usage. |
| GDDRRDF0 | DYNAPI - SRDF/A Commands | Call override byte 6<br>Specify **Y** to perform SRDF device actions against SRDF/A devices using the GDDR command queue.<br>Specify **N** to suppress usage of the GDDR command queue for SRDF/A devices.<br>In both cases, call overrides 4 and 7 further control GDDR command queue usage. |
| GDDRRDF0 | DYNAPI - Swap and Half-Swap commands | Call override byte 7<br>Specify **Y** to perform SRDF Host Component SC VOL SWAP and HSWAP commands using the GDDR command queue. They will be executed in parallel at the storage system level.<br>Specify **N** to suppress usage of the GDDR command queue for all SWAP and HSWAP commands. They will be executed one command at a time. |
| GDDRRDF0 | Manage external SRDF devices | Not applicable |
| GDDRXACT | Perform SQAR Single Region Restart | Enables to perform single-region restart. |
| GDDRXACT | Perform SQAR Single Side Restart | Enables to perform single-side restart. |
| GDDRXDRV | Manage Distributed Workload | Enables to manage distributed workloads during GDDR script execution. |
| GDDRZDP0 | Perform zDP PAUSE and RELDLOCK Actions | Enables to pause zDP processes and release device locks.<br>**Note:** See "zDP call overrides" on page 78 for an overview of zDP call overrides. |
| GDDRZDP0 | Perform zDP RESUME and START Actions | Enables to start or resume zDP processes.<br>**Note:** See "zDP call overrides" on page 78 for an overview of zDP call overrides. |
| GDDRZDP0 | Perform zDP Unplanned RESUME at SITEA/SITEB/SITEC/SITED | Allows (**Y**, default) or suppresses (**N**) VDG resumption at the indicated site (as applicable) at the end of unplanned scripts, based on the site role during script generation.<br>**Note:** Script GDDRUP31 (Recover after unplanned swap) with SRDF/S preference resumes all SRDF operations regardless of this setting.<br>The Perform zDP Unplanned RESUME at SITEA/SITEB/SITEC/SITED call override settings are in effect only when the GDDRZDP0 Perform zDP RESUME and START Actions call override is set to **Y**. Otherwise, the Perform zDP Unplanned RESUME at SITEA/SITEB/SITEC/SITED call override is forced to **N**.<br>**Note:** See "zDP call overrides" on page 78 for an overview of zDP call overrides. |

# GDDR user message automation

GDDR user message automation (UMA) enables GDDR to intercept and react to any z/OS message, except for GDDR-produced messages.

**Note:** GDDR message interception is described in "Message interception" on page 66.

You can specify a message ID and the action to take upon message interception using the MSG statements in the GDDRPARM file.

**Note:** GDDRPARM MSG statements are described in "MSG ENABLE|DISABLE" on page 267 and "MSG ADD|MODIFY" on page 268. Possible actions are listed in "MODIFY actions" on page 268.

In particular, GDDR can run a user-supplied REXX program upon interception of a certain message ID. For this case, GDDR provides three functions to extract information about the intercepted message. These functions can be used as follows:

`result = GDDRXMSG('OSYS')`

Returns the name of the system where the message originated.

`result = GDDRXMSG('LINES')`

Returns the number of lines comprising the message.

`result = GDDRXMSG('LINE`*n*`')`

Returns the text comprising the nth line of the message.

# GDDR user exits

GDDR provides the following exit points that you may use to augment GDDR facilities:

- GDDRUX01
- GDDRUX02
- GDDRUX03
- GDDRUX04

User exits must be written in TSO/E REXX. Consult the *z/OS V1R11.0 TSO/E REXX Reference*, SA22-7790-08, for programming assistance. A special procedure is provided below for using compiled CA-OPS/MVS REXX.

In the exit descriptions that follow, all parameters are positional within a single REXX argument. That is, the arguments are accessed by a REXX instruction such as:

```
parse arg parm1 parm2 . . .
```

### Sample procedure to use interpreted REXX user exits

1. Create a user-partitioned dataset patterned after the *hlq*.GDDR*vrm*.RCXFE library to contain user-customized exit modules.
2. Concatenate this new library behind the *hlq*.GDDR*vrm*.RCXFE library referenced by the SYSEXEC DD statements in the GDDRPROC JCL procedure.
3. Rename the existing exit module(s) in *hlq*.GDDR*vrm*.LINKLIB to allow the system to use your customized exit modules in your user partitioned dataset. This will allow you to rename the LINKLIB member back to its original name should you decide to no longer use the exit override.

### Sample procedure to use CA-OPS/MVS compiled REXX user exits

1. Code a TSO REXX module named GDDRUX*nn*.
2. In that module perform a CALL OI *modname*, where *modname* is the name of your actual user exit code in CA-OPS/MVS compiled REXX.
3. Modify your GDDRPROC JCL, adding the required OPSEXEC / OPSCOMP DD cards as well as your CA-OPS linklib in STEPLIB.

## GDDRUX01

This exit is called from planned or unplanned scripts at a point appropriate for starting production mainframe workloads. The exit must be named GDDRUX01.

### Parameters

1. Mode:
   - SYNCH — Caller will wait for result
   - ASYNC — Caller will not wait for result
2. System ID on which to start workload

3. Number of this system in the list of systems for which user exit 1 will be called
4. Number of systems in the list of systems for which user exit 1 will be called

### Return code

If zero, the exit will not be called on a script rerun. If non-zero, the exit will be called on a script rerun.

### Example

An example of how user exit 1 could be programmed is provided in the GDDR SAMPLIB distribution library member GDDRUX01.

## GDDRUX02

This exit is called from planned or unplanned scripts at a point appropriate for stopping production mainframe workloads. The exit must be named GDDRUX02.

### Parameters

1. Mode:
   - SYNCH — Caller will wait for result
   - ASYNC — Caller will not wait for result
2. System ID on which to stop workload
3. Number of this system in the list of systems for which user exit 2 will be called
4. Number of systems in the list of systems for which user exit 2 will be called

### Return code

If zero, the exit will not be called on a script rerun. If non-zero, the exit will be called on a script rerun.

### Example

An example of how user exit 2 could be programmed is provided in the GDDR SAMPLIB distribution library member GDDRUX02.

## GDDRUX03

This exit is called from planned or unplanned scripts at a point appropriate for starting production open systems workloads. The exit must be named GDDRUX03.

### Parameters

1. Mode:
   - SYNCH — Caller will wait for result
   - ASYNC — Caller will not wait for result
2. Source (site moving from)
3. Target (site moving to)
4. Context (reason for call - values are swap, rdr, tdc3, ldr)

### Return code

If zero, the exit will not be called on a script rerun. If non-zero, the exit will be called on a script rerun.

### Example

An example of how user exit 3 could be programmed is provided in the GDDR SAMPLIB distribution library member GDDRUX03.

## GDDRUX04

This exit is called from planned or unplanned scripts at a point appropriate for stopping production open systems workloads. The exit must be named GDDRUX04.

### Parameters

1. Mode:
   - SYNCH — Caller will wait for result
   - ASYNC — Caller will not wait for result
2. Source (site moving from)
3. Target (site moving to)
4. Context (reason for call–values are swap, rdr, tdc3, ldr)

### Return code

If zero, the exit will not be called on a script rerun. If non-zero, the exit will be called on a script rerun.

### Example

An example of how user exit 4 could be programmed is provided in the GDDR SAMPLIB distribution library member GDDRUX04.

## Built-in routines available to exits

Exit routines may save and retain values across separate invocations using built-in routines as follows:

- GDDRUXSV
- GDDRUXGV
- GDDRUXDV

### GDDRUXSV

GDDRUXSV allows you to save a value in a specified durable variable, creating the variable if necessary.

Invocation format is:

call gddruxsv *variable-name variable-value*

The return code is found in REXX variable 'result'. Any return code other than 0 indicates an error.

## GDDRUXGV

GDDRUXGV allows you to retrieve a value previously saved by GDDRUXSV.

Invocation format is:

call gddruxgv *variable-name*

The value is returned in REXX variable ‘result’. If no value is available, the ‘result’ variable is dropped and consequently becomes ‘RESULT’ when evaluated.

## GDDRUXDV

GDDRUXDV allows you to delete a durable variable previously created by GDDRUXSV.

Invocation format is:

call gddruxsv *variable-name variable-value*

The return code is found in REXX variable ‘result’. Any return code other than 0 indicates an error.

# CHAPTER 3
# Installing GDDR

This chapter includes the following topics:


- Hardware and software requirements ..... 96
- Licensing requirements ..... 99
- Required installation information ..... 100
- GDDR installation sequence ..... 101
- Steps to install GDDR ..... 102
- Post-installation tasks ..... 108

# Hardware and software requirements

Before installing GDDR, review the hardware and software requirements.

## Mainframe environment requirements

The basic infrastructure must support SRDF/SQAR with AutoSwap. In addition to this, GDDR has the following infrastructure requirements:

- There must be network connectivity between the following:
  - All C-systems

    **Note:** "C-systems" on page 46 explains C-systems.

  - C-systems and managed systems (if you configure GDDR to manage production systems)

    **Note:** "Managed systems" on page 47 explains managed systems and lists their types.

- For DLm integration using DLMDR, there must be network connectivity between the following:
  - The C-systems and the DLMDR instances providing management access to the DLm systems
  - All DLMDR instances
- If you configure GDDR for BCPii support, an HMC (Hardware Management Console) must be available to the systems defined to GDDR as *B-systems,* as described in "B-systems" on page 83 (access to these HMCs can be protected by means of a private VLAN).

## Mainframe hardware and software requirements

Table 6 lists minimum mainframe hardware and software requirements for a GDDR C-system.

**Table 6** Mainframe hardware and software requirements

| Item | Requirements |
|---|---|
| Processor hardware configuration | Any system that supports current IBM mainframe operating systems |
| Software | z/OS 2.2.x or later<br>JES2 or JES3 environments<br>**Note:** GDDR does not run VARY commands for tape devices in a JES3 environment.<br>BCPii (IBM Base Control Program internal interface)<br>BCPii is supported if the C-systems are using z/OS 1.10 or a later release. In addition, the CPC must be a z9 or higher (BC or EC). "BCPii support" on page 82 provides additional information.<br>The z/OS level of the managed systems is not a consideration for the use of BCPii for HMC operations. BCPii operations are conducted on the CPC named in GDDRMAIN control statements. That CPC may or may not host a C-system, but it must host some system running GDDRMAIN (C-system or production system).<br>The MCL levels that must be met are explained in the BCPii chapter of the MVS Programming Callable Services for High Level Languages document (SA22-7613). |
| Logical processors | Recommended: 2<br>Required: 1 |
| MSU | 15 on an IBM EC12 2827-417 (or equivalent, recommended) |
| Storage | 1 GB |
| Logical paths to own local DASD devices | 4 |
| Logical paths to managed DASD devices | 4<br>It is recommended to have separate channels for GDDR-managed storage gatekeepers and production gatekeeper functions. |

## Dell EMC software requirements

Table 7 lists Dell EMC software requirements for GDDR.

**Table 7** Dell EMC software requirements

| Item | Requirements |
|---|---|
| Mainframe Enablers | 8.3[a] |
| Disk Library for mainframe | DLMDR 4.5.3 |

a. Install Mainframe Enablers maintenance prior to GDDR SMP/E installation.

**Note:** GDDR does not support Dynamic Volume Expansion (DVE).

The minimum software prerequisites needed to run GDDR are as follows:

- ResourcePak Base with SRDF/A Multi-Session Consistency (MSC)
- SRDF Host Component
- Consistency Groups
- AutoSwap

Installation procedures for the Dell EMC software products are provided in the *Mainframe Enablers Installation and Customization Guide*.

### Configuration requirements

See the *SRDF Host Component for z/OS Product Guide* for additional information on configuring an SRDF/SQAR environment.

GDDR-specific configuration requirements include:

- CAX protection must be added to the SRDF/SQAR-defined consistency groups.

  **Note:** The *Consistency Groups for z/OS Product Guide* describe CAX.

- LOSTOWNERPOLICY ONSWAP=OPERATOR must be specified.

  **Note:** The *Consistency Groups for z/OS Product Guide* describes the LOSTOWNERPOLICY ONSWAP option.

- There must be one or more gatekeeper devices for each MSC-controlled SRDF group. These gatekeeper devices must be:
  - *Not* SRDF/S-protected
  - *Not* ConGroup and CAX-protected
  - *Not* SRDF/A-protected
  - If CKD, in OS configuration as OFFLINE at IPL

#### Restrictions

- SRDF/SQAR configurations do not support “external” devices on the SRDF/S leg.
- SRDF Host Component actions which change the devices defined to SQAR MSC groups require the MSC tasks to be down at the time of the change.

## DASD requirements

GDDR has the DASD requirements listed in Table 8.

**Table 8** DASD requirements

| Item | Requirements |
|---|---|
| DASD hardware configuration | Any supported PowerMax/VMAX DASD model at the operating environment level specified in the *GDDR Release Notes* |

GDDR supports and can manage the following combinations of DASD:

- Single or multiple storage systems per site configured with any of the following:
  - All CKD devices
  - All FBA and FBA-META devices
  - Any combination of CKD, FBA and FBA-META devices

You can have multiple storage systems at one site and a single storage subsystem at another site.

At any site, you can have any mix of supported PowerMax and VMAX models.

Management and monitoring of both CKD and FBA/FBA-META devices is performed from the z/OS platform where the GDDR application resides. From the GDDR point of view, CKD and FBA/FBA-META PowerMax/VMAX devices are the same; that is, each is treated no differently than the other. They are all command targets of SRDF Host Component configuration commands using local, remote or GNS syntax.

GDDR requires that if even only one device in an SRDF group is defined to GDDR, then *all* devices in that group must be defined to GDDR. Most GDDR actions are directed at the SRDF group level (although in some cases, GDDR will act on device ranges if that is appropriate).

GDDR has no limitations on the number of PowerMax/VMAX systems or devices that can be managed. Any limitations are subject to restrictions in Dell EMC hardware and software.

# Licensing requirements

Ensure you have the "GDDR" Licensed Feature Code (LFC) available. You specify the LFC as part of GDDR post-installation tasks, according to the procedure described in "Install GDDR Licensed Feature Code" on page 144.

# Required installation information

Before beginning the GDDR installation, identify or decide on the following items:

- CLIST library and edit macro

  Determine a name for the edit macro created by the installation dialog. You also need to determine the name of a CLIST library where you can store the edit macro.

- Product dataset name prefix

  Choose the dataset prefix you will use to install GDDR. Names for the product datasets consist of a final qualifier, such as LINKLIB, and a dataset prefix. For example, if you choose a dataset prefix of EMC.GDDR*vrm*, the LINKLIB dataset will be named EMC.GDDR*vrm*.LINKLIB.

  Ensure that you have RACF ALTER authority (or the equivalent from another security manager) for the datasets created with this dataset prefix.

  ---

  **Note:** Throughout this guide, datasets created using this dataset prefix are referred to as if they had been created with the suggested value. The actual value for your installation may be different.

  ---

- Mainframe Enablers dataset name prefix

  Specify the dataset name prefix you used when you installed Mainframe Enablers. It is recommended to use EMC.*fmid* if it agrees with your site standards.

  ---

  **Note:** Further in this document, the Mainframe Enablers dataset name prefix is referred to as MFE*vrm*.

  ---

- SMP/E dataset name prefix

  Choose the name prefix for the SMP/E datasets into which you installed GDDR. If you have installed another Dell EMC product using SMP/E, you should install GDDR into the same CSI. If you are installing a Dell EMC SMP/E maintained product for the first time, it is recommended to use "EMC.SMPE."

- SMP/E datasets volser

  Choose the disk volume onto which you will install the distribution libraries required by SMP/E. This may be the same volume you use for the product libraries. However, many customer sites prefer to keep SMP/E-related datasets on separate volumes from product libraries. An amount of space similar to that needed for the product libraries is required.

- Install-to-disk volser

  Determine the disk volume onto which you will install the target (that is, runtime) datasets. The space required is nominal. Dell EMC suggests that you use EMC.*fmid* if it agrees with your site standards.

- Disk unit name

  Decide on a disk unit name for the above volumes. For many users, "SYSDA" will suffice. However, use whatever generic or esoteric name your local standards require.

# GDDR installation sequence

GDDR installation includes the following activities:

- Install and customize Mainframe Enablers

  Complete the installation and customization procedures provided in the *Mainframe Enablers Installation and Customization Guide.*

- Install and customize GDDR

  Follow the procedures described in "Steps to install GDDR" on page 102 and "Integrating GDDR" on page 109.

# Steps to install GDDR

The GDDR installation kit is provided as an electronic download from Dell EMC Online Support.

The GDDR installation kit consists of a PDS containing TSO TRANSMIT images of files needed to perform an SMP/E indirect-library installation.

## Overview

To install GDDR on a C-system, take the following steps:

1. Load the TSO TRANSMIT file, GDDR*vrm*.XMITLIB, to the mainframe disk, as described in "Load GDDRvrm.XMITFILE to disk" on page 102.
2. Run GDDR*vrm*.XMITLIB(#EXTRACT) to extract *ds-prefix*.RIMLIB and the SMP/E indirect libraries, as described in "Run GDDRvrm.XMITLIB(#EXTRACT)" on page 104.
3. Customize the RIMLIB JCL as described in "Customize RIMLIB JCL" on page 104.
4. Run the installation jobs as described in "Run installation jobs" on page 107.
5. Perform cleanup as described in "Perform cleanup" on page 108.
6. Apply maintenance updates as described in "Apply maintenance updates" on page 108.

## Load GDDR*vrm*.XMITFILE to disk

1. Complete the following steps:
    a. Log in to a privileged account on an open systems host (root on UNIX or administrator on Windows).
    b. Select a working directory on the open systems host for the installation.
    c. Log in to www.dell.com/support.
    d. Search for **Geographically Dispersed Disaster Restart (GDDR)** in the 'Enter a Service Tag, Serial Number, Service Request, Model, or Keyword' field.

    **Result**: A support page for GDDR is displayed.

    **Note:** If you are not able to access this location, you may not have registered the software or registered it incorrectly. Follow the prompts to register the software, correct the registration, or contact Dell EMC in the event of a problem.

    e. Click **Drivers & Downloads** on the GDDR support page.

    **Result**: The Downloads for Geographically Dispersed Disaster Restart (GDDR) page is displayed.

    f. Click the required product version in the Version field to filter on the version.
    g. Click the **Download** button for the GDDR electronic distribution kit and download it into the working directory that you selected in step b.

2. If your current host is a Windows system, unzip the file in the working directory. If your current host is a UNIX system, unzip and untar the file into the working directory.
3. Locate GDDR*vrm*.XMITFILE.

   This file is in TSO TRANSMIT format and contains a flattened copy of GDDR*vrm*.XMITLIB, a PDS that holds other TRANSMIT images, the JCL to extract them, and necessary SMP/E installation files.
4. On the target mainframe, allocate a file to which you can FTP GDDR*vrm*.XMITFILE.

   Use the dataset name prefix you intend to use for GDDR installation. The final qualifier must be XMITFILE. For example, if you intend to install GDDR with a dataset name prefix of EMC.GDDR*vrm,* name the file EMC.GDDR*vrm*.XMITFILE.

   Allocate the dataset with the following characteristics:

   ```
   LRECL=80
   BLKSIZE=3120
   DSORG=PS
   SPACE=(CYL,(44,2))
   ```

   **Note:** The SPACE parameter assumes that you are allocating the dataset on a 3390 device.
5. FTP the file to the mainframe in binary format.

   Your FTP session may look something like the following:

   ```
   ftp hostname
   (username and password prompts)
   cd ..
   25 "'' is working directory name prefix
   binary
   200 Representation type is image
   put GDDRvrm.XMITFILE EMC.GDDRvrm.XMITFILE
   ```
6. Use **TSO RECEIVE** to receive the file into a PDS.

   The PDS is created by the RECEIVE command and does not have to be preallocated. However, you *must* specify a dataset name using the DA[taset] parameter or the file will be allocated using your TSO prefix (usually your logon ID). The dataset name specified must have the final qualifier of XMITLIB.

   For example:

   ```
   receive indataset('EMC.GDDRvrm.XMITFILE')
   INMR901I Dataset EMC.GDDRvrm.XMITLIB from userid on nodename
   INMR906A Enter restore parameters or 'DELETE' or 'END' +
   da('EMC.GDDRvrm.XMITLIB')
   ```

   If you did not specify "DA(...)" as above, the dataset would be allocated as *userid.*XMITLIB.

## Run GDDR*vrm*.XMITLIB(#EXTRACT)

Now run GDDR*vrm*.XMITLIB(#EXTRACT) to extract *ds-prefix*.RIMLIB and the SMP/E indirect libraries. Take the following steps:

1. Edit the #EXTRACT member of the newly RECEIVED library.

   You can edit the #EXTRACT job by running the SETUP REXX program you can find in the XMITLIB dataset. The SETUP REXX program prompts you for all of the information needed to edit the job.

   If you wish to edit the job manually, make the following changes:

   - Change the job card to one that conforms to your standards.
   - Globally change *ds-prefix* to the dataset prefix of this library (which will be the dataset prefix for GDDR libraries).
   - Globally change DVOL to the disk volser onto which you want to place the extracted libraries.
   - Globally change DISK-UNIT to an esoteric unit name such as "SYSDA" that is appropriate for your site.

2. Submit #EXTRACT. Step completion codes should be 0, except for the DELETE step, which will have a step completion code of 8 unless the job is a rerun.

## Customize RIMLIB JCL

The RIMLIB library (*ds-prefix*.RIMLIB) is a PDS containing JCL to install GDDR. After you extract the RIMLIB PDS, you find that RIMLIB has the contents shown in Table 9.

**Table 9** RIMLIB library contents

| File | Contents |
|---|---|
| #01ALLOC | Allocates target and distribution libraries |
| #02DDDEF | Adds or replaces GDDR library DDDEFS to SMP/E CSI |
| #03RECEV | SMP/E RECEIVE function into global zone |
| #04APPLY | SMP/E APPLY function into target zone |
| #05ACCPT | SMP/E ACCEPT GDDR sysmods into distribution zone |
| #06CLEAN | Deletes indirect libraries and DDDEFs used for them |
| #99MAINT | SMP/E RECEIVE and APPLY service |
| GDRJCL | REXX to customize the install process |
| GDRWIN1 | ISPF panel used in REXX install process |
| SETUP | REXX to simplify the customization process |

Complete the following steps to customize the installation JCL using the automated dialog:

1. Edit the RIMLIB library (*ds-prefix*.RIMLIB).

2. Locate the member named SETUP on the member selection list (shown in Figure 24) and type **EX** in the selection column next to it. Press **Enter**.

```
Menu  Functions  Confirm  Utilities  Help
------------------------------------------------------------------------------
EDIT              EMC.GDDRvrm.RIMLIB                      Row 00001 of 00013
Command ===>                                                Scroll ===> CSR
            Name      Prompt       Size   Created          Changed          ID
__________ #01ALLOC                  45  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ #02DDDEF                  51  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ #03RECEV                  22  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ #04APPLY                  22  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ #05ACCPT                  22  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ #06CLEAN                  53  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ #99MAINT                  27  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ GDRJCL                   206  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
__________ GDRWIN1                   51  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
ex________ SETUP                     13  yyyy/mm/dd yyyy/mm/dd hh:mm:ss idstring
           **End**
```

**Figure 24** Member selection list

**Result**: The Dell EMC JCL Customization Utility panel is displayed, shown in Figure 25.

```
+-------------------- Dell EMC JCL Customization Utility -------------------+
| COMMAND ==> _____________________________________________________________ |
|                                                                           |
| Type EXEC on the command line and press ENTER to proceed, or PF3 to exit. |
|                                                                           |
| CLIST library         ==> 'hlq.GDDRvrm.RIMLIB'                            |
| Edit macro name       ==> GDR                                             |
| Product dsname prefix ==> hlq.GDDRvrm                                     |
| Mainframe Enablers                                                        |
|   dsname prefix       ==> hlq.MFEvrm                                      |
| SMP/E dsname prefix   ==> EMC.SMPE                                        |
| SMP/E datasets volser ==> ______                                          |
| Install-to disk volser==> ______       Disk unit name ==> SYSDA           |
|                                                                           |
| Enter your job card below ('%MEMBER%' will be replaced by member name):   |
| => //%MEMBER% JOB MSGCLASS=A,CLASS=A,MSGLEVEL=(1,1)                       |
+---------------------------------------------------------------------------+
```

**Figure 25** Dell EMC JCL Customization Utility

3. Enter or change the following information on the Dell EMC JCL Customization Utility panel (shown in Figure 25) to customize your installation:

   a. The CLIST library field is set by default to the name of the RIMLIB library. This field should contain the name of a library in which you want the edit macro created by this dialog to be stored.

      The default value is fine for most users and need not be changed.

   b. In the Edit macro name field, either:

      – Accept the default name displayed.

      or

      – If necessary, change the name of the edit macro.

      **Note:** Normally, you should not have to change the name.

**Result**: The edit macro is created in the CLIST library from the data entered on the Dell EMC JCL Customization Utility panel and applied to all members of RIMLIB that start with a # character.

c. In the Product dsname prefix field, enter the dataset name prefix you want to use for the target datasets. Dell EMC suggests *hlq*.GDDR*vrm*.

d. In the Mainframe Enablers dsname prefix field, enter the dataset name prefix you want to use for Mainframe Enablers. Dell EMC suggests *hlq*.MFE*vrm*.

e. In the SMP/E dsname prefix field, enter the dataset name prefix of the SMP/E datasets into which you installed GDDR.

   For example, if you named the SMPSCDS dataset EMC.SMPE.SMPSCDS, enter EMC.SMPE.

f. In the SMP/E datasets volser field, enter the six-character volume serial number of the disk volume on which you want to allocate the SMP/E distribution libraries for GDDR.

   This volume may be the same as the volume you specify in the Install-to disk volser field (next step), or you may elect to keep these datasets on a separate volume.

g. In the Install-to disk volser field, enter the six-character volume serial number of the disk volume to which you want to install the GDDR libraries.

h. In the Disk unit name field, you can specify an esoteric disk name that is appropriate to your site. SYSDA is the default, but you can overtype it with another esoteric disk name.

i. Enter a site-appropriate job card.

   The job card is initially set to a value which may be suitable to many users. The first seven characters of the job name is set to your TSO user ID, plus "X."

   You can set the job name to %MEMBER%. This causes the edit macro to set each job name equal to the JCL member name (that is, #01ALLOC, #02DDDEF, and so forth).

   Do not use any parameter that contains an ampersand (&), such as NOTIFY=&SYSUID. An ampersand in the job card can cause edit macro errors.

Figure 26 shows an example of a completed Dell EMC JCL Customization Utility panel as the user is about to press **Enter** and complete the dialog.

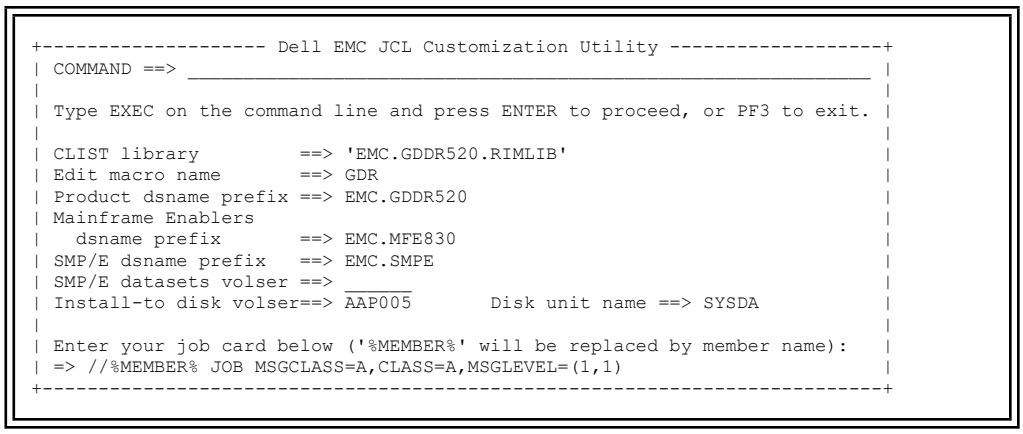



**Figure 26** DELL EMC JCL Customization Utility completed panel

4. When you are satisfied with your entries, type **exec** on the command line and press **Enter**.

   **Result**: If the dialog completes successfully, you see something similar to the following:

```
BUILDING AN EDIT MACRO(GD) IN 'EMC.GDDRvrm.RIMLIB'
PROCESSING MEMBER:  #01ALLOC
PROCESSING MEMBER:  #02DDDEF
PROCESSING MEMBER:  #03RECEV
PROCESSING MEMBER:  #04APPLY
PROCESSING MEMBER:  #05ACCPT
PROCESSING MEMBER:  #06CLEAN
PROCESSING MEMBER:  #99MAINT
***
```

## Run installation jobs

Carefully examine each job before you submit it to ensure that it was customized the way you intended.

Submit the customized jobs in the following order, ensuring that each job completes successfully before submitting the next one:

1. #01ALLOC
2. #02DDDEF
3. #03RECEV
4. #04APPLY

You should expect completion codes of 0 (zero) for all jobs except for #02DDDEF, where 04 is acceptable if this is a new installation rather than an upgrade.

If your test results are positive, run #05ACCPT to update the distribution libraries and zone. The #05ACCPT job completes with an RC=04. This is normal for the SMP/E ACCEPT process. You can ignore it.

SMP/E installation is now complete.

## Perform cleanup

After you are satisfied that GDDR is correctly installed and functioning properly, run the #06CLEAN job to delete datasets and DDDEFs used during the installation process that are no longer needed.

## Apply maintenance updates

If you have received maintenance cover letters from Dell EMC or have instructions to apply maintenance from Dell EMC support personnel, use the supplied job #99MAINT. This job receives and applies APARs and PTFs. This job may require further customization before you run it, depending on the nature of the maintenance.

**Note:** Do not attempt to apply maintenance until the GDDR ACCEPT job has completed successfully and then only if instructed to do so by Dell EMC support personnel.

# Post-installation tasks

Having completed the SMP/E installation steps, complete the tasks described in Chapter 4, "Integrating GDDR."

# CHAPTER 4
# Integrating GDDR

This chapter includes the following topics:


- Introduction ..... 110
- Update system parameter files ..... 110
- Configure BCPii ..... 114
- Create SRDF parameter members (SITxxxxx) ..... 119
- Set SCF initialization parameters ..... 121
- Authorize ConGroup to use Trip API ..... 122
- Configure ConGroup ..... 123
- Set up ConGroup automated startup ..... 124
- Configure MSC ..... 126
- Set up GDDR security ..... 130
- Define GDDR datasets ..... 139
- Install GDDR started procedures ..... 141
- Install GDDR Licensed Feature Code ..... 144
- Customize GDDRMAIN parameters ..... 145
- Customize PROCLIB member GDDRPROC ..... 150
- Customize GDDR ISPF interface invocation REXX exec ..... 151
- Update GDDR ISPF profile ..... 151
- Configure GDDR ..... 152
- Configure optional features ..... 155
- Modify GDDR user exits ..... 161

# Introduction

When you have completed the SMP/E installation steps described in "Installing GDDR" on page 95, complete the tasks listed in this chapter before using GDDR.

IMPORTANT

Unless noted otherwise, these changes are made on the C-systems only.

# Update system parameter files

Complete the following tasks to update system parameter files:

- Customize SYS1.PARMLIB(IKJTSOxx)
- Customize TSO logon
- APF-authorize LINKLIB
- Customize LINKLIB and REXX parameter files

## Customize SYS1.PARMLIB(IKJTSO*xx*)

1. Confirm that the following entries exist in AUTHCMD, AUTHPGM, and AUTHTSF lists. Add any missing entries to the IKJTSO*xx* member of SYS1.PARMLIB of each C-system.

To **AUTHCMD** add entries:

```
SCFRDFME              /* Dell EMC ME utility                                  */  +
SCFRDFM6              /* Dell EMC M6 utility                                  */  +
EHCMSCM9              /* Dell EMC M9 utility                                  */  +
SCFRDFM9              /* Dell EMC M9 utility                                  */  +
EHCMSCME              /* Dell EMC ME utility                                  */  +
EHCMSCM6              /* Dell EMC M6 utility                                  */  +
```

To **AUTHPGM** add entries:

```
GDDFLIST              /* SDDF list utility                                    */  +
GDDRDAP1              /* GDDR                                                 */  +
GDDRDAP3              /* GDDR                                                 */  +
GDDRXCMD              /* GDDR                                                 */  +
GDDRSTAT              /* GDDR check for presence of an active task            */  +
GDDRSSVI              /* Initialize GDDR command queue                        */  +
GDDRQFCN              /* Manipulate GDDR command queue                        */  +
GDDRCGTP              /* GDDR - ConGroup communication                        */  +
GDDRSTOK              /* GDDR                                                 */  +
GDDRQRY5              /* GDDR                                                 */  +
GDDRAUTH              /* GDDR authorization check                             */  +
GDDCAUTH              /* GDDR authorization check                             */  +
GDDRMCMD              /* GDDR command processor                               */  +
GDDGATEK              /* GDDR gatekeeper validation utility                   */  +
GDDMSCFX              /* GDDR configuration validation utility                */  +
SCFRDFME              /* Dell EMC ME utility                                  */  +
SCFRDFM6              /* Dell EMC M6 utility                                  */  +
EHCMSCM9              /* Dell EMC M9 utility                                  */  +
EMCTF                 /* Dell EMC TimeFinder Mirror                           */  +
EHCMSCM6              /* Dell EMC M6 MSC cleanup utility                      */  +
EHCMSCME              /* Dell EMC ME MSC cleanup utility                      */  +
SCFRDFM9              /* Dell EMC M9 utility                                  */  +
ECGUTIL               /* Dell EMC ConGroup cleanup utility                    */  +
EMCSNAP               /* Dell EMC TimeFinder Snap                             */  +
```

To **AUTHTSF** add entries:

```
GDDBCPC2              /* BCPII AUTH CONNECT                                   */ +
GDDBCPC3              /* BCPII AUTH COMMAND                                   */ +
GDDBCPD2              /* BCPII AUTH DISCONNECT                                */ +
GDDBCPL2              /* BCPII AUTH LIST                                      */ +
GDDBCPQ2              /* BCPII AUTH QUERY                                     */ +
GDDBCPS2              /* BCPII AUTH SET                                       */ +
GDDRXMPS              /* GDDR                                                 */ +
GDDRAWTZ              /* GDDR                                                 */ +
GDDRINF2              /* GDDR                                                 */ +
GDDRMCS1              /* GDDR                                                 */ +
GDDRQDE2              /* GDDR                                                 */ +
GDDRQDE3              /* GDDR                                                 */ +
GDDRQDE4              /* GDDR                                                 */ +
GDDRQDE5              /* GDDR                                                 */ +
GDDRQDE6              /* GDDR                                                 */ +
GDDRQDE7              /* GDDR                                                 */ +
GDDRQDE8              /* GDDR                                                 */ +
GDDRQDE9              /* GDDR                                                 */ +
GDDRQDEA              /* GDDR                                                 */ +
GDDRQDEB              /* GDDR                                                 */ +
GDDRQDEC              /* GDDR                                                 */ +
GDDRQDED              /* GDDR                                                 */ +
GDDRQDEE              /* GDDR                                                 */ +
GDDRQDEF              /* GDDR                                                 */ +
GDDRQDEG              /* GDDR                                                 */
```

2. Activate the changes using an IPL or dynamically with the SET IKJTSO=*xx* or TSO PARMLIB UPDATE(*xx*) command.

## Customize TSO logon

1. You may need to increase the region size of TSO logon procedures that use the GDDR ISPF interface and batch jobs that run GDDR scripts. It is recommended to allocate a TSO logon procedure region of 2,100,000, as a starting point.
2. Ensure that TSO logon procedures of all TSO users who wish to run the GDDR ISPF interface contain a SYSEXEC DD which points to *hlq*.GDDR*vrm*.RCXFE. This library is allocated during GDDR installation.

## APF-authorize LINKLIB

APF-authorize the *hlq*.MFE*vrm*.LINKLIB and *hlq*.GDDR*vrm*.LINKLIB libraries.

## Customize LINKLIB and REXX parameter files

**Note:** You can skip this step if your site prefers to reference *hlq*.GDDR*vrm*.LINKLIB and *hlq*.MFE*vrm*.LINKLIB by using STEPLIB DD statements.

GDDR uses a REXX function package named GDDFUSER with an alias of IRXFUSER. IRXFUSER is a placeholder in SYS1.LINKLIB, providing flexibility for customization of REXX function packages. You have several choices for installation, depending on whether you have applications using previously customized REXX function packages on C-systems and managed systems. These instructions apply to each C-system and each managed system on which GDDRMAIN is to be installed.

- If no applications use previously customized REXX function packages, or all such applications use STEPLIB DD statements to reference a customized REXX function package module, proceed as described in "Customize LINKLIST" on page 112.
- If customized REXX function packages named IRXFUSER, accessed via the LINKLST, exist on one or more managed systems or C-systems, perform the steps listed in "Customize LINKLIST" on page 112 and "Customize REXX parameter files" on page 113.

### Customize LINKLIST

1. Delete the placeholder IRXFUSER module from SYS1.LINKLIB.
2. Add *hlq*.MFE*vrm*.LINKLIB and *hlq*.GDDR*vrm*.LINKLIB to the LINKLIST using one of the following methods:
   - Add the following LNKLST entries in a PROG*xx* member:

```
LNKLST ADD NAME(LNKLST) DSN(hlq.MFEvrm.LINKLIB)
LNKLST ADD NAME(LNKLST) DSN(hlq.GDDRvrm.LINKLIB)
```

or

- Add the following entries in a LNKLST*xx* member:

```
hlq.MFEvrm.LINKLIB(vvvvvv)
hlq.GDDRvrm.LINKLIB(vvvvvv)
```

In these entries, *vrm* is the current GDDR version, release, modification identifier and *vvvvvv* is the volser where the *hlq*.GDDR*vrm*.LINKLIB dataset resides. The volser specification is only required if the dataset is not cataloged in the master catalog.

3. Replace *hlq*.GDDR*vrm*.LINKLIB with the dsname of the GDDR LINKLIB SMP/E target library allocated and filled during GDDR installation. Activate this change using one of the following methods:
   - IPL
   - Issue the SET PROG=*xx* command.
   - Issue the SETPROG LINKLIST,ADD command.

## Customize REXX parameter files

This procedure is used when there is an application using previously customized REXX function packages not accessed by a STEPLIB DD statement. The application has already taken advantage of the "placeholder" aspect of IRXFUSER in SYS1.LINKLIB, so additional steps are required to ensure the GDDR uses the proper function package named GDDFUSER, without the alias IRXFUSER.

1. Add GDDFUSER to the three REXX parameter files used in establishing the REXX processing environment. Before you begin, create a backup copy of the files.

   The REXX parameter files to be edited are:
   - IRXPARMS for REXX under MVS
   - IRSTSPRM for REXX under TSO
   - IRXISPRM for REXX under ISPF
2. Place the newly assembled versions of IRXPARMS, IRXISPRM, and IRXTSPRM into SYS1.LPALIB overlaying the default members.

**Note:** For more information, see the *TSO/E REXX Reference SA22-7790*, chapter Language Processing Environments, subtopic 'Changing the default values for initializing an environment'.

# Configure BCPii

Configure BCPii on all systems where GDDRMAIN is expected to support BCPii actions.

**Note:** "BCPii support" on page 82 discusses BCPii.

## BCPii requirements

The following requirements must be met to use BCPii:

- The z/OS operating system be at level 1.11 or higher. (BCPii can run under z/OS 1.10 with the installed BCPii deliverable; however, not all functionality is available at the 1.10 level).
- The CPC must be a z9 or higher (BC or EC). There are certain MCL levels that must be met which are explained in the BCPii chapter of the *MVS Programming Callable Services for High Level Languages* document (SA22-7613).
- Specific actions must be performed on the SE/HMC for each CPC in order to enable BCPii to run on that CPC. This is referred to as LIC Enabled status for the HMC in reference to the SE/CPC.

## Set up security for BCPii

BCPii security is controlled both at the SE for each CPC holding z/OS images that will participate in BCPii communications and also within the security product running on each of the participating z/OS systems.

**Note:** Examples provided in this guide illustrate configuration using only the IBM Security Server (RACF) product.

At the SE side, follow the instructions in the *MVS Programming Callable Services for High Level Languages* document (SA22-7613) in the Callable Service chapter, section "Base Control Program Internal Interface" under the heading "BCPii Setup and Installation". For each z/OS Image on a CPC that will participate, cross-partition authority must be enabled through the SE. Additionally, an SNMP Community Name for each CPC must be established. Due to restrictions with the security products on z/OS, the Community Name cannot contain any lowercase characters. See the documentation for specific steps to follow to establish the connection between authorized applications and the SE.

At the z/OS security product side, the section entitled "Setting up Authority to Use BCPii" of the *MVS Programming Callable Services for High Level Languages* document (SA22-7613) outlines the steps required to update your security product to allow the connection to BCPii.

The HWIBCPII address space under z/OS is the pathway between the authorized applications and the SE when communicating via BCPii. If everything is configured properly, this address space comes up automatically during IPL and remains available for the life of the image. There are instructions for shutting this address space down and restarting if necessary.

IMPORTANT

It is important that all of the setup and requirements outlined in the BCPii chapter in *MVS Programming Callable Services for High Level Languages* are understood. This ensures the security configuration meets your specific needs in a safe and straightforward fashion.

## BCPii facility classes

Pertinent BCPii facility classes related to GDDR are as follows:

- HWI.APPLNAME.HWISERV

  HWI.APPLNAME.HWISERV in the Facility Resource Class controls which applications can use BCPii services. A minimum of READ access is required and BCPii requires that the Facility Resource Class be RACLISTED. See the *MVS Programming Callable Services for High Level Languages* document (SA22-7613) for an example of creating, using and refreshing this profile.

- HWI.TARGET.*netid*.*nau*

  HWI.TARGET.*netid*.*nau* in the Facility Resource Class controls access to a specified CPC using the full format name comprised of network ID *netid* followed by a period followed by the CPC name *nau* (network addressable unit). *netid* and *nau* can be up to 8 characters in length each, so that the *netid*.*nau* field can be from 3 to 17 characters total length.

  The APPLICATION DATA field for this facility class must match exactly the community name entered in the SE for the CPC during BCPii setup. This is the connection between the security product and access to the desired CPC. This facility class is also used to control access to any activation profiles stored on the CPC (image, load or reset).

- HWI.TARGET.*netid*.*nau*.*imagename*

  HWI.TARGET.*netid*.*nau*.*imagename* in the Facility Resource Class controls access to the specified image on the specified CPC. *imagename* can be up to 8 characters in length.

RACF allows for the use of generic facility classes. This can facilitate setup for the site. Keep in mind that RACF determines which profile to use when more than one applies by using the more specific class. For example, you define HWI.TARGET.IBM390PS.C.* as a generic Facility class to cover access to all LPARs on the C CPC. You also define HWI.TARGET.IBM390PS.* as a generic Facility class to cover access to all LPARs regardless of CEC. If you are working with an LPAR on the C CPC, RACF will use HWI.TARGET.IBM390PS.C.*; otherwise, HWI.TARGET.IBM390PS.* will be used.

## Customize SYS1.PARMLIB for BCPii

Authority to run the GDDR BCPii API functions requires changes to the AUTHTSF section of IKJTSO00 in SYS1.PARMLIB. Add the following entries on all GDDR B-systems (C-system or managed system) named on a CPC statement in the GDDRPARM file:

```
GDDBCPC2              /* BCPII AUTH CONNECT      */ +
GDDBCPC3              /* BCPII AUTH COMMAND      */ +
GDDBCPD2              /* BCPII AUTH DISCONNECT   */ +
GDDBCPE2              /* BCPII AUTH EVENT        */ +
GDDBCPL2              /* BCPII AUTH LIST         */ +
GDDBCPQ2              /* BCPII AUTH QUERY        */ +
GDDBCPS2              /* BCPII AUTH SET          */ +
```

## Specify CPC parameters

The general syntax of the CPC parameters is explained in “CPC” on page 261. There are two types of CPC parameters: NETID parameters and METHOD parameters.

### CPC NETID parameters

A CPC NETID parameter entry is required in the GDDRPARM file for each GDDR C-system specifying the LAN network ID associated with the HMC network that is affiliated to the SE within the GDDR C-system's CPC. The purpose is to identify the NETID portion of the C-system CPC names, as this cannot be discovered.

One of the tasks within the GDDR Parameter Wizard process involves setting up the GDDR C-systems with values including the CPC and LPAR names along with LOAD PARM data associated with the image.

GDDR requires the NETID in the C-systems’ GDDR Parameter Wizard setup as this fully qualified CPC name is stored in a global variable used throughout GDDR. This requires that each C-system have a self-defining NETID entry in a CPC NETID parameter in the GDDRPARM file. The format for the CPC NETID parameter is identical to that of the CPC HMC METHOD parameter, with the exception that in place of the HMC method at the end of the parameters, the literal “NETID” is substituted. As an example the CPC NETID record for the B-system at DC1 running on CPCA would be as follows:

```
CPC       CSYS1     GDDR DC1,IBM390PS.CPCA,NETID
```

IBM390PS in the above CPC parameter represents the NETID, and CPCA the NAU. The example network ID (NETID) IBM390PS used here may be in use at your facility; however, this value may not be IBM390PS. Check with your systems programmer.

### CPC METHOD parameters

At least one CPC METHOD parameter entry per managed CPC is required. A managed CPC is any CPC where either a C-system or a GDDR-managed system can run. Multiple entries for the same CPC are allowed, one per B-system in the configuration. The METHOD records:

- Inform GDDR about the method to use for HMC actions against the listed CPC. Currently, only the BCPii method is supported.
- Define B-systems which are expected to have SE/HMC network access to the listed CPC.

Routing of BCPii actions between GDDR B-systems is allowed and is explained in the following section.

## CPC parameter SE/HMC control requirements

A key point when discussing BCPii revolves around where BCPii is actually required. The system on which the BCPii activity is initiated is where the BCPii address space is required in order to communicate with the SE of the local CPC and, if necessary, the HMC for remote SE requests. The target systems do *not* need to be running the BCPii address space in order for the functions to work as BCPii is communicating with the SE (remote SE via the affiliated HMC if necessary), not the images running on those systems. As long as the GDDR B-system is running a new enough version of z/OS on a new enough CPC, it can manage z/OS images back to the earliest level of z/OS supported on the required CPC level (z9 and higher). BCPii is supported with GDDR if at least one of the CPCs on which a B-system is running meets the requirements and that system has network connectivity to the HMC/SE associated with all other CPCs.

During GDDRMAIN startup, the CPC entries within the GDDRPARM file are read and validated. Updates can be made to the CPC entries if changes are desired. Those changes are applied to the running GDDRMAIN by issuing a restart of the MISC task (F GDDRMAIN,RESTART MISC). As the GDDRPARM file is required to be in sync across the GDDRMAIN tasks running, the Broadcast command option (BC) can be used to restart the MISC task on all GDDR B-systems.

All GDDR HMC tasks are done as external work within the REXX worker (GDDRWORK tasks GDDWXH and/or GDDWXR). All information related to the HMC task including, if requested, tracing information, can be found in the SYSTSPRT sysout stream associated with the GDDRWORK address space on the appropriate B-system. Again, keep in mind this might not be the C-system on which the original task was initiated (either via scripting, batch work, or the ISPF interface). This is the system upon which the task is *executed*.

As an example, consider the following configuration:

CSYS1—C-system 1 running on CPC IBM390PS.CPCA at DC1

CSYS2—Master C-system 2 running on CPC IBM390PS.CPCB at DC2

CSYS3—C-system 3 running on CPC IBM390PS.CPCC at DC3

PSYS1—Managed system PSYS1 running on IBM390PS.CPCD at DC1

PSYS2—Managed system PSYS2 running on IBM390PS.CPCD at DC1

PSYS3—Managed system PSYS3 running on IBM390PS.CPCE at DC2

PSYS4—Managed system PSYS4 running on IBM390PS.CPCF at DC2

PSYS5—Managed system PSYS5 running on IBM390PS.CPCG at DC3

```
CPC      CSYS1    GDDR DC1,IBM390PS.CPCA,BCPII
CPC      CSYS1    GDDR DC1,IBM390PS.CPCD,BCPII
CPC      CSYS1    GDDR DC2,IBM390PS.CPCE,BCPII
*
CPC      CSYS2    GDDR DC2,IBM390PS.CPCB,BCPII
CPC      CSYS2    GDDR DC2,IBM390PS.CPCE,BCPII
CPC      CSYS2    GDDR DC2,IBM390PS.CPCF,BCPII
*
CPC      CSYS3    GDDR DC3,IBM390PS.CPCC,BCPII
CPC      CSYS3    GDDR DC3,IBM390PS.CPCG,BCPII
```

HMC functions originating on C-system CSYS1 targeting system PSYS1 at DC1 run on CSYS1.

- There is a CPC record connecting CSYS1 to CPC IBM390PS.CPCD.

HMC functions originating on C-system CSYS1 targeting system PSYS3 at DC2 run on CSYS1.

- There is a CPC record connecting CSYS1 to CPC IBM390PS.CPCE.

HMC functions originating on C-system CSYS1 targeting system PSYS4 at DC2 run on CSYS2.

- There is no CPC record connecting CSYS1 to CPC IBM390PS.CPCF so the work is run remotely at CSYS2.

HMC functions originating on z/OS system CSYS2 targeting system PSYS3 at DC2 run on CSYS2.

- There is a CPC record connecting CSYS2 to CPC IBM390PS.CPCE. The CPC parameter statements below are used for the description of the validation that occurs when GDDRMAIN is started.

```
CPC      CSYS1    GDDR DC1,IBM390PS.CPCA,BCPII
CPC      CSYS1    GDDR DC2,IBM390PS.CPCB,BCPII
CPC      CSYS1    GDDR DC3,IBM390PS.CPCC,BCPII
CPC      CSYS1    GDDR DC1,IBM390PS.CPCD,BCPII
CPC      CSYS1    GDDR DC2,IBM390PS.CPCE,BCPII
CPC      CSYS1    GDDR DC2,IBM390PS.CPCF,BCPII
CPC      CSYS1    GDDR DC3,IBM390PS.CPCG,BCPII
*
* NETID records for GDDI123B C system processing
*
CPC      CSYS1    GDDR DC1,IBM390PS.CPCA,NETID
CPC      CSYS2    GDDR DC2,IBM390PS.CPCB,NETID
CPC      CSYS3    GDDR DC3,IBM390PS.CPCC,NETID
```

The CPC entries for CSYS1, CSYS2, and CSYS3 which specify NETID are populated on the Define C-Systems panel by Auto-Discovery, using the GDDR Parameter Wizard. GDDRMAIN startup or refresh of the MISC task will check and ensure a valid NETID CPC record exists for each C-system specified in the GDDRPARM file. If not found, GDDR error message GDDM194E is displayed in the GDDRMAIN output to alert the user. All HMC functions against the managed systems PSYS1, PSYS2, and PSYS3 is routed to C-system CSYS1, work for PSYS4 to CSYS2, and work for PSYS5 to CSYS3.

When GDDRMAIN is started on each B-system, an attempt is made to verify BCPii connectivity to the indicated CPC (NETID.NAU). Failure during this check will result in the display of GDDR error message GDDM195E to alert the user to a networking problem.

# Create SRDF parameter members (SIT*xxxxx*)

GDDR SIT*xxxxx* members contains SRDF parameters customized to a specific configuration type and state. They have hardcoded member names (listed in Table 10 on page 119), and are customized when the configuration is defined to GDDR.

At script runtime, GDDR uses a utility named in the Utility.IEBGENER GDDR parameter to copy the appropriate SIT*xxxxx* member over the currently used SRDF Host Component RDFPARM member.

The SIT*xxxxx* members in *hlq*.GDDR.*vrm*.SAMPLIB with the names listed in Table 10 on page 119 must be created in the PDS pointed to by the SRDF entries within the Define EMC Mainframe Enablers STCs panel (M,P,H,E). These members contain complete and identical SRDF Host Component initialization parameters, but are different with respect to MSC group definition, as described in Table 10.

IMPORTANT

These members define MSC gatekeeper devices. Avoid using these gatekeepers elsewhere in the GDDR-managed configuration.

**Note:** WF indicates the MSC weight factor. The primary server is the server running with MSC_WEIGHT_FACTOR=0. The secondary server is the server running with MSC_WEIGHT_FACTOR=2.

The MSC group names specified in SIT*xxxxx* members must match those defined in the GDDRPARM MSCGROUP parameters (described in "MSCGROUP" on page 266) for the affected site pair(s) of the configuration. The MSCGROUP names specified in SIT*xxxxx* members are tied exclusively to the SRDF/A site pair and direction that will be active when the member is used, and are not related to the active MSC mode or topology.

**Table 10** SRDF parameter members

| Member | Description |
|---|---|
| SITEQR1 | Defines SRDF/SQAR mode, current primary region=1, WF=0 |
| SITHQR1 | Defines SRDF/SQAR mode, current primary region=1, WF=2 |
| SITEQR2 | Defines SRDF/SQAR mode, current primary region=2, WF=0 |
| SITHQR2 | Defines SRDF/SQAR mode, current primary region=2, WF=2 |
| The following are needed for the out-of-region recovery scripts with SRDF/A and in cases where SRDF/A cannot be resumed in SQAR mode, for example, after LDR at DC1 or when the links required for the RCVRY groups are unavailable: | |
| SITEUQC1 | Defines an MSC group in MSC-only mode, current primary region=1, DC1-DC3 leg, WF=0. Specifies the DC1->DC3 MSC group name. |
| SITHUQC1 | Defines an MSC group in MSC-only mode, current primary region=1, DC1-DC3 leg, WF=2. Specifies the DC1->DC3 MSC group name. |

**Table 10** SRDF parameter members

| Member | Description |
|---|---|
| SITEUQC2 | Defines an MSC group in MSC-only mode, current primary region=1, DC2-DC4 leg, WF=0. Specifies the DC2->DC4 MSC group name. |
| SITHUQC2 | Defines an MSC group in MSC-only mode, current primary region=1, DC2-DC4 leg, WF=2. Specifies the DC2->DC4 MSC group name. |
| SITEUQC3 | Defines an MSC group in MSC-only mode, current primary region=2, DC3-DC1 leg, WF=0. Specifies the DC3->DC1 MSC group name. |
| SITHUQC3 | Defines an MSC group in MSC-only mode, current primary region=2, DC3-DC1 leg, WF=2. Specifies the DC3->DC1 MSC group name. |
| SITEUQC4 | Defines an MSC group in MSC-only mode, current primary region=2, DC4-DC2 leg, WF=0. Specifies the DC4->DC2 MSC group name. |
| SITHUQC4 | Defines an MSC group in MSC-only mode, current primary region=2, DC4-DC2 leg, WF=2. Specifies the DC4->DC2 MSC group name. |

The members listed in Table 10 must be created on each C-system, in a PDS pointed to on the Define EMC Mainframe Enablers STCs panel (M,P,H,E). This panel defines one additional member (default: SRDFSTAR) which is actively used by the SRDF Host Component instance on the specified C-system.

# Set SCF initialization parameters

On each managed system and C-system, ensure the following SCF parameters are set correctly:

- SCF.MSC.ADCOPY.ONDROP is set to NO.

  (This is the default setting for this parameter.) Setting this parameter to NO eliminates a potential conflict between GDDR processing and SRDF Host Component actions following GDDR actions to stop SRDF/A during planned or unplanned operations.

- SCF.MSC.ENABLE is set to NO.

  Setting this parameter to NO prevents “accidental” MSC activation when software is recycled outside the control of GDDR.

  The Mainframe Enablers software, SCF in particular, is never stopped on the master C-system. For a planned shutdown of the master C-system, use option T, Transfer Master C-System in the Setup and Maintenance Menu panel (M) to move the master C-system function. Use the GDDRPXAS Transfer AutoSwap Owner script and the GDDRPA36, Planned Star-HA Takeover script, found on the Select Script to Run on GDDRPLEX panel (S) under the Special scripts heading to move the AutoSwap owner and the MSC control function to the non-master C-system at the primary or secondary site.

  When the original master C-system is restarted, move the applicable control functions back to it. If these guidelines are followed, GDDR will ensure MSC is enabled when and where needed for its automation purposes.

- SCF.MSC.OVERWRITE is set to YES.

  This setting allows GDDR to overwrite an existing MSC environment when required during a GDDR automation sequence.

# Authorize ConGroup to use Trip API

Ensure that the user authorized to the Dell EMC Consistency Groups (ConGroup) started tasks on the C-systems is also authorized with READ access to the EMC.ADMIN.FNC.CG.TRIP facility class.

The authorization commands are:

```
RDEFINE FACILITY EMC.ADMIN.FNC.CG.TRIP UACC(NONE)
PERMIT EMC.ADMIN.FNC.CG.TRIP CLASS(FACILITY) ID(uuuuuuu) ACCESS(READ)
```

Where *uuuuuuu* is the user ID assigned to the ConGroup STC on the C-systems at DC1 and DC2.

**Note:** The *Consistency Groups for z/OS Product Guide* provides detailed instructions.

# Configure ConGroup

ConGroup parameter members for C-systems as well as managed systems at DC1 and DC2 must be identical, and specify as owner, the SMFID for the C-system at DC2. Each ConGroup parameter member must define two consistency groups, one including the DC1 devices and the other including the DC2 devices.

ConGroup parameter members for C-systems as well as managed systems at DC3 and DC4 must be identical, and specify as owner, the SMFID for the C-system at DC4. Each ConGroup parameter member must define two consistency groups, one including the DC3 devices and the other including the DC4 devices.

The ConGroup configuration parameters must be the same for the C-system at a site and all the managed systems at that same site.

Specify the MODE ConGroup parameter with the MULTI keyword. Use of the MULTI keyword enables the use of an internal lock, ALL-CONGROUPS to serialize many global operations to ensure the integrity of ConGroup functionality.

See the *Consistency Groups for z/OS Product Guide* for more information about these configuration parameters.

# Set up ConGroup automated startup

By default, GDDR does *not* perform ConGroup Stop/Start commands to refresh ConGroup parameters. There are two exceptions:

- You can use the Perform EMCCGRP Shutdown and Perform EMCCGRP Startup call overrides to force GDDR to perform stop/start instead of using the REFRESH command.

  **Note:** "GDDR script call overrides" on page 85 discusses call overrides.

- In out-of-region recovery scenarios the ConGroup STC will usually not be running at the time GDDR would do a ConGroup REFRESH command. GDDR detects this and automatically performs a start of ConGroup, regardless of call override settings.

If the ConGroup STC is found to be running at the time GDDR needs to issue a ConGroup REFRESH command, the use of ConGroup Stop/Start commands is controlled by the Perform EMCCGRP Shutdown and Perform EMCCGRP Startup call overrides, which are available for certain scripts. When invoked, these call overrides stop or start the ConGroup started tasks running on all z/OS systems under GDDR management; this includes C-systems, production systems, and contingency systems.

## Specify ConGroup STC parameter member

GDDR allows some flexibility in the way you specify the location of the software parameters for the ConGroup STCs on each z/OS system.

The //CONFIG member included in the ConGroup STC startup proc contains the ConGroup startup parameters which define the consistency group. The name for the //CONFIG member can be specified in two ways:

1. Hardcode the name within the startup JCL.

   If the //CONFIG member name is hard-coded within the startup JCL, then any GDDR script that starts ConGroup creates a ConGroup startup command string similar to the following:

   ```
   S cgstcname
   ```

   Where *cgstcname is t*he ConGroup proc name.

2. Use a procedure substitution variable called MBR to reference a PDS member name.

   If the JCL procedure substitution variable MBR is used, then any GDDR script that starts ConGroup creates a ConGroup startup command string similar to the following:

   ```
   S cgstcname,MBR=cgrp_mbr
   ```

   Where:

   - *cgstcname is t*he ConGroup proc name.

- MBR is a JCL procedure substitution variable name required by GDDR which must be defined in your JCL procedure. *hlq*.GDDR*vrm*.SAMPLIB(GDDRCGRP) shows the usage of this variable.
- *cgrp_mbr* is the member name assigned to variable MBR to be substituted in the //CONFIG DD.

## Determine which command string to use

In the Define EMC Mainframe Enablers STCs panel (M,P,H,E), define the following parameters:

**Note:** "Define Mainframe Enablers STCs (M,P,H,E)" on page 337 provides more information about this panel.

- System

  The MVS system name of the system to which the statement applies.

- STC Name

  The started task name associated with STC Type=CG. GDDRCGRP is recommended, but you may select a name that matches your actual ConGroup procedure name *cgstcname*.

- Parameter Dataset and Member(s)

  X(CGRPCAX6) is recommended, but you may specify a member name of your choice that matches your actual *cgrp_mbr* member name. X is used as a DUMMY dataset name for ConGroup parameters, as the actual dataset name is not relevant to constructing a ConGroup start command.

The GDDR scripts use the variables to determine which of the command strings to use.

- If System, CG, STC Name, and Parameter Dataset name are supplied in the Define EMC Mainframe Enablers STCs panel (M,P,H,E), then GDDR uses this value and builds the following command string:

  ```
  S cgstcname,MBR=cgrp_mbr
  ```

  Where *cgrp_mbr* is the value assigned to procedure variable MBR.

- If a parameter member name is not supplied for STC Type=CG in the Define Dell Mainframe Enablers STCs panel (M,P,H,E), then the value of *cgstcname* is derived from the STC Name field associated with STC Type=CG in this panel. GDDR assumes that the //CONFIG DD member name value is hardcoded in the JCL procedure and builds the following command string:

  ```
  S cgstcname
  ```

# Configure MSC

This section describes how to define the MSC environment required by SRDF/SQAR.

Figure 27 illustrates the MSC servers in SRDF/SQAR configuration.

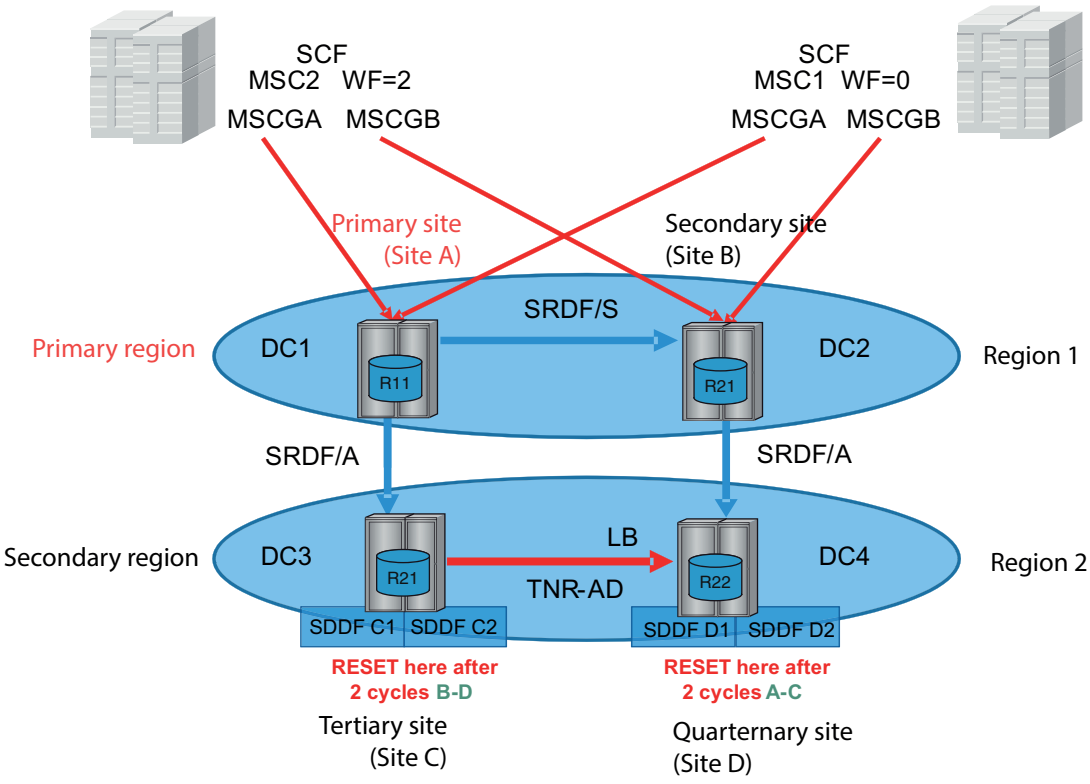



**Figure 27** MSC servers in SRDF/SQAR mode

## MSC statement definitions

A SQAR MSC group is designated with the keyword ",SQAR" after the MSC group name, on the MSC_GROUP_NAME statement. Two MSC_GROUP_NAME statements with accompanying dependent statements MSC_SQAR, MSC_INCLUDE_SESSION, and MSC_GROUP_END are required to form a SQAR configuration.

**Note:** For a detailed description of each MSC statement, see the *SRDF Host Component for z/OS Product Guide*.

The MSC_SQAR statement contains the consistency group name of the primary site followed by the MSC group name of the secondary SQAR group. The consistency group name must be different on each of the primary and secondary MSC SQAR group definitions.

The MSC_INCLUDE_SESSION statement is specified similar to a MSC_INCLUDE_SESSION statement for an SRDF/Star configuration; the second SRDF group is the recovery group from Site C to Site D or Site D to Site C. Each R2 device must be related to the other R2 device, which must form a "square" when viewing the relationship of each set of devices: DC1 (R11), DC2 (R21), DC3 (R21), and DC4 (R22).

MSC_INCLUDE_SESSION requirements are as follows:

- The number of asynchronous (SRDF/A) groups configured for the two SQAR groups must be equal.
- The number of recovery groups cannot be more than the number of asynchronous groups.
- The total number of SRDF/S and SRDF/A devices must be equivalent in each SQAR group.
- Only one SQAR configuration (two related MSC SQAR groups) will be allowed in a single MSC environment.
- STAR and SQAR definitions are not supported in the same MSC environment.

The MSC_CYCLE_TARGET and MSC_TAKEOVER_THRESHOLD statements are optional. If not specified, the cycle target value defaults to 30 seconds, and the takeover threshold value defaults to 32. MSC_TAKEOVER_THRESHOLD is the number of consecutive successful cycle switches that must be completed by a secondary server in a High Availability (HA) environment before issuing the SCF15C8W message.

## Sample RDFPARMS for Region 1

These parameters are all in the same member and are available to the C-systems at DC1 and DC2.

```
MSC_GROUP_NAME=mscgrpA,SQAR
MSC_SQAR=cngrpA,mscgrpB
MSC_INCLUDE_SESSION=ccuu1,(async_srdfgrp1),(rcvy_srdfgrp1)
MSC_INCLUDE_SESSION=ccuu2,(async_srdfgrp2),(rcvy_srdfgrp2)
MSC_INCLUDE_SESSION=ccuu3,(async_srdfgrp3),(rcvy_srdfgrp3)
MSC_INCLUDE_SESSION=ccuu4,(async_srdfgrp4),(rcvy_srdfgrp4)
MSC_CYCLE_TARGET=xx
MSC_TAKEOVER_THRESHOLD=yy
MSC_GROUP_END

MSC_GROUP_NAME=mscgrpB,SQAR
MSC_SQAR=cngrpB,mscgrpA
MSC_INCLUDE_SESSION=ccuu5,(async_srdfgrp5),(rcvy_srdfgrp1)
MSC_INCLUDE_SESSION=ccuu6,(async_srdfgrp6),(rcvy_srdfgrp2)
MSC_INCLUDE_SESSION=ccuu7,(async_srdfgrp7),(rcvy_srdfgrp3)
MSC_INCLUDE_SESSION=ccuu8,(async_srdfgrp8),(rcvy_srdfgrp4)
MSC_CYCLE_TARGET=xx
MSC_TAKEOVER_THRESHOLD=yy
MSC_GROUP_END
```

## Sample RDFPARMS for Region 2

These parameters are all in the same member and are available to the C-systems at DC3 and DC4). Note that in this example, there are four SRDF/A groups, but only three RCVY groups. ASYNC groups ra3 and ra4 use the same RCVY group.

```
MSC_GROUP_NAME=mscgrpC,SQAR
MSC_SQAR=cngrpC,mscgrpD
MSC_INCLUDE_SESSION=ccuu1,(async_srdfgrp1),(rcvy_srdfgrp5)
MSC_INCLUDE_SESSION=ccuu2,(async_srdfgrp2),(rcvy_srdfgrp6)
MSC_INCLUDE_SESSION=ccuu3,(async_srdfgrp3),(rcvy_srdfgrp7)
MSC_INCLUDE_SESSION=ccuu4,(async_srdfgrp4),(rcvy_srdfgrp7)
MSC_CYCLE_TARGET=xx
MSC_TAKEOVER_THRESHOLD=yy
MSC_GROUP_END
```

```
MSC_GROUP_NAME=mscgrpD,SQAR
MSC_SQAR=cngrpD,mscgrpC
MSC_INCLUDE_SESSION=ccuu5,(async_srdfgrp5),(rcvy_srdfgrp5)
MSC_INCLUDE_SESSION=ccuu6,(async_srdfgrp6),(rcvy_srdfgrp6)
MSC_INCLUDE_SESSION=ccuu7,(async_srdfgrp7),(rcvy_srdfgrp7)
MSC_INCLUDE_SESSION=ccuu8,(async_srdfgrp8),(rcvy_srdfgrp7)
MSC_CYCLE_TARGET=xx
MSC_TAKEOVER_THRESHOLD=yy
MSC_GROUP_END
```

## MSC SWITCH command

The MSC SWITCH command is used by GDDR in a site-swap situation. It swaps the contents of the MSC control blocks; the MSCGA definition will become MSCGB and vice-versa.

The SWITCH command allows GDDR to perform planned and unplanned in-region site swaps while maintaining continuous consistency protected replication on the MSCGB group. It avoids the SRDF replication interruption and subsequent resumption and resynchronization exposures. It shortens GDDR script elapsed time considerably, and significantly reduces the number of operations GDDR has to perform during these scripts.

F GDDRSCF,MSC,SWITCH,MSCG(*mscgname*)

The SWITCH command is used in the following GDDR scripts: GDDRPA42, GDDRPA21, GDDRUP31 (with SRDF/S preference), and GDDRPA0A.

## MSC Cleanup utility

The MSC Cleanup utility (ME) performs case processing and resets host intervention only. The M6 utility must be run to remove the SQAR environment. The M9 utility must be used to drop SRDF/A in cases where it was not possible for MSC to perform the drop.

## MSC, ConGroup, and AutoSwap integration

The following items describe integration between MSC, ConGroup, and AutoSwap operations under the control of GDDR when in an SRDF/SQAR with AutoSwap configuration.

MSC in a SQAR configuration runs in HA mode and requires AutoSwap:

- The master C-system at Site B runs as the primary MSC server and AutoSwap owner.
- The C-system at Site A runs as the secondary MSC server.
- Four C-systems are required.
- MSC gatekeeper is not part of the AutoSwap group.
- Cross-site channel connectivity is available for the DC1-DC2 site pair and the DC3-DC4 site pair.

MSC in SQAR mode detects the following events and reacts as follows:

- ConGroup trip:
  - Stop SDDF reset at Site C and Site D.
  - Continue SRDF/A cycle switching A-C and set the C-ahead flag.
  - Drop the B-D leg.
- ConGroup resumption:
  - Wait for ConGroup Enable and then clear the C-ahead flag.
  - Wait for GDDR instructions to resume B-D cycle switching (MSC,RESTART command).
  - Resume SDDF reset at Site C and Site D.
- SRDF/A drop:
  - For a drop on the A-C leg, drop only the other A-C leg SRDF/A sessions
    - SDDF sessions at D will age automatically as A-C no longer cycles
  - For a drop on the B-D leg, drop only the other B-D leg SRDF/A sessions
    - SDDF sessions at C will age automatically as B-D no longer cycles
- Planned or unplanned swap:
  - Stop SDDF reset at Site C and Site D.
  - Continue SRDF/A cycle switching B-D.
    - Avoid SRDF/A drop.
  - SRDF/A is dropped on the A-C leg by the operating environment.
  - Unregister from ConGroup.
  - Wait for the GDDR SWITCH,MSCG(...) command to swap the contents of the MSC control blocks with MSC and SRDF/A active A-C and B-D (done when access to old primary is restored).

# Set up GDDR security

Complete the following tasks to define the security environment required by GDDR:

- Define GDDR RACF functional groups
- Define GDDR ISPF interface security
- Authorize RACF for HMC LPAR actions
- Authorize EMCSAFI security interface
- Verify module and RACF-protected resource authorization
- Update ACF2 TSO command list for command limiting

## Define GDDR RACF functional groups

Define the RACF groups listed in Table 11 to grant the appropriate access based on job function.

**Table 11** RACF functional groups

| Functional group | Description | Required or optional |
|---|---|---|
| GDDR$ADM | For systems programmers who install and configure GDDR. For GDDR administrators who configure GDDR.<br>**Note:** If the GDDR$REV groups are not defined, the GDDR$ADM group should be given the authorizations recommended for the GDDR$REV groups. | Required |
| GDDR$STC | For the GDDR monitors, planned and unplanned processes. | Required |
| GDDR$USR | For operators and operations support staff who operate GDDR. | Required |
| GDDR$REV | For management or interested parties who require the GDDR Parameter Review capability. | Optional |

## Summary of RACF permissions

Table 12 provides an overview of the RACF profiles and permissions required to protect GDDR resources.

**Table 12** RACF permissions

| GDDR resource owning group | GDDR STC user group | GDDR user group | GDDR reviewer group | Admin/ Sysprog user group |
|---|---|---|---|---|
| GDDR$ | GDDR$STC **Access needed** | GDDR$USR **Access needed** | GDDR$REV **Access needed** | GDDR$ADM **Access needed** |
| **Dataset profile** | | | | |
| *hlq*.GDDR*vrm*..LINKLIB [a] | READ | - | READ | ALTER |
| *hlq*.GDDR*vrm*..ISPMLIB | READ | READ | READ | ALTER |
| *hlq*.GDDR*vrm*..OPSEXEC | READ | READ | READ | ALTER |
| *hlq*.GDDR*vrm*..ISPPLIB | - | READ | READ | ALTER |
| *hlq*.GDDR*vrm*..PROCLIB | READ | - | - | ALTER |
| hlq.GDDR*vrm*.REXX | READ | - | - | ALTER |
| *hlq*.GDDR*vrm*..ISPSLIB | READ | READ | - | ALTER |
| *hlq*.GDDR*vrm*..PARMLIB | READ | - | - | ALTER |
| *hlq*.GDDR*vrm.csys*.DIV.* | ALTER | - | - | ALTER |
| *hlq*.GDDR*vrm*..* | - | - | - | ALTER |
| *workhlq.qualifier*.* [b] | ALTER | ALTER | - | ALTER |
| *hlq*.GDDR*vrm*..BKUPVARS.CNTL | ALTER | READ | - | ALTER |
| *hlq*.GDDR*vrm*.PARMS.BKUP | ALTER | - | READ | ALTER |
| *hlq*.GDDR*vrm*.PARMS.WORK | ALTER | - | ALTER | ALTER |
| *hlq*.GDDR*vrm*.PARMS.LAPD | ALTER | READ | READ | ALTER |
| **FACILITY profile** | | | | |
| GDDR.HMC.LISTOBJECTS | READ | READ | - | READ |
| GDDR.HMC.GETMSGS | READ | READ | - | READ |
| GDDR.GLOBAL.VARIABLE.ACCESS | UPDATE | UPDATE | UPDATE | UPDATE |
| HWI.APPLNAME.HWISERV | READ | READ | - | READ |
| HWI.TARGET.*netid*.nau#1 | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau#2 | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau#n | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau#1.image#1 | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau#1.image#2 | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau#2.image#1 | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau#2.image#2 | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.ima*.* | ALTER | ALTER | - | ALTER |
| HWI.TARGET.*netid*.nau.* | ALTER | ALTER | - | ALTER |
| **TSOAUTH profile** | | | | |
| OPER | READ | - | - | - |
| **SURROGAT profile** | | | | |
| *userid*.SUBMIT [c] | READ | READ | - | READ |
| **JESSPOOL profile** | | | | |
| *Jes2node*.GDDR.*.*.*.* [d] | - | READ | - | ALTER |
| **OPERCMDS profile** | | | | |
| MVS.MODIFY.STC.*.* | - | - | - | - |

a. *hlq* is any dataset high level qualifier, if one is used.
b. *workhlq.qualifier*.* is defined using the Script JCL Parameters panel (M,P,O,J), field 'Work HLQ' as shown on page 344. The default value for *workhlq* is "GDDR". The user IDs assigned to GDDR started tasks, as well as user IDs authorized to submit GDDR scripts must be authorized with ALTER to this HLQ.
c. *userid* is the Surrogate User ID defined on the Script JCL Parameters panel (M,P,O,J).
d. *Jes2node* is the JES2 node name of the C-system. See "Determining JES2 node name" on page 132.

All GDDR RACF non-generic profiles should have a universal access (UACC) of NONE.

**Note:** Use the JCL provided in *hlq*.GDDR*vrm*.SAMPLIB member GDDCRACJ to define RACF authorizations using *hlq*.GDDR*vrm*.SAMPLIB member GDDCRACF for C-systems and member GDDPRACF for managed systems. SAMPLIB members GDDCRACD and GDDPRACD are provided for ease of deleting RACF authorization definitions from systems.

## Determining JES2 node name

**Note:** This procedure applies to JES2 installations only.

To determine the JES2 node name, issue the JES2 console command $DNODE,OWNNODE=YES on the appropriate C-system.

The output of the JES2 command is as follows:

```
$HASP826 NODE(1)
   $HASP826 NODE(1)   NAME=MFSYS3,STATUS=(OWNNODE),AUTH=(DEVICE=YES,
   $HASP826           JOB=YES,NET=NO,SYSTEM=YES),TRANSMIT=BOTH,
   $HASP826           RECEIVE=BOTH,HOLD=NONE,PENCRYPT=NO,
   $HASP826           SIGNON=COMPAT,DIRECT=NO,ENDNODE=NO,REST=0,
   $HASP826           SENTREST=ACCEPT,COMPACT=0,LINE=0,LOGMODE=,
   $HASP826           LOGON=0,NETSRV=0,OWNNODE=YES,
   $HASP826           PASSWORD=(VERIFY=(NOTSET),SEND=(NOTSET)),
   $HASP826           PATHMGR=YES,PRIVATE=NO,SUBNET=,TRACE=YES
```

The actual JES2 node name is identified on the NAME=*output* statement.

## OPERCMDS class resource definitions (optional)

If command authorization checking is in place, see the following OPERCMDS class resource definitions in Table 13. Sample commands are supplied in *hlq*.GDDR&*vrm*.SAMPLIB(GDDCRACF).

**Table 13** RACF permissions, OPERCMDS class

| **GDDR resource owning group**<br>CLASS OPERCMDS | Command/keyword | **GDDR$STC** | **GDDR$USR** | **GDDR$ADM** |
|---|---|---|---|---|
| MVS.REPLY | REPLY | READ | READ | |
| MVS.MODIFY.STC.*.* | MODIFY *jobname*<br>MODIFY *jobname.id*<br>MODIFY *id* | UPDATE | UPDATE | - |

Apply the RACF RDEFINE and PERMIT commands for the MVS.MCSOPER facility class which are contained in *hlq*.GDDR*vrm*.SAMPLIB(GDDCRACF).

IBM z/OS apar OA26369, which is contained in PTF UA48307 for z/OS 1.9, UA48308 for z/OS 1.10 and UA48309 for z/OS 1.11 enforces the authorization check on facility class MVS.MCSOPER.

## Define GDDR ISPF interface security

You may choose to implement role-based access controls with GDDR ISPF interface profiles. Your site's security authorization product controls access to the entire GDDR ISPF interface, to selected menus, and to selected actions within menus.

**Table 14** Summary of GDDR ISPF RACF permissions (1 of 2)

| Facility profile | Description and panel ID | GDDR $ADM group | GDDR $USR group | GDDR $REV group | GDDR $TAP group |
|---|---|---|---|---|---|
| GDDRISPF.ACCESS | GDDR Primary Options Menu (GDDRPRIM) | READ | READ | READ | READ |
| GDDRISPF.ACTCBU.*site.cpc* | Authorizes CBU actions by SITE and CPC | READ | READ | | |
| GDDRISPF.ACTIONS.* | Authorizes all GDDR CBU, HMC and SYSPLEX actions | | | | |
| GDDRISPF.ACTIONS.ACCESS | Perform GDDR Actions (GDDRPRIM) | READ | READ | | |
| GDDRISPF.ACTIONS.CBU.ACCESS | Perform HMC CBU actions (GDDRACT0) | READ | READ | | |
| GDDRISPF.ACTIONS.HMC.ACCESS | Perform HMC LPAR actions (GDDRACT0) | READ | READ | | |
| GDDRISPF.ACTIONS.HMC.*system_name* | Authorizes HMC actions by system name | READ | READ | | |
| GDDRISPF.ACTIONS.SYSPLEX | Manage Couple Datasets (GDDRACT0) | READ | READ | | |
| GDDRISPF.SCRIPTS.* | Authorizes all Script management actions | | | | |
| GDDRISPF.SCRIPTS.CHECKUP | Perform pre-script checkup (GDDRPRIM) | READ | READ | | |
| GDDRISPF.SCRIPTS.MANAGE.JCL | Job cards for your user (GDDR0PRJ) | READ | READ | | |
| GDDRISPF.SCRIPTS.RUN.ACCESS | Run GDDR Scripts (GDDRPRIM) | | READ | | |
| GDDRISPF.SCRIPTS.VIEWSTATS | View GDDR Script Statistics (GDDRPRIM) | | READ | | |
| GDDRISPF.SETUP.* | All Setup and Maintenance Actions (GDDRMNT0) | | | | |
| GDDRISPF.SETUP.ACCESS | Setup and Maintenance Submenus (GDDRMNT0) | READ | READ | | READ |
| GDDRISPF.SETUP.AUTO | GDDR ON/OFF Automation (GDDRPRIM)<br>GDDR ON/OFF Automation (GDDRMNT0) | READ | | | |
| GDDRISPF.SETUP.DEBUG | Message, Debug and Trace options (GDDRMNT0) | READ | READ | | READ |
| GDDRISPF.SETUP.DISCOVERHMC | Perform HMC Discovery (GDDRMNT0)<br>Perform HMC Discovery (GDDRACT0) | READ | | | |
| GDDRISPF.SETUP.PARMS.* | Authorizes all Parameter Edit, Validate and Activate actions | | | | |
| GDDRISPF.SETUP.PARMS.ACCESS | Perform GDDR Setup and Maintenance Functions (GDDRMNT0) | READ | READ | READ | READ |
| GDDRISPF.SETUP.PARMS.ACTIVATE | Activate GDDR Parameter Set (GDDRPRM0) | READ | | | |
| GDDRISPF.SETUP.PARMS.BACKUP | Manage GDDR Parameter backups (GDDRPRM0) | READ | | | READ |

**Table 14** Summary of GDDR ISPF RACF permissions (2 of 2)

| | | | | | |
|---|---|---|---|---|---|
| GDDRISPF.SETUP.PARMS.FORCE | Authorizes the override of the parameter edit-in-progress serialization lock.<br>Parameter Management Options Menu (GDDRPRMI)<br>Select Parameter Input Dataset (GDDRPRM0)<br>Define Configuration Basics (GDDRPRM0)<br>Define Data Storage objects (GDDRPRM0)<br>Define Host objects (GDDRPRM0)<br>Specify GDDR Options (GDDRPRM0)<br>Validate GDDR Parameter Set (GDDRPRM0) | READ | | | |
| GDDRISPF.SETUP.PARMS.LOAD | Authorizes Parameter Edit and Validation actions<br>Update personal GDDR ISPF profile, job cards for your user (GDDR0PRJ)<br>Parameter Management Options Menu (GDDRPRMI)<br>Select Parameter Input Dataset (GDDRPRM0)<br>Define Configuration Basics (GDDRPRM0)<br>Define Data Storage objects (GDDRPRM0)<br>Define Host objects (GDDRPRM0)<br>Specify GDDR Options (GDDRPRM0)<br>Validate GDDR Parameter Set (GDDRPRM0) | READ | | | |
| GDDRISPF.SETUP.PARMS.REVIEW | Authorizes Parameter Review actions<br>Parameter Management Options Menu (GDDRPRMI)<br>Select Parameter Input Dataset (GDDRPRM0)<br>Define Configuration Basics (GDDRPRM0)<br>Define Data Storage objects (GDDRPRM0)<br>Define Host objects (GDDRPRM0)<br>Specify GDDR Options (GDDRPRM0) | | | READ | |
| GDDRISPF.SETUP.QUEUE | Manage GDDR Internal Command Queue (GDDRMNT0) | READ | | | |
| GDDRISPF.SETUP.REFRESH | Refresh GDDR Message Table (GDDRMNT0) | READ | | | |
| GDDRISPF.SETUP.SITEROLES | Transfer Master C-System (GDDRRMFXR) | READ | | | |
| GDDRISPF.SETUP.STATE | Manage GDDR System variables (GDDRMNT0) | READ | | | |
| GDDRISPF.STEM.ACCESS | STC Execution Manager | READ | READ | READ | READ |
| GDDRISPF.VIEW.CONFIG | View GDDR Configuration (GDDRPRIM) | READ | READ | | READ |

Member GDDIRACF in *hlq*.GDDR*vrm*.SAMPLIB lists the RACF commands used for GDDR ISPF permissions. These commands are optionally used to protect GDDR functions.

## Authorize RACF for HMC LPAR actions

The facility profile GDDRISPF.ACTIONS.HMC.*system_name* provides for site-specific authorization of GDDR LPAR processing.

Authorization is enabled through the Perform HMC LPAR Actions panel (A,L). Following your specification of GDDRISPF.ACTIONS.HMC facility profiles, the Perform HMC LPAR Actions panel (A,L) will only list the system names for actions by authorized GDDR users. Unprotected system names will display for any GDDR user.

The value of *system_name* corresponds to the name found in the system field of the Define Managed LPARs panel (M,P,H,L). You can use generic or discrete system names in the facility profile.

The following rules apply to generic profile names, where valid generic characters are *, %, and **:

- Specify **%** in the profile name to match any single non-blank character (except a period) in the same position of the system ID name.
- Specify * or ** in the profile name to match more than one character in the same position of the system ID name.

The *z/OS Security Server RACF Command Language Reference* (SA22-7687) provides detailed descriptions and examples that illustrate how to specify generic profile names.

**Note:** The sample GDDRISPF.ACTIONS.HMC facility profile definition, supplied with *hlq*.GDDR*vrm*.SAMPLIB member GDDIRACF authorizes all systems to users who are connected to the GDDR$USR group.

### Examples

The following examples demonstrate the use of the GDDRISPF.ACTIONS.HMC facility profile:

```
RDEFINE FACILITY GDDRISPF.ACTIONS.HMC.** UACC(NONE) OWNER(GDDR$ADM)
PERMIT GDDRISPF.ACTIONS.HMC.ZOSESYS* CLASS(FACILITY) ACCESS(READ) ID(GDDR)
```

In the example, the Perform HMC LPAR Actions panel (A,L) at a given C-system will display all system names defined in the System field of the Define Managed LPARs panel (M,P,H,L) that match the generic name 'ZOSESYS*' for all users connected to the group 'GDDR'. System names which do not match 'ZOSESYS*' are protected and will not be displayed on any Perform HMC LPAR Actions panel (A,L).

```
RDEFINE FACILITY GDDRISPF.ACTIONS.HMC.ZOSESYS1 UACC(NONE) OWNER(GDDR$ADM)
PERMIT GDDRISPF.ACTIONS.HMC.ZOSESYS1 CLASS(FACILITY) ACCESS(READ) ID(SYSPGM1)
```

In the example, the Perform HMC LPAR Actions panel (A,L) at a given C-system will display only system name ZOSESYS1 for user SYSPGM1. All other systems defined in the System field of the Define Managed LPARs panel (M,P,H,L) are not protected and therefore will be displayed on the Perform HMC LPAR Actions panel (A,L).

## Authorize EMCSAFI security interface

If you have protected access to AutoSwap, ConGroup and SRDF Host Component commands, authorize the GDDR user ID and the GDDR$STC group with SAF profiles described in the *Mainframe Enablers Installation and Customization Guide* for the named applications.

If TimeFinder/Mirror for z/OS and TimeFinder/Mirror clone emulation are in use and commands are protected, the GDDR user ID and the GDR$STC group also require authorization.

Similarly, if devices to be managed by GDDR have been protected using the EMCSAFI security interface of ResourcePak Base, authorization must be added for the GDDR user ID and the GDDR$STC group.

The *Mainframe Enablers Installation and Customization Guide* provides information about the security interface and the class and resource names used.

# Verify module and RACF-protected resource authorization

After completing the steps listed in "Customize SYS1.PARMLIB(IKJTSOxx)" on page 111 and in "Set up GDDR security" on page 130, check that GDDR load modules are properly authorized and accessible and that RACF resources have been properly defined and authorized using member GDDRECHK in *hlq*.GDDR*vrm*.SAMPLIB.

**Note:** "GDDR Environment Check utility (GDDRECHK)" on page 442 provides instructions on how to use the GDDRECHK utility.

## RACF-protected resource authorization and verification SAMPLIB members

Table 15 lists *hlq*.GDDR*vrm*.SAMPLIB members used to define and validate GDDR security controls.

**Table 15** SAMPLIB security members

| Member | Description |
|---|---|
| GDDCRACF | Define RACF C-system resources |
| GDDCRACD | Delete RACF C-system resources |
| GDDCRACJ | JCL to delete/define C-system resources |
| GDDIRACF | Define RACF GDDR ISPF resources |
| GDDIRACD | Delete RACF GDDR ISPF resources |
| GDDIRACJ | JCL to delete/define GDDR ISPF resources |
| GDDPRACF | Define RACF production system resources |
| GDDPRACD | Delete RACF production system resources |
| GDDPRACJ | JCL to delete/define production system resources |
| GDDRECHK | Verify GDDR load module authorizations and RACF access authorizations |

# Update ACF2 TSO command list for command limiting

If your site is using ACF2 for security, and has implemented the ACF2 TSO command list for command limiting, add the following modules to the list:

**Note:** The following is a complete list of modules that might be required, including both GDDR and GDDR Tape installations.

```
GDDDLCIM - Issue operator message
GDDDLCPU - Update tape profile(s)
GDDDLCSH - Suspend DLm Heartbeat
GDDDLCVP - Validate Tape Profile(s) and report discrepancies
GDDDLVCS - Issue VTECFGSWAP
GDDDLVHC - DLMAUT Health Check utility
GDDDLVQC - Issue DEVSTATUS,CHECK
GDDDLVQU - Issue DEVSTATUS,QUIESCE
GDDDLVRP - Issue REPLSWAP,PLANNED
GDDDLVRR - Issue REPLSWAP,RESTORE
GDDDLVRU - Issue REPLSWAP,UNPLANNED
GDDDLVRV - Issue REPLSWAP,VALIDATE
GDDDLVSC - Issue DLMSNAP,CREATE
```

```
GDDDLVSD - Issue DLMSNAP,DESTROY
GDDDLVSV - Vary SMS STORGRP
GDDDLVSW - Discover tape devices in SMS STORGRP
GDDDLVUQ - Issue DEVSTATUS,UNQUIESCE
GDDDLVUS - Issue DEVSTATUS,QUIESCE
GDDDLVVF - Vary tape devices offline
GDDDLVVO - Vary tape devices online
GDDDLVWV - Vary tape devices online/offline for Switchover/Switchback
GDDI123A - Define Configuration Features
GDDI123C - Define GDDR Data Sets
GDDIBKUP - Parmlib backup
GDDICFG0 - Define Config Basics
GDDIHOBJ - Define Host Objects
GDDIHSYS - Define Managed Systems
GDDIMNT0 - Maintenance
GDDIMXFR - Transfer
GDDIOCOV - Default Script Call Overrides
GDDIOJCL - Script JCL Parameters
GDDIOPT0 - Options
GDDIOUTI - Utility Parameters
GDDIPACT - Activate GDDR Parameter set
GDDIPRIM - Primary Option menu
GDDIPRM0 - Parms
GDDIPRMI - Select Parm input DSN
GDDIPVAL - Validate GDDR Parameter set
GDDIRPBU - Backup
GDDISDLM - Define DLm Systems
GDDISDLP - Define DLm Profiles
GDDISPCK - Verify ISPF library allocation for GDDR ISPF interface
GDDISTOB - Define Data Storage Objects
GDDISTRT - ISPF Start program
GDDITUNE - Tuning Values
GDDIULBL - GDDR user labels
GDDPPACT - Post-activation program
GDDR999I - Display ConGroup and AutoSwap states
GDDRACDD - Automated Configuration Discovery for DASD
GDDRBCVG - BCVGROUP utility
GDDRCLN5 - List SDDF sessions
GDDRCPL0 - Manage coupled datasets
GDDRDFCK - Perform RDF group check
GDDRDIRS - Display overview of RDF directors
GDDRECG0 - ECGUTIL utility driver
GDDREE00 - Expected events utility
GDDREM01 - GDDR event monitor
GDDRG3AH - Run EHCMSCM6 MSC cleanup
GDDRG3AM - Run EHCMSCM9 RDF group cleanup
GDDRG3AQ - Run EHCMSCME MSC cleanup
GDDRGF0G - Wait for SRDF/A session consistency
GDDRGF0S - Manage HMC message scanner
GDDRGF3M - Wait for invalid tracks to synch up
GDDRGS0D - Refresh Message Table
GDDRGS0R - Reload GDDR message table
GDDRGVBB - Global variable backup
GDDRGVBX - Rewrite backup global variable
GDDRGVRL - Reload global variables (utility)
GDDRHMC2 - Perform HMC actions
GDDRKF0D - Activate or drop SRDF/A
GDDRKF13 - Cleanup ConGroup devices
GDDRKF20 - Perform BCV management
GDDRKF8P - Issue RDF-RSUM for SRDF/A leg
GDDRLPRF - Load profile processor
GDDRMIN0 - Invalid track monitor script interface
GDDRMSC0 - Issue MSC command
GDDRMSCP - Activate SRDF/Star on primary MSC server
GDDRMSCQ - Query MSC group
GDDROP20 - Scripts
```

```
GDDROP93 - GDDR Config
GDDROPDT - Debug
GDDROPGV - System
GDDROPHC - Checkup
GDDROPOS - Timing
GDDRPROF - Profile
GDDRRDF0 - Issue SC VOL command
GDDRRXSL - Start REXX routine using ISPEXEC SELECT
GDDRRXST - Start REXX routine using ISPSTART
GDDRSC05 - Perform RDF device status check
GDDRSC06 - Perform a series of RDF device status checks
GDDRSDDF - Verify Star/SQAR SDDF sessions
GDDRSRCK - Perform RDF group check for all C-systems
GDDRSTPW - Stop GDDR worker
GDDRSTR0 - Manage CF structures
GDDRSTR3 - Perform CF structure management
GDDRSTUB - Invoke GDDR REXX routine
GDDRTF00 - Query BCV state
GDDRTOPC - Check RDF topology map
GDDRTRCK - System call tracking monitor
GDDRUNME - Run EHCMSCME MSC cleanup
GDDRUP03 - Takeover system
GDDRUP04 - Warn operator of unplanned event
GDDRUPPI - Turn on planned IPL event
GDDRWARN - Issue warning message to operator
GDDRXACT - Script generator and executor
GDDRXBEG - Script prolog
GDDRXCLQ - Clear command queue on all C-systems
GDDRXCMR - Clear command queue
GDDRXD2G - Load generated script from dataset
GDDRXEND - Script termination
GDDRXG2D - Save generated script to dataset
GDDRXLGX - Display generated script
GDDRXPG1 - Expert system test driver
```

# Define GDDR datasets

Complete the following tasks to define GDDR datasets:

- Define global variable datasets
- Define parameter management datasets
- Define GDDRPARM file

C-systems use permanent global variable data-in-virtual (DIV) datasets, GDDR Parameter Wizard work datasets, and parameter backup datasets. Catalog these datasets on each C-system using the jobs supplied in *hlq*.GDDR*vrm*.SAMPLIB.

## Define global variable datasets

Define a VSAM linear dataset to manage global variables on each C-system. The recommended dataset name convention is *hlq*.GDDR*vrm*.*csys*.DIV. Use the JCL provided in *hlq*.GDDR*vrm*.SAMPLIB(GDDIVDEF).

## Define parameter management datasets

Customize and run the job in member GDDRABDS in *hlq*.GDDR*vrm*.SAMPLIB. Ensure it has run successfully.

### GDDR Parameter Wizard work dataset

The *GDDR Parameter Wizard work dataset* serves as a temporary store for GDDR parameters in preparation of GDDR parameter activation. This dataset contains your 'work-in-process', enabling you to assemble a complete parameter set by saving your input data from each of the parameter definition panels, and returning to the task at a later time.

The GDDR Parameter Wizard work dataset for your user ID is a dataset with the same attributes as a parameter backup dataset. The GDDR Parameter Wizard work dataset must be allocated as a PDS, with attributes FB and LRECL=80.

### Last activated parameter dataset

The *last activated parameter dataset* has the same allocation parameters as the GDDR Parameter Wizard work dataset and contains a copy of the GDDR Parameter Wizard work dataset used during the most recent GDDR parameter activation.

### Parameter backup datasets

It is recommended to use two *parameter backup datasets*:

- The parameter backup dataset for parameter management functions is defined to GDDR in your personal GDDR ISPF profile (option P in the Primary Options Menu panel).

  This dataset is used when you create a backup of GDDR parameters using option B, Manage GDDR Parameter backups, in the Parameter Management Options Menu panel (M,P). It is also used for implicit backups created before and after the update of GDDR parameters during GDDR parameter activation.

- The parameter backup dataset for backups performed when the GDDR Heartbeat Monitor initializes is defined in the Define GDDR Datasets panel (M,P,C,D) using the BKUPVARS field type. See "Define GDDR datasets (M,P,C,D)" on page 307 for details.

## Define GDDRPARM file

Allocate or choose a PDS, FB, LRECL=80, to store GDDRMAIN parameters. You can use *hlq*.GDDR*vrm*.PARMLIB for this purpose.

**Note:** "GDDRPARM statements" on page 256 describes GDDRMAIN parameters.

# Install GDDR started procedures

The GDDRMAIN task must be started and remain running on the master C-system at all times. It is recommended to run GDDRMAIN on all systems in the GDDR-plex, depending on feature requirements. Vary Tape Drive and Vary SMS processing require GDDRMAIN presence on affected systems. On C-systems, GDDRMAIN will automatically start GDDREVM and GDDRWORK started tasks. Before using GDDR, you must customize these started procedures and make them available.

**Note:** Use automation to start GDDRMAIN on the C-systems at system startup. No GDDR started tasks are to be started SUB=MSTR.

Whenever you make changes to your configuration, ensure you first update relevant GDDRMAIN parameters in the GDDRPARM dataset.

1. Update members GDDRMAIN, GDDRWORK, and GDDREVM in *hlq*.GDDR*vrm*.PROCLIB so that the following DD statements point to the datasets resulting from your SMP/E installation: ISPPLIB, ISPMLIB, ISPSLIB, and SYSTSIN.
2. Update member GDDRMAIN in *hlq*.GDDR*vrm*.PROCLIB so that the GDDREXEC DD statement points to the *hlq*.GDDR*vrm*.REXX dataset resulting from your SMP/E installation, and the GDDRPARM DD points to the GDDRMAIN parameter dataset (allocated in "Define GDDRPARM file" on page 140).

   If you run multiple GDDR-plexes on the same C-systems, create an additional copy of the GDDRMAIN JCL, with a member name of your choice. (It is recommended to replace "GDDR" in the member name with the subsystem name chosen for the affected GDDR-plex.) In each such copy, add a GDD$*nnnn* DD DUMMY card in the JCL, where *nnnn* is a unique 1-4 character subsystem name for this GDDR-plex. The subsystem name on the GDDRPARM statements for this GDDRMAIN instance must match the name specified in the GDDRMAIN JCL.
3. Make the GDDR started procedures available to the C-systems by copying members GDDRMAIN, GDDRWORK, and GDDREVM from *hlq*.GDDR*vrm*.PROCLIB to SYS1.PROCLIB or equivalent library for started tasks.

   If you run multiple GDDR-plexes on the same C-systems, create additional copies of the GDDRWORK and GDDREVM JCL, replacing "GDDR" in the member name with the subsystem name chosen for the affected GDDR-plex.
4. Copy member GDDREVMP from *hlq*.GDDR*vrm*.PARMLIB to the parameter library referenced in the GDDREVM started procedure member.
5. Make the GDDRMAIN started procedure available to the managed systems by copying GDDRMAIN from *hlq*.GDDR*vrm*.PROCLIB to SYS1.PROCLIB or equivalent library for started tasks on the respective system.
6. If you plan to use customized versions of GDDR user exits, include a SYSEXEC DD statement referencing your user RCXFE library in the GDDRPROC procedure member.

   **Note:** "Sample procedure to use interpreted REXX user exits" on page 90 provides more information.

7. Ensure the GDDRMAIN started tasks connect to the appropriate SCF instance.

   The GDDRMAIN started task must be associated with a specific SCF instance. To do this, provide an identical SCF$*nnnn* DD DUMMY statement in the started procedure of the SCF started task and the GDDRMAIN started task, where *nnnn* is a user-supplied value that is unique to the SCF instance and the GDDRMAIN started procedures. The *nnnn* value cannot match any GDDR subsystem name.

   If you run multiple GDDR-plexes on the same C-systems, connect each GDDRMAIN instance to a separate SCF instance.

   Do not add SCF$*nnnn* DD cards in the GDDRWORK or GDDREVM JCL. SCF connection for these dependent address spaces is handled automatically.

   The default procedures supplied for GDDRWORK and GDDRPROC refer to the *hlq*.GDDR*vrm*.ISPPROF library allocated during GDDR installation. Even though GDDR does not use or save any ISPF profile variables, this ISPPROF library is required to be able to run ISPF in batch, as ISPF will not start without a ISPPROF DD statement pointing to a PDS.

## Running GDDR without STEPLIB

You can choose to run all GDDR address spaces from LNKLST or from STEPLIB.

When running multiple GDDR instances on C-systems with different maintenance levels, only one instance can run from LNKLST; all others should run from STEPLIB.

GDDRMAIN loads the following modules into dynamic LPA at startup: GDDRPCS, GDDRPCR, GDDRARR, GDDRXG1A, GDDRSRB1, GDDBENF1, and GDDBENF2.

GDDM218I messages are issued at GDDRMAIN startup indicating where each of these modules is loaded from, for example:

```
GDDM218I Module GDDRXG1A loaded into dynamic LPA from STEPLIB
```

When running multiple instances of GDDR, each instance has its own copy of these modules in dynamic LPA.

GDDR does not require any other modules in LPA.

If any GDDR libraries have been added to LNKLST, LLA (library lookaside) must be refreshed when GDDR maintenance is applied. In addition, if you have decided to put selected GDDR modules in LPA, those modules must also be refreshed. If you run GDDR from LNKLST, outdated copies of GDDR modules in LPA could lead to maintenance mismatch messages from GDDR, potentially preventing GDDR address spaces from initializing. To determine the mismatching module(s), compare the module-level maintenance report (message GDDUG33I) in SYSPRINT between GDDRMAIN and the address space where the GDDM075E (maintenance mismatch) message was issued.

If module GDDRXG1A needs to be manually deleted for whatever reason, ensure that the dynamic exit GDDREXG1GDDR (the last four characters are the GDDR subsystem name) is inactive, as described "Dynamic exits" on page 143.

## Dynamic exits

When GDDRMAIN starts, it creates a dynamic exit called GDDREXG1GDDR (the final four GDDR characters are variable and represent the GDDR subsystem name). LPA module GDDRXG1A is associated with this exit.

If consistency is requested for global variable backup or restore, the exit is activated. If the backup or restore ends abnormally (for example, by being canceled), it is possible that the exit will be left active and may block or complain about global variable updates. If this occurs, the exit can be manually deactivated by issuing the following command:

```
SETPROG EXIT,MODIFY,EXITNAME=GDDREXG1GDDR,MODNAME=GDDRXG1A,STATE=INACTIVE
```

## Running GDDR with JES3

GDDR supports GDDR C-systems running as a standalone JES3 GLOBAL.

Due to the nature of JES3 and its management of multiple systems as part of a JES3 complex, all GDDR scripts and utilities must specify SYSAFF=* on the JOB card. This means that SYSAFF=* must be included in the JOB card specification on the Change GDDR ISPF Profile Variable Values panel (P) as well as on the Script JCL Parameters panel (M,P,O,J) in the GDDR Parameter Wizard. All batch JCL in SAMPLIB contains SYSAFF=* on the sample JOB card.

**Note:** GDDR does not issue VARY commands for tape devices or SMS storage groups in a JES3 environment.

# Install GDDR Licensed Feature Code

To use GDDR, you must install a Licensed Feature Code (LFC). An LFC is a 16-character alphanumeric string that is attached to a product or a feature within a product. The LFC is provided on the Licensed Feature Authorization Card included with GDDR.

The LFC is maintained and verified in SCF (Symmetrix Control Facility, part of ResourcePak Base, which is a component of Mainframe Enablers).

If SCF is not active on GDDRMAIN startup, GDDRMAIN waits up to 40 minutes for SCF to start and initialize. If after 40 minutes SCF has not initialized, GDDRMAIN terminates with an error message. When SCF is initialized, GDDRMAIN performs a check to ensure the local system has a valid GDDR license in the SCF initialization file. If neither license is detected, GDDRMAIN terminates with an error message.

- Enter the GDDR LFC in the SCF initialization file as described in the *ResourcePak Base for z/OS Product Guide*. For additional support, email licensekeys@emc.com.

# Customize GDDRMAIN parameters

The GDDRMAIN parameters define the global variable data in virtual (DIV) dataset names, configured sites and regions, CPCs, C-system names, storage systems, MSC group names, communication settings, message automation rules, and also other settings and values.

These parameters are defined in the GDDRPARM file (allocated as described in "Define GDDRPARM file" on page 140) and propagated to be available to C-systems and managed systems during GDDR installation.

**Note:** "GDDRPARM statements" on page 256 describes the GDDRPARM statements.

GDDRMAIN parameters must be maintained as the following environment changes occur:

- IP address and IP port changes (COMM parameters)
- Adding or removing storage systems (SYMM parameters)
- Adding or removing a C-system or managed system (CSYSSITE and CPC parameters)

**Note:** Chapter 10, "Maintaining GDDR Environment," describes the maintenance procedures.

Sample GDDRPARM statements for GDDR can be found in the GDDMPARM member in SAMPLIB.

Complete the following tasks to customize GDDRMAIN parameters:

- Install GDDRPARM file
- Verify GDDRPARM file consistency

## Install GDDRPARM file

You must install the GDDRPARM file and run GDDRMAIN before attempting to use the GDDR ISPF interface.

In general, managed systems do not have DASD which is shared by all systems. Some systems may have common DASD (such as production systems at one site), but none are required to. In a typical installation, C-systems may not share DASD with managed systems within a site. Therefore, you will maintain multiple copies of the GDDRPARM file.

To define and maintain systems so that they all reference consistent copies of the GDDRPARM file:

1. Create a member for the GDDRPARM file in the designated parameter library on one system by editing member GDDMPARM in SAMPLIB. Include the following definitions:
   - A SITE statement to define each site in the configuration, as described in "SITE" on page 272.
   - A DIV dataset definition for each C-system (GVDIVDSN)

The DIV dataset names are conveyed to GDDRMAIN via the GVDIVDSN statement described in "GVDIVDSN" on page 265.

**Note:** If you need to change the DIV dataset, see the instructions in "Changing global variable DIV dataset or WORKER parameters" on page 588.

- A definition for each C-system (CSYSSITE)

  CSYSSITE statements define the system names of the C-systems and the site where they are located, as described in "CSYSSITE" on page 263.

- An IP communications definition for each system (COMM)

  COMM statements specify the IP addresses and ports for GDDRMAIN to use for GDDR inter-system communication, as described in "COMM" on page 259.

  **Note:** If you need to change any existing COMM statements, see the instructions in "Changing C-system or managed system IP port" on page 578.

  Communication with an LPAR that is participating in a disaster recovery (DR) test is done using the DRTCOMM statement described in "DRTCOMM" on page 264.

- SYMM parameters for all GDDR-managed storage systems

  SYMM parameters define storage systems to GDDRMAIN, as described in "SYMM" on page 273.

- MSCGROUP parameters to identify MSC group names for each site pair in the configuration, as described in "MSCGROUP" on page 266.

- Optionally: CPC parameters for each CPC where C-systems or managed systems may run

  A CPC parameter specifies the fully qualified CPC name in the format *netid.nau* for GDDR-managed systems, and provides the 'NETID' literal associated with each C-system, as described in "CPC" on page 261.

- Optionally: any WORKER statements needed

  WORKER parameters specify the names of the worker tasks and the minimum and maximum number of worker tasks per system, as described in "WORKER" on page 276.

  **Note:** If you need to change any worker parameters, see the instructions in "Changing global variable DIV dataset or WORKER parameters" on page 588.

- Optionally: any MSG statements needed

  Specify your message status preference for selected message by adding a MSG ENABLE|DISABLE statement to enable or disable the messages, as described in "MSG ENABLE|DISABLE" on page 267.

You can define message automation rules by adding a MSG ADD|MODIFY statement. In this statement, you can either define a message ID for GDDRMAIN to intercept and process or you can define an additional action for a message that GDDRMAIN already intercepts.

**Note:** See "MSG ADD|MODIFY" on page 268 for detailed information about the MSG ADD|MODIFY statements.

For example, you could add a new message that forwards the message to other systems in the GDDR sysplex for processing, or runs a specified REXX routine. Or you could modify an existing GDDR-defined message to specify a REXX routine to be executed after all GDDR-defined actions have been taken for the message.

This capability is referred to as User Message Automation (UMA), and if used, the GDDWCX worker will be started in a GDDRWORK address space. This worker processes all user-defined actions. The normal console message worker (GDDWCM) runs as a subtask in the GDDRMAIN address space. Using a worker in a GDDRWORK address space protects GDDRMAIN from the user-defined REXX routine and any modules it might execute.

2. When the initial GDDRPARM file is ready, start GDDRMAIN on the system where you created the GDDRPARM file.
3. Check for any parameter error messages at startup. If there are any messages, stop GDDRMAIN and correct the GDDRPARM file definitions.
4. Once GDDRMAIN starts without problem, issue the F GDDRMAIN,COMM command and check the displayed list of systems and IP addresses and ports to ensure the list is complete and all addresses are correct.
5. When the COMM command output is correct, propagate the GDDRPARM file to all the other GDDR systems (C-systems and managed systems).

   If you have systems which have common DASD, you do not have to propagate the dataset to each system, but each system must have a copy that is accessible. It is recommended to use an FTP job or exec to accomplish this, and keep the job or exec for future use.
6. Start GDDRMAIN on all the other GDDR systems.
7. Check for parameter error messages at startup. If there are error messages, stop GDDRMAIN on all systems and go back to step 2.
8. Once all GDDRMAIN instances start without error, verify the GDDRPARM file consistency as described in "Check GDDRPARM file consistency with MPARM command" on page 148 and correct any discrepancies as described in "Correct GDDRPARM file inconsistencies" on page 149.

# Verify GDDRPARM file consistency

IMPORTANT

GDDRPARM file contents are crucial to GDDRMAIN operation. If systems have different values for system names or IP addresses, communication between GDDRMAIN instances may be impossible.

## Automatic GDDRPARM file consistency checks

GDDRMAIN takes steps to ensure that the GDDRPARM file in use by all systems is identical:

- GDDRMAIN checks the consistency of the GDDRPARM file data in use every time the COMM subtask is started.
- GDDRMAIN also checks for consistency every eight hours and checks that the GDDRPARM file data matches the data in use by the COMM subtask. If the data does not match, message GDDM144W is issued and the checks are made every hour until they are found to be equal, at which time the interval reverts to eight hours.

  **Note:** Message GDDM144W may not represent a problem if you are in the process of propagating an updated GDDRPARM file, but have not yet activated it by restarting COMM.

To correct any GDDRPARM file inconsistencies, follow the instructions in "Correct GDDRPARM file inconsistencies" on page 149.

## Check GDDRPARM file consistency with MPARM command

You can perform a GDDRPARM file consistency check at any time by using the F GDDRMAIN,MPARM command described in "MPARM" on page 216.

- Issue F GDDRMAIN,MPARM on any system.

  **Note:** You only need to execute the MPARM command on one C-system or managed system. If the system list is complete and all systems respond, the output would be the same on every system.

  The following example shows MPARM command output:

```
GDDM140I GDDRPARM Status
 Sys PRD1     : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
 Sys SYS3     : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
 Sys SYS2     : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
 Sys SYS1     : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
```

  **Note:** In the example, all GDDRMAIN instances are up and are consistent.

- Check the command output to ensure that a valid data line is shown for each system (no occurrences of `Unable to communicate`). On each line, ensure that the values shown for `In-use` and `Dataset` are the same. Check that those values are the same on each line for every system.

- If any values are different, or if any show `Unable to communicate`, you have not propagated the GDDRPARM file correctly to the systems with different values. Copy the original GDDRPARM file to those systems and restart the GDDRMAIN COMM subtask on those systems. After doing this, repeat the verification procedure.

## Correct GDDRPARM file inconsistencies

If a discrepancy is found during GDDRPARM file consistency check, message GDDM141E is issued and Degraded mode is set.

The warning for inconsistent GDDRPARM file data indicated by message GDDM141E is:

```
GDDM141E GDDRPARM Inconsistency detected, System 001E is RESPONSE=inconsistent
```

If this occurs, complete the following steps:

1. Issue the F GDDRMAIN,MPARM command to identify systems that use inconsistent GDDRPARM file data.
2. Correct the GDDRPARM file data on those systems, and restart the COMM subtask on those systems.
3. If this action does not solve the problem, issue F GDDRMAIN,MPARM,CHECK from any system.

   The MPARM,CHECK command retrieves the in-use data from each system and compares them all. If all are equal, consistency is restored and Degraded mode is turned off.

   The MPARM,CHECK command will tolerate no response from production systems, but must receive a response from each C-system in order to declare consistency. The lack of response from any system, in and of itself, does not lead to global inconsistency or Degraded mode. If any system (C-system or managed system) responds with a different in-use value, consistency is not declared and error messages are issued.

# Customize PROCLIB member GDDRPROC

Customize member GDDRPROC in *hlq*.GDDR*vrm*.PROCLIB used to run GDDR scripts to your environment.

1. Update the STEPLIB DD statement to include the following load libraries:
   - *hlq*.GDDR*vrm*.LINKLIB resulting from your GDDR SMP/E installation
   - Your Mainframe Enablers load library
2. Ensure the following DD statements are pointed to the GDDR datasets resulting from your GDDR SMP/E installation:
   - ISPPLIB
   - ISPMLIB
   - ISPSLIB
3. If you plan to use customized versions of GDDR user exits, see "Sample procedure to use interpreted REXX user exits" on page 90.
4. Add GDD$*nnnn* and SCF$*nnnn* DD connector statements to match GDDRMAIN.

   **Note:** "GDDR and SCF connectors" on page 169 describe the GDD$*nnnn* and SCF$*nnnn* DD statements.

# Customize GDDR ISPF interface invocation REXX exec

Customize the GDDR ISPF interface invocation REXX exec for your installation. A sample REXX exec is supplied in *hlq*.GDDR*vrm*.SAMPLIB, member GDDREXC.

- Set `gddrpfx` to the initial qualifiers of your GDDR installation dataset names.

  An extract from the sample GDDREXC member is shown below:

```
/*****************************************************************/
/* Initialize custom variables */
/*****************************************************************/
gddrpfx = "EMC.GDDR520."
```

## Call GDDR ISPF interface from ISPF selection panel

To start the GDDR ISPF interface by entering an option character from the ISPF primary options menu panel (ISR@PRIM) or from any other ISPF selection menu panel, add an option in the PROC section for that panel that issues the following SELECT service call:

```
'CMD(GDDREXC) NEWAPPL(GDDR) MODE(FSCR)'
```

## Call GDDR ISPF interface from any ISPF panel with TSO GDDR command

To start the GDDR ISPF interface from *any* ISPF panel, by issuing the command TSO GDDR from any ISPF panel's primary command line, create a new REXX exec by the name ‘GDDR’ with the following contents:

```
/* REXX */
address ISPEXEC "SELECT CMD(GDDREXC) NEWAPPL(GDDR) MODE(FSCR)"
```

Specify the NEWAPPL(GDDR) keyword to ensure that all profile variables are stored properly in the same ISPF application profile.

# Update GDDR ISPF profile

To perform GDDR setup and maintenance actions, populate your ISPF profile dataset with basic information. Complete the steps listed in “Updating personal GDDR ISPF profile” on page 284 to update your personal GDDR ISPF profile.

# Configure GDDR

Use the GDDR ISPF interface described in Chapter 6, "Using GDDR ISPF Interface" to complete the following tasks:

- Perform initial parameter activation
- Customize GDDR parameters

The Parameter Management Options Menu panel (M,P) described in "Manage GDDR parameters (M,P)—GDDR Parameter Wizard" on page 287 presents parameter input panels which conditionally prompt for required parameter values based on your particular disaster restart topography and SRDF configuration. The systems, devices, and GDDR options defined during the parameter input task specify how GDDR handles conditions requiring the use of Dell EMC recovery and business continuity facilities.

The GDDR parameter management process uses an existing backup of parameter values as the starting point for subsequent updates. To facilitate the parameter customization associated with a new GDDR implementation, several initial parameter definitions are populated from the GDDRPARM file statements supplied during installation as described in "Install GDDRPARM file" on page 145.

Subsequent parameter changes will be applied to a copy of your parameters, created as a backup dataset at the time the existing parameters were applied, or any later time.

---

**Note:** You must install the GDDRPARM file and run GDDRMAIN before attempting to perform further GDDR configuration. This is true for initial parameter activation as well as for any later configurations changes.

---

## Perform initial parameter activation

To perform an initial parameter activation:

1. From the Primary Options Menu panel, select option **M,** Setup and Maintenance, then select option **P,** Manage GDDR Parameters.

Because there are no existing parameters, option P displays a Select Parameter Input Dataset panel, as shown in Figure 28:

```
------------------- GDDR - Select Parameter Input Dataset ---- Row 1 to 1 of 1

     Parameter input dataset ===> JABCDE1.GDDR520.PARMS
             Selected Member ===>                Unselect? ===> N  (Y/N)
 Parameter Load work-dataset ===> ____________________________________________
  Parameter Load description ===> ____________________________________________

   Currently activated GDDR Parameter source
   - *** Unknown *** -
   - No description found -

 Select '--NONE--' or choose a different GDDR Parameter Load input dataset.
 Press <F3> to return to the Parameter Management Options Menu
 Line commands: S elect, U nselect, B rowse, E dit
 ---------------------- Parameter Input Member Selection ----------------------
 Act  Member    Date   Time   Userid                Description
 --- -------- -------- ----- -------- ----------------------------------------
  _  --NONE-- 09/05/27 11:04 JABCDE1 GDDR parameter load from scratch
 ****************************** Bottom of data ********************************

 Command ===>
```

**Figure 28** Select Parameter Input Dataset panel during initial parameter activation

2. To begin parameter definition with an initial parameter set, type **S** in the `Act` field for member name `--NONE--`, `GDDR parameter load from scratch`, and press **Enter**.

   The panel prompts you to specify a work dataset and description:

```
Please provide a work-dataset name and a description for this Parameter load.
```

3. Specify the work dataset and description.

   In the Parameter Load work-dataset field, specify the GDDR Parameter Wizard work dataset defined in step "Define parameter management datasets" on page 139. See "Define GDDRPARM file" on page 140 for details.

   The dataset specified in the Parameter Load work-dataset field must be different from the dataset specified in the Parameter input dataset field. This is because the contents of the parameter load work dataset are overwritten when exiting the Select Parameter Input Dataset panel in step 6.

   The dataset specified in the Parameter Load work-dataset field must also be different from the dataset that is defined to GDDR as the *last activated parameter dataset*, as described in "Last activated parameter dataset" on page 139.

4. Press **F3** to exit.

   On exiting from the Select Parameter Input Dataset panel, the work dataset is initialized with PDS members which are associated with each of the Parameter Load function panels. Messages associated with the work dataset initialization are returned to your TSO session, as shown in Figure 29 and Figure 30:

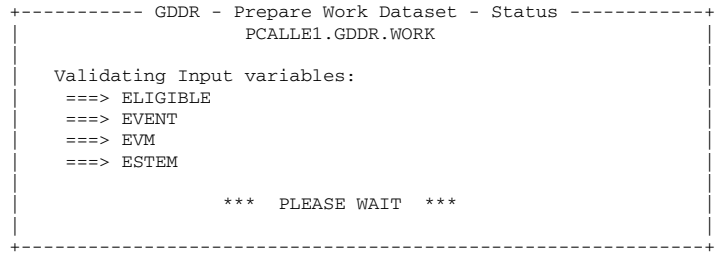



**Figure 29** Prepare Work Dataset - Status panel, input variables

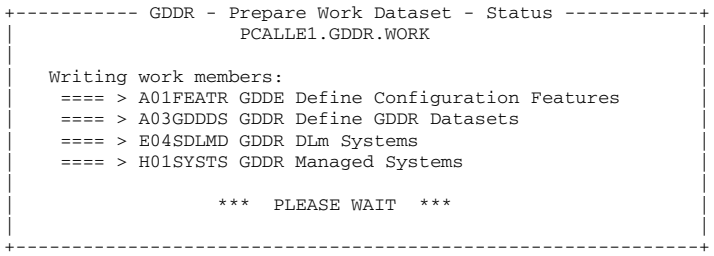



**Figure 30** Prepare Work Dataset - Status panel, work members

5. On selection of the initial parameter backup member and completion of the work dataset initialization process, the Parameter Management Options Menu panel (M,P) is displayed with the complete list of Parameter Load functions.

   **Note:** "Manage GDDR parameters (M,P)—GDDR Parameter Wizard" on page 287 provides a screenshot and explanation of this panel.

   Proceed to step "Customize GDDR parameters" on page 154.

## Customize GDDR parameters

Use the Parameter Load functions in the Parameter Management Options Menu panel to customize the contents of the GDDR Parameter Wizard work dataset with your site-specific values.

This process is identical for an initial parameter customization or for any subsequent customization. "Manage GDDR parameters (M,P)—GDDR Parameter Wizard" on page 287 provides details on this process.

# Configure optional features

Complete the following tasks to configure GDDR optional features:

- Configure GDDR C-System Multi-Tenancy
- Configure GDDR Multi-Tenancy for TimeFinder
- Set SRDF recovery preference
- Configure SRDF/SQAR High Availability (HA)
- Configure DLm support

## Configure GDDR C-System Multi-Tenancy

Complete the following steps to configure GDDR C-System Multi-Tenancy:

- Define GDDR subsystem names
- Customize JCL
- Update automation
- Allocate GDDR gatekeepers
- Customize group and snapshot names
- Customize SRDF Host Component started task
- Customize Mainframe Enablers parameter files
- Customize GDDRPARM file
- Enable multi-tenancy
- Manage other GDDR installation and user datasets

### Define GDDR subsystem names

Assign a unique GDDR subsystem name to each GDDR-plex, as described in "GDDR and SCF connectors" on page 169.

When running multiple C-instances on the same C-system, it is recommended to use the format GDD*x*, where *x* is a number from 1 to 9, starting with GDD1 and incrementing *x* by 1 for each additional C-instance.

GDDR uses the selected GDDR subsystem name as the ISPF application ID. When choosing GDDR subsystem names, ensure that they do not clash with existing ISPF application IDs in use in your environment.

All GDDRMAIN instances within the same GDDR-plex must use the same GDDR subsystem name. This is to ensure global variable names are consistent between systems. If the subsystem name is inconsistent between systems in a GDDR-plex, degraded mode is set.

Ensure that each GDDRMAIN PROC has a unique SCF connector, otherwise the default SCF subsystem name of EMC is used. Likewise, each GDDRMAIN PROC must have a unique GDDR connector, otherwise the default subsystem name of GDDR is used.

## Customize JCL

### GDDR and Mainframe Enablers PROCs

Ensure that each GDDR instance has its own set of GDDR and Mainframe Enablers PROCs as applicable to the type of GDDR presence (C or P) (GDDRMAIN, GDDREVM, GDDRWORK, GDDRPROC, GDDRSCF, GDDRSRDF, GDDRCGRP).

Replace the default subsystem name 'GDDR' with a customized subsystem name as required (GDD1MAIN and so on).

It is recommended that each GDDR instance has its own PROCLIB to contain the customized GDDRPROC procedure.

### TSO logon procs

If different C-instances run different GDDR maintenance levels, use separate TSO logon procs. When accessing the GDDR ISPF interface for a particular C-instance, the GDDR maintenance level of the TSO user's address space must match that of the GDDRMAIN started task for the selected instance. TSO logon procs must not have GDDR or SCF connector DDs.

### Batch jobs

Ensure that all batch jobs interfacing with GDDR (that is, the utilities provided in SAMPLIB) have a GDDR connector matching that of the target GDDR instance.

All batch jobs requiring SCF functionality must also have an SCF connector matching that of the target GDDR instance.

The only exception is if the target GDDR instance is using the default GDDR and/or SCF subsystem names, in which case the connector DD is optional.

Batch jobs using GDDRPROC benefit from the customization done there, and do not need further updates, except for the JCLLIB statement (must specify the PROCLIB containing GDDRPROC for the selected GDDR instance) and any parameters specific to that utility.

### Scripts

To run scripts in parallel across different C-instances on the same z/OS system, specify a unique jobname prefix for each C-instance in the Script JCL Parameters panel (M,P,O,J).

## Update automation

Update automation to use the customized subsystem names.

Global variable names begin with GLOBAL.*subsys*, where *subsys* is the GDDR subsystem name. In practice, this has traditionally meant that all global variable names began with GLOBAL.GDDR. However, if the GDDR subsystem name is GDD1, for example, its global variable names will start with GLOBAL.GDD1.

Likewise, software consoles started by GDDRMAIN are traditionally named GDDRCON*x*, where *x* is a number from 1-9. If the GDDR subsystem name is GDD1, for example, the console names will be GDD1CON*x*.

## Allocate GDDR gatekeepers

It is recommended that each C-instance has its own dedicated GDDR gatekeepers.

As long as GDDR script operations are serialized across C-instances, it may be possible to share GDDR gatekeepers among C-instances, for instance if one test GDDR instance is running alongside one production GDDR instance.

## Customize group and snapshot names

Ensure that each C-instance uses its own dedicated set of Mainframe Enablers group and snapshot names, including the following:

- GNS group names
- Consistency group names
- MSC group names
- Managed VDG group names
- SnapVX snapshot names

If no storage systems are shared between two instances, you can use the same group names or snapshot names, but to keep operations of multiple C-instances manageable, it is still strongly recommended that each instance uses its own names.

## Customize SRDF Host Component started task

Ensure that each C-instance has its own SRDF Host Component subsystem name and command prefix.

---

**Note:** C-instances can share Dell EMC storage systems. However, MSC activation cannot occur simultaneously in two or more instances if they share a storage system. In this case, only one of the activations will succeed; the other(s) will have to be re-driven.

---

## Customize Mainframe Enablers parameter files

Ensure that each C-instance has dedicated SCF, SRDF Host Component, and ConGroup parameter files.This includes the required SIT*xxxxx* memberslisted in "Create SRDF parameter members (SITxxxxx)" on page 119.

## Customize GDDRPARM file

Ensure that a correct GDDR subsystem name is specified on the GDDRPARM file entries (in column 19). While different C-instances can share the same GDDRPARM file, each C-instance only processes GDDRPARM statements matching its GDDR subsystem name.

For each C-instance, specify its own dedicated TCP/IP port, which must be the same across the entire GDDR-plex.

Ensure that each C-instance has its own dedicated DIV dataset.

CPCs can be shared among C-instances. However, managed LPARs cannot be shared. It is recommended that only one C-instance be using the CBU Activate/Undo capability.

### Enable multi-tenancy

To enable multi-tenancy for an active C-instance, add a MULTI ENABLE statement for that subsystem in the GDDRPARM file, and issue the F GDDRMAIN,PARM_REFRESH command (or recycle GDDRMAIN) on all systems in the affected GDDR-plex.

If a C-instance is already active on a particular z/OS system, a subsequent C-instance can only be started if multi-tenancy is enabled for all subsystems.

**Note:** If only one C-instance will ever be active on each z/OS system, the MULTI statement is not required for the subsystem.

### Manage other GDDR installation and user datasets

Different C-instances can run from the same GDDR installation datasets, such as LINKLIB, ISPPLIB, ISPSLIB, ISPMLIB, PARMLIB, REXX, and RCXFE.

It is recommended to have a separate PROCLIB and SAMPLIB for each C-instance so that it can be permanently customized with appropriate GDDR and SCF connectors.

Each C-instance must have its own dedicated Global Variable Backup datasets.

It is recommended to have separate GDDR parameter wizard datasets, such as LAPD (last activated parameter dataset), Global Variable Backup datasets, and the GDDR Parameter Wizard work dataset.

### Set up use of LNKLST or STEPLIB

When running multiple C-instances with different GDDR software versions or maintenance levels on the same C-system, only one version can use LNKLST. All other instances must use STEPLIB for all address spaces in the affinity group.

When using LNKLST, follow the instructions provided in "Customize LINKLIB and REXX parameter files" on page 112.

### Customize SAF profiles

GDDR uses the same SAF profiles for all GDDR instances.

Some of the profiles include procedure names and may need additional customization, for example, to allow each GDDR instance to issue MSC commands against the Mainframe Enablers started tasks in its affinity group.

## Configure GDDR Multi-Tenancy for TimeFinder

GDDR Multi-Tenancy for TimeFinder provides system-level isolation needed when an environment, which is normally managed by GDDR at the site level, is shared by multiple tenants.

**Note:** "GDDR multi-tenancy for TimeFinder" on page 79 discusses this feature.

To implement GDDR Multi-Tenancy for TimeFinder, perform the following steps any time after a normal GDDR installation process, including GDDR Parameter Wizard activation, has completed:

- Allocate BCVGROUP dataset
- Add BCVGDEVS DD to GDDRPROC

## Allocate BCVGROUP dataset

The BCVGROUP dataset defines a group of BCVs by listing SRDF devices subject to GDDR Multi-Tenancy for TimeFinder processing.

The BCVGROUP dataset can be a sequential dataset or PDS or PDS/E member.

Each line in the BCVGROUP dataset should have at least 4 words:

```
SYMMSERIAL# LOWSYMDEV# HIGHSYMDEV# SYSTEMNAME
```

Where:

`SYMMSERIAL#`

12-digit PowerMax/VMAX system serial number (numerical). This serial number must be specified on a SYMM parameter in the GDDRPARM file.

`LOWSYMDEV# HIGHSYMDEV#`

Specify the low and high PowerMax/VMAX device numbers (one to eight character hex values) that define a contiguous range of GDDR-managed SRDF devices.

GDDR automatically determines BCV devices associated with the specified SRDF devices.

`SYSTEMNAME`

A one to eight character name of a system known to GDDR and eligible to perform a test IPL from BCVs.

If the specified SRDF devices are common to multiple systems:

- Specify a list of system names in SYSTEMNAME, or
- Repeat the lines listing common devices as many times as there are systems using these devices.

The layout is not column-sensitive. Words can be separated by blanks, commas (,), or hyphens (-). Put an * in column 1 to insert comment lines. Comment lines are ignored.

To verify the BCVGROUP dataset contents against the GDDR RDF.DEVICES and STDBCV parameters, run the BCVGROUP Validation utility (GDDRBCVG) described in

## Add BCVGDEVS DD to GDDRPROC

In your customized GDDRPROC member, add a BCVGDEVS DD card, and point it to your BCVGROUP dataset, with DISP=SHR. GDDR scripts need read access to this dataset.

## Configure GDDR support for external devices

GDDR support for external devices adds appropriate commands to GDDR scripts to manage SRDF groups outside of the GDDR-managed consistency protection.

"GDDR support for external devices" on page 81 provides more information about this feature.

### Add external devices to GDDR

The addition of external devices to GDDR typically requires changes to SYMM parameters in the GDDRPARM file, as additional gatekeeper devices are usually needed. You must make the updates to the SYMM parameters, as described in "Restoring normal operations after MSC THAW" on page 590, before completing this procedure.

1. Run the GDDRACDD utility to discover all devices to be managed by GDDR, as described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412. Or, define the external devices in the Define SRDF Device Ranges panel (M,P,D,S) and set the EXT (External) field to **YES**.
2. In the Specify Default Script Call Overrides panel (M,P,O,O), set Manage External Devices to **Y**.

## Set SRDF recovery preference

SRDF recovery preference enables you to specify the desired preference for the order that SRDF replication is established following an unplanned swap.

Set the SRDF recovery preference in the Define GDDR Configuration features panel (M,P,C,F) to SYNC or ASYNC:

- SYNC—The GDDRUP31 script first tries to recover SRDF/S before moving on to SRDF/A recovery, given that SRDF/S replication can be recovered based on the state of the environment.
- ASYNC—The GDDRUP31 script recovers SRDF/A only and leaves the SRDF/S recovery to be performed by the GDDRPA0A script.

## Configure SRDF/SQAR High Availability (HA)

SRDF/SQAR HA ensures continuous operation of SRDF/SQAR in the event of loss or removal of the primary MSC server. "SRDF/SQAR High Availability (HA)" on page 74 provides more information about this feature.

- Select the required SRDF/Star SQAR support option in the Define GDDR Configuration features panel (M,P,C,F).

## Configure DLm support

GDDR can manage DLm configurations using the DLMDR control interface, as described in "DLm integration using DLMDR" on page 72.

- Define at least one DLm system for each site to GDDR.

- Configure GDDR for DLm support in the GDDR Parameter Wizard:
  - Indicate DLm is included in GDDR device management.
  - Define the DLm SRDF devices to GDDR.
  - Specify the DLm IP addresses at each site.
  - Set Tape Drive vary online/offline parameters.
  - Set the DLm command timeout value.

# Modify GDDR user exits

This task is optional.

Modify sample user exits provided in the *hlq*.GDDR*vrm*.SAMPLIB distribution library, or write your own, following the guidelines in "GDDR user exits" on page 90.

# CHAPTER 5
# Running GDDRMAIN

This chapter includes the following topics:


- What is GDDRMAIN? ..... 164
- GDDRMAIN sample JCL ..... 169
- GDDRMAIN EXEC parameters ..... 171
- GDDRMAIN console commands ..... 173
- GDDRMAIN startup command sequence ..... 249
- GDDRMAIN validation checks ..... 251
- GDDRMAIN supported configuration types ..... 255
- GDDRPARM statements ..... 256
- GDDR locks ..... 277

# What is GDDRMAIN?

GDDRMAIN is the main GDDR address space. GDDRMAIN controls GDDR global variable management, message interception, and communication between C-systems and managed systems. GDDRMAIN manages remote command processing using GDDRMAIN subtasks.

## Remote command processing

Remote command processing issues commands to managed systems in parallel and asynchronously acknowledges the completion status of each command. When the WTOR associated with a remote command is issued, the command processor waits for a response to the WTOR and also waits for the command to end. If the command ends, the command processor cancels the associated WTOR.

## GDDRMAIN subtasks

GDDRMAIN subtasks are listed in Table 16.

**Table 16** GDDRMAIN subtasks

| GDDRMAIN subtask name | C-systems | Managed systems | Function |
|---|---|---|---|
| CONSOLE | Yes | Yes | Console command processing |
| COMM | Yes | Yes | Inter-system communication |
| GVT | Yes | No | Global variable data management—load from and save to DIV |
| HBM | Yes | No | GDDR Heartbeat Monitor |
| MCSOPER<br>MCSOCART | Yes<br>Yes | Yes<br>Yes | Console message interception<br>Console command response processor |
| MISC<br>MISCBS*nn* | Yes<br>Yes | Yes<br>Yes | Recurring timed functions<br>BCPii console message scanning |
| WORKMGR<br>GDDWST*nn*<br>GDDWDV*nn*<br>GDDWGV*nn*<br>GDDWCO*nn*<br>GDDWCM*nn*<br>GDDWSX*nn* | Yes<br>Yes<br>Yes<br>Yes<br>Yes<br>Yes<br>Yes | Yes<br>Yes<br>No<br>No<br>Yes<br>Yes<br>Yes | Worker task management<br>Status worker<br>Storage device worker<br>Global variable worker<br>Console command worker<br>Console message worker<br>Restricted status worker |

## GDDRMAIN dependent address spaces

GDDRMAIN starts and stops (or cancels) dependent address spaces. The GDDR Event Monitor and GDDRWORK dependent address spaces are described in Table 17.

**Table 17** GDDRMAIN dependent address spaces

| Dependent address space | C-systems | Managed systems | Function |
|---|---|---|---|
| GDDRWORK | | | Worker in external address space |
| GDDWXH*nn* | Yes | No[a] | HMC worker |
| GDDWXQ*nn* | Yes | No | Command queue worker |
| GDDWXR*nn* | Yes | No | REXX worker |
| GDDWCX*nn* | Yes | Yes | User message automation worker |
| GDDREVM | Yes | No | GDDR Event Monitor |

a. GDDWXH can run on a managed system if that system runs BCPii work.

**Note:** Certain scripts perform actions on a C-system other than the master C-system. When script actions are performed on another C-system, the output for those steps appears in the joblog of a GDDRWORK address space on the targeted system.

The GDDR Event Monitor is automatically restarted each night at midnight.

## Worker task management

GDDR includes a number of worker tasks that process work on behalf of GDDR scripts and processes. Some workers run internal to GDDRMAIN as GDDRMAIN subtasks, and some workers run in external GDDRWORK address spaces (replacing 'GDDR' in GDDRWORK with your GDDR subsystem name).

### Worker task names

Table 18 on page 165 lists worker task names and their functions:

**Table 18** Worker task names

| Task name | Description | Running as |
|---|---|---|
| GDDWCM | Console message worker | GDDRMAIN subtask |
| GDDWCO | Console command worker | GDDRMAIN subtask |
| GDDWCX*nn* | User message automation worker | Worker in external address space |
| GDDWDV | Storage device worker | GDDRMAIN subtask |
| GDDWGV | Global variable worker | GDDRMAIN subtask |
| GDDWST | Status worker | GDDRMAIN subtask |
| GDDWSX | Restricted status worker | GDDRMAIN subtask |
| GDDWXH*nn* | HMC worker | Worker in external address space |
| GDDWXQ*nn* | Command queue worker | Worker in external address space |
| GDDWXR*nn* | REXX worker | Worker in external address space |

*Worker task type* or *worker type* is the portion of the worker task name (as listed in Table 18 on page 165) without the *nn* suffix specified. For example, for REXX workers, the worker task type is GDDWXR and the worker task names are GDDWXR00, GDDWXR01, GDDWXR02, and so on.

## Requirements and restrictions

On a production system, only GDDWCM, GDDWCO, GDDWST, GDDWSX, and GDDWCX are allowed. GDDWXH may be allowed if the managed system handles BCPii requests.

C-systems must have at least one instance of each worker task type active, with the exception of the GDDWSX, GDDWXH, or GDDWCX worker. If a worker (other than GDDWSX, GDDWXH, or GDDWCX) is not running on a C-system, Degraded mode is set for that C-system.

**Note:** "Degraded mode" on page 70 discusses Degraded mode.

## MIN and MAX limits

The worker manager (WORKMGR subtask of GDDRMAIN) monitors the backlog of requests to all worker types. When the backlog grows, additional tasks are started up to the current MAX limit. When the backlog has been processed, idle worker tasks are stopped until the current MIN limit is reached.

You can set the minimum and maximum number of worker tasks initially started per system using the GDDRPARM file WORKER statements (described in "WORKER" on page 276). However, it is recommended that you do ***not*** specify WORKER statements in your GDDRPARM file. This maintains the minimum and maximum numbers for all workers at their default values, which should be appropriate for most configurations.

Default MIN and MAX limit values are listed in Table 19 on page 166. These are the recommended settings for most configurations.

**Table 19** Default MIN and MAX limits

| Worker running on... | ...C-system | ...Managed system |
|---|---|---|
| GDDWCM | MIN=1, MAX=2 | MIN=1, MAX=1 |
| GDDWCO | MIN=1, MAX=5 | MIN=1, MAX=1 |
| GDDWCX | MIN=0, MAX=0 | MIN=0, MAX=0 |
| GDDWDV | MIN=1, MAX=25 | |
| GDDWGV | MIN=1, MAX=5 | |
| GDDWST | MIN=1, MAX=4 | MIN=1, MAX=1 |
| GDDWSX | MIN=1, MAX=1 | MIN=1, MAX=1 |
| GDDWXH[a] | MIN=1, MAX=5 | MIN=1, MAX=5 |
| GDDWXQ | MIN=1, MAX=1 | |
| GDDWXR | MIN=1, MAX=5 | |

a. For systems with BCPii

You can dynamically adjust the MIN and MAX limits using the GDDRMAIN WORKER command (described in "WORKER" on page 247).

If the worker manager encounters three unexpected worker task failures within a one minute period, the worker manager sets the MAX limit for that worker type to '0' and sets the GDDRMAIN status to Degraded and issues message GDDM150E. You must take action to resolve the cause of the failing worker task, and then change the MAX value for the task to the recommended non-zero value (using the GDDRMAIN WORKER command).

## Viewing worker status

- Use the D (Detail) line command in the Perform Health Check panel (C) to view the status of GDDR workers on a certain system, as described in "Viewing system details" on page 372. Alternatively, run the GDDRMAIN CHECKUP command described in "CHECKUP" on page 182 with the DETail keyword.
- Run the GDDRMAIN WORKER command (described in "WORKER" on page 247) to view worker status details with possibility to restrict the display to a certain worker type.

## Starting workers

Normally you do not need to take any action to start workers.

It is recommended to have the GDDRMAIN work manager start the default number of workers of each type based on detected system type. For this purpose, do ***not*** specify any WORKER statements in the GDDRPARM file.

When GDDRMAIN has stopped a worker type because of a repeated error condition, restore the MIN and MAX limits for that type back to the default values listed in Table 19 on page 166.

You can start additional workers by raising the MIN limit above its current value.

## Stopping workers

Use the GDDRMAIN STOP command described in "STOP" on page 234 to stop worker tasks normally. The STOP command interrupts the work being handled by the worker(s) to be stopped and eliminates the need to cancel a GDDR script.

If the worker to be stopped is found busy (as indicated by message GDDM242E), you can, under direction of Dell EMC Customer Support, reissue the STOP command with the FORCE parameter specified to interrupt and stop the worker immediately. In this case, the associated script step ends with RC 995 and message GDDR992E is issued. This allows the script to be rerun. If the script step is PowerMax or VMAX-related, it is recommended to investigate the problem before deciding whether to rerun or skip the failed step.

As a last resort, the GDDRMAIN CANCEL command (described in "CANCEL" on page 181) may be used, under direction of Dell EMC Customer Support, to cancel the worker unconditionally. The CANCEL command abruptly terminates the worker using the DETACH or ASDES macro. The GDDRMAIN CANCEL command is only allowed for a given worker if a STOP command with the FORCE parameter has been attempted; otherwise, the CANCEL command fails.

## Disabling workers

You can disable a particular type of workers to intentionally prevent GDDR from taking actions that would normally be performed by the disabled worker type. For example, you might want to keep GDDR running but not reacting to any messages for a period of time.

Disabling a worker type on a C-system results in Degraded mode (with the exception of the GDDWSX, GDDWXH, or GDDWCX workers), and should therefore only be done to address a bigger problem (such as a runaway condition) and under instruction of GDDR Solution Support.

To disable workers of a particular type, set the MAX value for that worker type to 0 (zero) using the GDDRMAIN WORKER command or the GDDRPARM file WORKER statements.

# GDDRMAIN sample JCL

GDDRMAIN sample JCL is as follows:

```
//GDDRMAIN PROC MFEPFX='HLQ.MFEVRM',
//             GDDRPFX='HLQ.GDDRVRM',
//             TCPNAME='TCPIP',
//             TCPDATA='TCPIP.SEZAINST(TCPDATA)',
//              PRM=
//*
//* GDDRMAIN - GDDR main address space (for C-systems and P-systems)
//*
//* Supported EXEC parms (specify on PRM above):
//*
//*  NOEVM    - Do not start event monitor (EVM)
//*  NOHBM    - Do not start heartbeat monitor (HBM)
//*  NOMCS    - Do not start MCSOPER console subtask
//*  TAPEONLY - Restrict licenses to GDDR Tape only
//*  VMAXONLY - Restrict licenses to GDDR (VMAX) only
//*  DEBUG    - Enable debugging in GDDRMAIN
//*  VERBOSE  - Enable verbose messaging in GDDRMAIN
//*  GVB={dd(hhmm)|*(hhmm)|NONE} - Set global variable backup schedule
//*
//* Note: Modify SCF and GDDR subsystem names below for your GDDR
//*       instance.
//*
//GDDRTCPA EXEC PGM=BPXTCAFF,PARM=&TCPNAME
//*
//GDDRMAIN EXEC PGM=GDDRMAIN,DYNAMNBR=20,REGION=0M,TIME=1440,
//             PARM='TCPIP=&TCPNAME,&PRM'
//STEPLIB  DD  DISP=SHR,DSN=&GDDRPFX..LINKLIB
//         DD  DISP=SHR,DSN=&MFEPFX..LINKLIB
//GDDREXEC DD  DISP=SHR,DSN=&GDDRPFX..REXX
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTCPD  DD  DISP=SHR,DSN=&TCPDATA
//GDDRPARM DD  DISP=SHR,DSN=&GDDRPFX..CUSTOM.PARMLIB(GDDMPARM)
//SCF$EMC  DD  DUMMY  <--- Your SCF subsystem name ('EMC' is default)
//GDD$GDDR DD  DUMMY  <--- Your GDDR subsystem name ('GDDR' is default)
//*
//         PEND
```

## GDDR and SCF connectors

If you run multiple GDDR instances on the same C-systems, then each instance must have a unique GDDR subsystem name and a unique SCF subsystem name.

Each GDDRMAIN instance in a GDDR-plex must use the same GDDR subsystem name and the same SCF subsystem name. One GDDR subsystem connects to only one SCF subsystem.

- A *GDDR connector* is the GDD$*nnnn* DD DUMMY statement in the JCL where *nnnn* specifies the GDDR subsystem name. The GDDR subsystem name must be 1 to 4 alphanumeric or national ($, #, or @) characters. The first character must be alphabetic or national ($, #, or @).

  The GDDR connector ties GDDR address spaces together. If no GDDR connector is specified, GDD$GDDR DD DUMMY is assumed. If a GDDR subsystem name is already in use by another GDDR instance active on a C-system, GDDRMAIN fails to start.

If GDDR C-System Multi-Tenancy is not enabled, it is recommended to use the default subsystem name 'GDDR'. Otherwise, see "Define GDDR subsystem names" on page 155.

- An *SCF connector* is the SCF$*nnnn* DD DUMMY statement in the JCL where *nnnn* specifies the SCF subsystem name. The SCF connector ties a C-system GDDR instance to a set of Mainframe Enablers started tasks, and ties the Mainframe Enablers started tasks together.Thus, each C-system GDDR instance has its own dedicated set of Mainframe Enablers started tasks (SCF, SRDF Host Component, ConGroup), all with the same SCF connector.

  If no SCF connector is specified, SCF$EMC DD DUMMY is assumed. If an SCF subsystem name is already in use by another GDDR instance active on a C-system, GDDRMAIN fails to start.

For GDDR address spaces, the GDDR and SCF connectors are specified in the GDDRMAIN JCL, in GDDRPROC, and in certain standalone batch utilities interfacing with GDDR and/or SCF.

For dependent address spaces (GDDREVM, GDDRWORK), the GDDR and SCF connectors are allocated dynamically and must not be specified in the JCL. TSO logon procs must not have GDDR or SCF connector DDs.

MFE address spaces do not need a GDDR connector, they only need an SCF connector.

# GDDRMAIN EXEC parameters

The GDDRMAIN EXEC parameters include:

- DeBuG
- GVB
- NOEVM
- NOHBM
- NOMCS
- TCPIP
- VERBose

The GDDRMAIN EXEC parameters can be specified in any order and any combination, separated by commas.

## DeBuG

Enables GDDRMAIN debugging (written to console and/or SYSPRINT DD).

## GVB

Schedules a regular backup of global variables or to perform a backup on demand.

**Note:** "Creating GDDR parameter backups" on page 291 provides information on backups.

When the scheduled time comes, the backup will be postponed if a script is in progress.

Settings made using the GVB EXEC parameter can be dynamically overridden with the GVB command described in "GVB" on page 206.

### Syntax

```
GVB={dw|*}(hhmm)|NONE}
```

Where:

*{dw|*}(hhmm)*

Schedules the backup at the specified time:

- *dw* is a day of the week (first two letters).
- * specifies every day.
- *hhmm* is a time (hour and minutes, 24 hour clock).

NONE

(**Default**) No backups are scheduled.

### Example

```
PARM='GVB=MO(1200)'
```

## NOEVM

The NOEVM parameter causes GDDRMAIN to not start the GDDR Event Monitor task. This is useful prior to the initial parameter activation or to correct some global variable problem which needs to be resolved prior to starting this task.

## NOHBM

The NOHBM parameter causes GDDRMAIN to not start the GDDR Heartbeat Monitor tasks. This is useful prior to the initial parameter activation or to correct some global variable problem which needs to be resolved prior to starting this task.

## NOMCS

The NOMCS parameter causes GDDRMAIN to not start the MCSOPER console subtask.

When NOMCS is set, GDDR does not react to any messages it would normally react to. In combination with the GDDR automation state set to OFF using the GDDR OFF command (as described in "Changing GDDR automation state" on page 286), this prevents any reaction of GDDR to planned environment changes.

It is recommended to specify NOEVM and NOHBM when NOMCS is specified.

## TCPIP

Sets TCP/IP stack name (stack affinity).

GDDR uses TCP/IP for internal communications between sites, as well as for DLm support. The TCPIP parameter specifies which TCP/IP address space to use.

### Syntax

`TCPIP=`*`stackname`*

Where:

*`stackname`*

The name of the TCP/IP address space. The default value is TCPIP.

## VERBose

Enables GDDRMAIN verbose messaging.

# GDDRMAIN console commands

GDDRMAIN accepts the z/OS modify command (F) and stop command (P).

## Stop command (P)

To stop GDDRMAIN, issue a P GDDRMAIN command (or use the current jobname, if different).

**⚠CAUTION**

**Do not cancel GDDRMAIN except in an emergency. Although GDDRMAIN has a TERM=YES ESTAE, actions by the ESTAE exit routine are limited when a cancel is used.**

When a P command is issued, GDDRMAIN will stop all subtasks except GVT, then stop GVT. If GVT is not running or if it is unable to write the DIV for any reason, a WTOR will be issued to allow saving the data space if that is desired (see "GDDR locks" on page 277, the DSPSAVE parameter).

## Modify commands (F) summary

GDDRMAIN accepts z/OS console commands in the form "F GDDRMAIN,*<verb-operand>*", where *<verb-operand>* is one of the commands listed in Table 20 through Table 26.

**Table 20** Subtask management commands

| Command | Purpose | Syntax | Page |
|---|---|---|---|
| TASKS | View subtasks | `F GDDRMAIN,TASKS` | 242 |
| START | Start subtasks | `F GDDRMAIN,START <subtask-name>` | 234 |
| STOP | Stop subtasks | `F GDDRMAIN,STOP <subtask-name> [,FORCE]` | 234 |
| RESTART | Restart subtasks | `F GDDRMAIN,RESTART <subtask-name>` | 224 |
| CANCEL | Cancel subtasks | `F GDDRMAIN,CANCEL <subtask-name>` | 181 |

Table 21 lists miscellaneous GDDRMAIN console commands.

**Table 21** Miscellaneous GDDRMAIN console commands (1 of 2)

| Command | Usage | Syntax | Page |
|---|---|---|---|
| BC | Broadcast GDDRMAIN console command | `F GDDRMAIN,BC,<to>,<cmd>` | 179 |
| BR | Broadcast GDDRMAIN console command with response | `F GDDRMAIN,BC,<to>,<cmd>` | 179 |
| BCPII<br>(alias: GIAM) | View status of GDDR IPL Assist Monitor<br>Enable or disable GDDR IPL Assist Monitor<br>Enable or disable debugging for GDDR IPL Assist Monitor | `F GDDRMAIN,BCPII`<br>`F GDDRMAIN,BCPII,{ON\|OFF}`<br>`F GDDRMAIN,BCPII,{DBGON\|DBGOFF}`<br>**Note:** You can specify 'GIAM' instead of the BCPII keyword. | - |
| COMM | View communication status | `F GDDRMAIN,COMM` | 193 |
| DLM | View DLm status<br>Set/reset DLm abort flag<br>Set active control station (for stateless DLm) | `F GDDRMAIN,DLM`<br>`F GDDRMAIN,DLM,{SET\|RESET},ABORT`<br>`F GDDRMAIN,DLM,{CS0\|CS1}` | 198 |
| FIXPERT | View status of the Resiliency Expert<br>Enable or disable the Resiliency Expert | `F GDDRMAIN,FIXPERT`<br>`F GDDRMAIN,FIXPERT,{ON\|OFF}` | 205 |
| GVB | View date and time of scheduled global variable backup<br>Administer global variable backup or adjust schedule | `F GDDRMAIN,GVB`<br>`F GDDRMAIN,GVB=NONE\|NOW`<br>`F GDDRMAIN,GVB={dw\|*}(hhmm)` | 206 |
| LOCK | View status of locks<br>Administer locks | `F GDDRMAIN,LOCK[,<lock-name>]`<br>`F GDDRMAIN,LOCK,<lock-name>,<cmd>` | 210 |
| MPARM | View MPARM checksums for each system<br>Force MPARM resynchronization across all systems | `F GDDRMAIN,MPARM`<br>`F GDDRMAIN,MPARM,CHECK` | 216 |
| MSGS | View message interception status<br>Enable or disable message interception for specified message | `F GDDRMAIN,MSGS[,<msgid>]`<br>`F GDDRMAIN,MSGS,<msgid>,{ENABLE\|DISABLE}` | 217 |
| RDFREFR | Refresh SRDF link topology map | `F GDDRMAIN,RDFREFR` | 219 |

**Table 21** Miscellaneous GDDRMAIN console commands (2 of 2)

| Command | Usage | Syntax | Page |
|---|---|---|---|
| SCRIPT | Execute GDDR script | F GDDRMAIN,SCRIPT,<*script-name*>,<br>[,1SIDE={Y\|N}]<br>[,ALLOWHMC={Y\|N}{Y\|N}{Y\|N}{Y\|N}]<br>(PRIMARY, SECONDARY, WTOR,<br>LOADCLEAR)<br>[,ALTPARMS={Y\|N}]<br>[,AUTO]<br>[,BCVGROUPS={Y\|N}]<br>[,BGCLONE={Y\|N}]<br>[,CALLOVER=<*call-overrides*>]<br>[,DRTPARMS={Y\|N}]<br>[,EXTBCV={Y\|N}]<br>[,EXTDEV={Y\|N}]<br>[,LOAD=<*script-parm-file*>]<br>[,PRISWAP=<*primary-swap-group*>]<br>[,RRSKIP={Y\|N}]<br>[,SDDFCLN={Y\|N}]<br>[,SECSWAP=<*secondary-swap-group*>]<br>[,SELSYS=<*lpar*>\|(<*lpar*>,<*lpar*>,...)]<br>[,SITE=DC*n*]<br>[,SUBMIT={Y\|N}]<br>[,SYSTEM=<*system-name*>]<br>[,TESTBCV1={Y\|N}]<br>[,TESTBCV2={Y\|N}]<br>[,TESTBCV3={Y\|N}]<br>[,TESTBCV4={Y\|N}]<br>[,USER=<*userid*>]<br>[,ZDPPAUSE={Y\|N}]<br>[,ZDPRESUME={Y\|N}]<br>[,ZDPUPLA={Y\|N}]<br>[,ZDPUPLB={Y\|N}]<br>[,ZDPUPLC={Y\|N}]<br>[,ZDPUPLD={Y\|N}] | 225 |
| UMA | View status of user message automation<br>Enable or disable user message automation | F GDDRMAIN,UMA<br>F GDDRMAIN,UMA,{ON\|OFF\|RELOAD} | 246 |
| WORKER | View workers<br>Adjust minimum or maximum number of worker tasks<br>Disable worker type | F GDDRMAIN,WORKER[,<*worker-name*>]<br>F GDDRMAIN,WORKER,<*worker-name*>,<br>{MIN=<*min*>\|MAX=<*max*>} | 247 |

## GDDRMCMD commands

Table 22-Table 26 list GDDRMCMD commands that can be issued as GDDRMAIN console commands.

Table 26 lists GDDRMCMD commands used to view configuration information.

**Table 22** GDDRMCMD commands: configuration information

| Command | Purpose | Syntax | Page |
|---|---|---|---|
| CONFIG | View and validate configuration of GDDR-managed SRDF groups | `F GDDRMAIN,ConFiG [,CONsole[={Yes\|No}]] [,ConTRoLler=<serial#>] [,DeBuG[={Yes\|No}]] [,SITE={DC1\|DC2\|DC3\|DC4}] [,SysMSG[={Yes\|No}]] [,SysPRinT[={Yes\|No}]]` | 194 |
| GATEK | View and validate paths to GDDR-managed storage systems | `F GDDRMAIN,GateK [,SITE={DC1\|DC2\|DC3\|DC4}] [,ConTRoLler=<serial#>] [,Auto_UnBoX[={Yes\|No}]] [,PRompt[={Yes\|No}]] [,SIMulate[={Yes\|No}]] [,FoRCe[={Yes\|No}]] [,ALL[={Yes\|No}]] [,CONsole[={Yes\|No}]] [,SysMSG[={Yes\|No}]] [,SysPRinT[={Yes\|No}]] [,DeBuG[={Yes\|No}]]` | 201 |
| TOPOLOGY | View SRDF topology | `F GDDRMAIN,TOPOlogy [,SITE={DC1\|DC2\|DC3\|DC4}] [,ConTRoLler=<serial#>] [,ALL[={Yes\|No}]] [,CONsole[={Yes\|No}]] [,SysMSG[={Yes\|No}]] [,SysPRinT[={Yes\|No}]] [,DeBuG[={Yes\|No}]]` | 243 |

Table 23 lists GDDRMCMD commands used to display GDDR-plex information.

**Table 23** GDDRMCMD commands: GDDR-plex information

| Command | Purpose | Syntax | Page |
|---|---|---|---|
| CHECKUP | View events and command queue; view and validate system status information across all systems in the GDDR-plex where GDDRMAIN is active | F GDDRMAIN,CHecKup<br>[,CMDQueue]<br>[,EVenTs]<br>[,SYStem[=*system-name*]]<br>[,SITE={DC1\|DC2\|DC3\|DC4}]<br>[,TiMeOut=*seconds*]<br>[,SUMmary[={Yes\|No}]]<br>[,DETail[={Yes\|No}]]<br>[,CONsole[={Yes\|No}]]<br>[,SysMSG[={Yes\|No}]]<br>[,SysPRinT[={Yes\|No}]]<br>[,DeBuG[={Yes\|No}]]<br>[,VERBose[={Yes\|No}]] | 182 |
| LICENSE | View and validate GDDR license information across all systems in the GDDR-plex where GDDRMAIN is active | F GDDRMAIN,LICense<br>[,SITE={DC1\|DC2\|DC3\|DC4}]<br>[,CONsole[={Yes\|No}]]<br>[,SysMSG[={Yes\|No}]]<br>[,SysPRinT[={Yes\|No}]]<br>[,DeBuG[={Yes\|No}]] | 207 |
| SYSTEMS | View all systems (both C-systems and production systems) defined to GDDR and their status | F GDDRMAIN,SYStems<br>[,SITE={DC1\|DC2\|DC3\|DC4}]<br>[,CONsole[={Yes\|No}]]<br>[,SysMSG[={Yes\|No}]]<br>[,SysPRinT[={Yes\|No}]]<br>[,SYStem=<*system-name*>]<br>[,DeBuG[={Yes\|No}]] | 240 |

Table 24 lists GDDRMCMD commands used to view local system information.

**Table 24** GDDRMCMD commands: local system information

| Command | Purpose | Syntax | Page |
|---|---|---|---|
| ENQ | View GDDR ENQs | F GDDRMAIN,ENQ<br>[,CONsole[={Yes\|No}]]<br>[,SysMSG[={Yes\|No}]]<br>[,SysPRinT[={Yes\|No}]]<br>[,DeBuG[={Yes\|No}]] | 199 |
| SUBSYS | View active and inactive GDDR subsystems | F GDDRMAIN,SubSys<br>[,ACTive[={Yes\|No}]]<br>[,CONsole[={Yes\|No}]]<br>[,SysMSG[={Yes\|No}]]<br>[,SysPRinT[={Yes\|No}]]<br>[,DeBuG[={Yes\|No}]] | 235 |
| SUMMARY | View summary information for system where command is run | F GDDRMAIN,SUMmary<br>[,CONsole[={Yes\|No}]]<br>[,SysMSG[={Yes\|No}]]<br>[,SysPRinT[={Yes\|No}]]<br>[,DeBuG[={Yes\|No}]] | 237 |

Table 26 lists GDDRMCMD commands used to view address space information.

**Table 25** GDDRMCMD commands: address space information

| Command | Purpose | Syntax | Page |
|---|---|---|---|
| MAINTENANCE | View GDDR maintenance information | `F GDDRMAIN,MAINTenance`<br>`[,SUMmary[={Yes\|No}]]`<br>`[,COMposite[={Yes\|No}]]`<br>`[,DETail[={Yes\|No}]]`<br>`[,LoadMOD=<module>]`<br>`[,CONsole[={Yes\|No}]]`<br>`[,SysMSG[={Yes\|No}]]`<br>`[,SysPRinT[={Yes\|No}]]`<br>`[,DeBuG[={Yes\|No}]]` | 211 |
| REGION | View above and below the line REGION limits and allocation information | `F GDDRMAIN,ReGioN`<br>`[,CONsole[={Yes\|No}]]`<br>`[,SysMSG[={Yes\|No}]]`<br>`[,SysPRinT[={Yes\|No}]]`<br>`[,DeBuG[={Yes\|No}]]` | 220 |

Table 26 lists utility GDDRMCMD commands.

**Table 26** Utility GDDRMCMD commands

| Command | Purpose | Syntax | Page |
|---|---|---|---|
| PARM_REFRESH | Reread the GDDRPARM file (MPARM) and perform various processing | `F GDDRMAIN,PaRM_REFresh`<br>`[,ECHO[={Yes\|No}]]`<br>`[,CONsole[={Yes\|No}]]`<br>`[,SysMSG[={Yes\|No}]]`<br>`[,SysPRinT[={Yes\|No}]]`<br>`[,DeBuG[={Yes\|No}]]` | 218 |
| RELOAD | Reload GDDRMAIN interface modules into dynamic LPA | `F GDDRMAIN,ReLoad`<br>`[,FoRCe[={Yes\|No}]]`<br>`[,CONsole[={Yes\|No}]]`<br>`[,SysMSG[={Yes\|No}]]`<br>`[,SysPRinT[={Yes\|No}]]`<br>`[,DeBuG[={Yes\|No}]]` | 222 |
| SET | Set GDDRMAIN debugging, tracing, and verbose messaging options | `F GDDRMAIN,seT`<br>`[,DeBuG[={ON\|OFF}]]`<br>`[,TRaCe[={ON\|OFF}]]`<br>`[,VERBose[={ON\|OFF}]]`<br>`[,CONsole[={Yes\|No}]]`<br>`[,SysMSG[={Yes\|No}]]`<br>`[,SysPRinT[={Yes\|No}]]` | 232 |
| SVCDUMP | Produce SVC dump including GDDR data spaces | `F GDDRMAIN,SVCDump` | 239 |
| TRACE RESET | Reset GDDR trace buffer | `F GDDRMAIN,TRaCe RESet`<br>`[,CONsole[={Yes\|No}]]`<br>`[,SysMSG[={Yes\|No}]]`<br>`[,SysPRinT[={Yes\|No}]]`<br>`[,DeBuG[={Yes\|No}]]` | 245 |

# BC and BR

GDDRMAIN allows you to broadcast an operator command from any managed system (where GDDRMAIN is running and in communication with other instances of GDDRMAIN) to any, all, or selected GDDR systems.

- BC (Broadcast) broadcasts the operator command but does not return any output of the executed command. It also does not indicate if the system actually received or processed the command. The BC command provides a message for each system to which it sends the command.
- BR (Broadcast and Response) broadcasts the operator command and returns the output of the executed command.

If BR is used to issue an F GDDRMAIN command to systems which include the system on which the BR command was issued, there is no response shown from the system on which the BR was issued. This is because the command processor on that system is busy with the BR command and will not execute the broadcast command until the BR completes. This has no effect because the command will be executed immediately after the broadcast processing, so the command output will be seen when it executes on the host system.

## Syntax

```
F GDDRMAIN,{BC|BR},<to>,<cmd>
```

## Parameters

*to*

Command destination, which can be one of the following:

*system-name* — the name of the system where the command is to be executed

*site* — the name of he site where command is to be executed

* — send to all managed systems (including the local system)

*C — send to all C-systems

*P — send to all production systems

*1 — send to all systems, but send to local system first

*Z — send to all systems, but send to local system last

*X — send to all systems except local system

*cmd*

The command to be sent to the indicated systems.

## Example

*Example 1* BC,*,D T sends a D T command to every GDDR system:

```
GDDM145I Command sent to system X19
GDDM145I Command sent to system X118
GDDM145I Command sent to system X99
GDDM145I Command sent to system X117
GDDM149I Issuing D T for system X117 Job JABCDEX (J0003662)
D T
IEE136I LOCAL: TIME=11.26.40 DATE=2010.008  UTC: TIME=16.26.40 DATE=2010.008
```

The GDDM145I messages show which systems were sent the command. The GDDM149I message is displayed on every system which receives the command to identify the source of the broadcast.

*Example 2* In case of an IP port change, COMM restart can be handled via the following command:

```
F GDDRMAIN,BC,*Z,F GDDRMAIN,RESTART COMM
```

(GDDRMAIN is the name of the GDDRMAIN address space on all systems.)

---

**Note:** *Z is required, because the command cannot be broadcast from the system where it is issued if that system is already using the new ports and the other systems are not.

---

*Example 3* When adding a new system, COMM restart can be handled by issuing the following command from the new system:

```
F GDDRMAIN,BC,*X,F GDDRMAIN,RESTART,COMM
```

*Example 4* When deleting a system, COMM restart can be handled by issuing the following command:

```
F GDDRMAIN,BC,*,F GDDRMAIN,RESTART COMM
```

Note that *1 or *Z could be used here as well.

*Example 5* To start GDDR Event Monitor and GDDR Heartbeat Monitor:

```
/F GDDRMAIN,BC,*C,START EVM
/F GDDRMAIN,BC,*C,START HBM
```

# CANCEL

Use CANCEL to unconditionally stop one or more running GDDRMAIN subtasks.

**Note:** "Stopping workers" on page 167 provides directions on how to stop GDDR workers.

**WARNING**

**Use under direction of Dell EMC Customer Support only.**

For GDDRMAIN internal subtasks, CANCEL requires a previous STOP command.

For GDDR workers (including those running as GDDRMAIN internal subtasks), the CANCEL command is only allowed for a given worker if a STOP command with the FORCE parameter has been attempted for the same worker; otherwise, the CANCEL command fails.

## Syntax

```
F GDDRMAIN,CANCEL <subtask-name>
```

## Parameters

*subtask-name*

Specify the subtask name. Issue the TASKS command to see the names of the subtasks.

For GDDR workers whose task names are appended with numbers (*nn*), you can specify a worker type instead of a specific worker name to stop all workers of that type. For example, specify **GDDWXR** to stop all REXX workers (GDDWXR00, GDDWXR01, GDDWXR02, and so on), or **GDDWXR02** to stop REXX worker GDDWXR02 only.

## CHECKUP

The CHECKUP command shows event, command queue, and system status information and validates the GDDR licenses, configuration type, subsystem name, GDDR version, maintenance level, and compatibility level across all systems in the GDDR-plex where GDDRMAIN is active.

Event information includes the following:

- Active and expected events
- Whether Degraded mode is on or off
- Whether the GDDRPARM file is consistent across all systems
- DLm abort status

Command queue information includes the following:

- Counts and listing of commands on the queue
- Owner of command queue ENQ
- Maximum number of active tasks

System information includes the following for each system in the GDDR-plex:

- Site, system name and type (control or production system, master or not), subsystem name
- IP address and port, system status, last communication date and time
- GDDR version and maintenance level, compatibility level
- GDDR licenses
- Configuration type and whether potential problems exist (C-systems only)

  **Note:** An asterisk after the configuration type indicates a potential problem with the configuration. Issue the CONFIG command on that system for more information. If the CONFIG command shows that there is no longer a problem, the asterisk will be cleared from the configuration type.

- System-level events

**Note:** Subsystem name, GDDR version and maintenance level, compatibility level, licenses and configuration type are displayed only if GDDRMAIN is active. These values are validated across all systems in the GDDR-plex where GDDRMAIN is active.

By default, all information is displayed. You can limit the output using the CMDQUEUE, EVENTS, and SYSTEM options.

Requirements and restrictions are as follows:

- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.
- CMDQUEUE and EVENTS options are supported only on C-systems.

The CHECKUP command is issued automatically on GDDRMAIN startup. If a subsystem name, license, version, or compatibility level mismatch is detected, a WTOR is issued. If the local system is in error, the operator can reply CANcel to terminate GDDRMAIN and resolve the mismatch before restarting. If a remote system is in error, the operator can resolve the mismatch on that system and then reply Retry to redrive cross-system validation after the issue has been resolved. Replying CONTinue to proceed with GDDRMAIN initialization is not recommended.

The CHECKUP command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,CHecKup
[,CMDQueue]
[,EVenTs]
[,SYStem[=system_name]]
[,SITE={DC1|DC2|DC3|DC4}]
[,TiMeOut=seconds]
[,SUMmary[={Yes|No}]]
[,DETail[={Yes|No}]]
[,VERBose[={Yes|No}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CMDQueue`

Displays command queue information.

`CONsole[={Yes|No}]`

- Yes — Write report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`DETail[={Yes|No}]`

- Yes — Display system status details, including GDDRMAIN subtasks and workers, and contents of command queue.

- No (**default**) — Do not display system status details.

`EVenTs`

Displays event information.

`SITE={DC1|DC2|DC3|DC4}`

Limits system status check to systems at the specified site.

`SUMmary[={Yes|No}]`

- Yes — Display system status summary.

  The following information is not included in the summary: subsystem name, IP address, port, GDDR maintenance level, compatibility level, licenses, and configuration type.

- No (**default**) — Do not display system status summary.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

`SYStem[=`*`system_name`*`]`

Displays and validates system status information across all systems where GDDRMAIN is active.

*`system_name`*

Limits system status check to the specified system.

`TiMeOut=`*`seconds`*

Sets system status check timeout.

By default, the timeout is variable depending on last known status of the remote system. Specifying an explicit timeout is not recommended.

*`seconds`*

Number of seconds to wait for status check per system. Valid values are from 0 to 60.

`VERBose[={Yes|No}]`

- Yes — Equivalent to the DETAIL option except that all events and their values are shown, as opposed to just showing the active events.
- No (**default**) — Shows only active events.

## Example

*Example 1: CHECKUP with no options*

The following example shows the CHECKUP command output:

**Note:** The example illustrates an SRDF/Star configuration.

```
GDDUC01I Health check information
 ------------------------------------------------------------------------
  Active events    : None
  Expected events  : None
  Degraded mode    : No       Star-HA    : 0
  Consistency      : Yes      DLM abort  : 0
 ------------------------------------------------------------------------
GDDUC02I Command queue
 ------------------------------------------------------------------------
  Status Owner    Count  Active Max
  ------ -------- ------ ------ ------
  Empty  n/a      0      0      8
 ------------------------------------------------------------------------
GDDUC03I System status
 ------------------------------------------------------------------------
  Site: DC1  System: X117  Type: Csys (master)  Subsys: GDDR

   IP: nnn.nnn.nnn.nnn Port: 7474

   Status: Active  Last comm: 06/23/17 12:24:39

   Maint level: V5.2.0 BASE (06/23/17)  Compat level: 2

   Licenses: GDDR  Config type: Cas Star-HA (CAX)

   System events: None
 ------------------------------------------------------------------------
  Site: DC2  System: X99  Type: Csys  Subsys: GDDR

   IP: nnn.nnn.nnn.nnn Port: 7474

   Status: Active  Last comm: 06/23/17 12:24:39

   Maint level: V5.2.0 BASE (06/23/17)  Compat level: 2

   Licenses: GDDR  Config type: Cas Star-HA (CAX)

   System events: None
 ------------------------------------------------------------------------
  Site: DC3  System: X98  Type: Csys  Subsys: GDDR

   IP: nnn.nnn.nnn.nnn Port: 7474

   Status: Active  Last comm: 06/23/17 12:24:39

   Maint level: V5.2.0 BASE (06/23/17)  Compat level: 2

   Licenses: GDDR  Config type: Cas Star-HA (CAX)

   System events: None
 ------------------------------------------------------------------------
  Site: DC1  System: HERCL887  Type: Psys

   IP: nnn.nnn.nnn.nnn Port: 7474

   Status: GDDR not active  Last comm: Never

   System events: None
 ------------------------------------------------------------------------
GDDUG48W 1 P-systems are inaccessible
```

## Output fields

### GDDUC01I Health check information

`Active events` Indicates any exceptions to conditions monitored by the GDDR Event Monitor.

"Monitored events" on page 63 lists the events monitored by the GDDR Event Monitor.

`Expected events` Lists expected events.

"Expected events" on page 69 discusses expected events.

`Degraded mode` Indicates whether Degraded mode is set (Yes) or not (No).

"Degraded mode" on page 70 explains Degraded mode.

`Star-HA` Indicates the status of the SRDF/Star High Availability (HA) feature. See "Star-HA" on page 369 for details.

`Consistency` Shows if there are any discrepancies between GDDR systems in regard to the GDDRPARM file or GDDR subsystem name. See "Consistency" on page 369 for details.

`DLM abort` Indicates the value of the DLM Abort global variable. See "DLm Abort" on page 370 for details.

**GDDUC02I Command queue**

`Status` The status of the GDDR command queue:

- `Empty`—No commands are on the queue.
- `In use`—One or more commands are on the queue.

  **Note:** This does not necessarily mean that the command queue ENQ is being actively held.

`Owner` The name of the job that owns the command queue ENQ.

`Count` The current queue size.

`Active` The number of active tasks in the command queue.

`Max` The maximum number of active tasks.

**GDDUC03I System status**

`Site` The site ID of the site where the system normally runs.

`System` The system name.

`Type` Indicates the GDDR system type:

- `Csys`—A control system. Master C-systems are shown as `(master)`.
- `Psys`—A GDDR-managed production or test system.

`Subsys` The GDDR subsystem name.

`IP` The system IP address.

`Port` The system port.

`Status` The system status.

The following statuses indicate that the system is in a valid state:

- `Active`—The system is accessible. GDDRMAIN and all required subtasks and workers are active.

- `Initializing(`*`subtasks`*`)` —The system is accessible, but GDDRMAIN is still initializing. GDDRMAIN subtasks and workers that are not yet active are listed in parentheses.
- `Terminating`—The system is accessible, but GDDRMAIN is in the process of shutting down.

The following statuses indicate that the system is not accessible, resulting in a warning message or messages and return code 4:

- `Request timed out`—The work request timed out waiting for response from the system. Ensure that the system is accessible and GDDRMAIN is accessible and active on the system, including the COMM subtask, the WORKMGR subtask, and GDDWST workers.
- `Request cancelled`—The work request was canceled. Ensure that the system is accessible and GDDRMAIN is active on the system, including the COMM subtask, the WORKMGR subtask, and GDDWST workers.
- `IP addr unreachable`—TCP/IP CONNECT failed with error code 49 (IP address is unreachable). Ensure that the IP address for the remote system is available.
- `Network down`—TCP/IP CONNECT failed with error code 50 (network is down). Ensure that the network path to the remote system is available.
- `Network unreachable`—TCP/IP CONNECT failed with error code 51 (network is unreachable). Ensure that the remote system is available and GDDRMAIN is active.
- `Connection timed out`—TCP/IP CONNECT failed with error code 60 (connection timed out). Ensure that the remote system is available and GDDRMAIN is active.
- `GDDR not active`—TCP/IP CONNECT failed with error code 61 (connection refused). Ensure that GDDRMAIN is active on the remote system and available on the port specified in GDDRPARM.
- `Host down`—TCP/IP CONNECT failed with error code 64 (host is down). Ensure that the remote system is available.
- `Host unreachable`—TCP/IP CONNECT failed with error code 65 (host is unreachable). Ensure that the network path to the remote system exists.
- *`tcpip-service`* `error` *`rc`*—Other TCP/IP error occurred. The named TCP/IP service failed with the specified error code.
- `Internal error` *`rc/rs/work-status/request-status`*—An internal error occurred. Contact Dell EMC Support providing the return code, reason code, work status, and request status.

The following statuses indicate that the system is in Degraded state, resulting in an error message or messages and return code 8:

- `Degraded(`*`subtasks`*`)` —The system is accessible, but at least one of the required GDDRMAIN subtasks or workers is not active. GDDRMAIN subtasks and workers that are not active are listed in parentheses.

`Last comm` — Last communication date and time. If no communication has been received from a given system, 'Never' is shown.

`Maint level` — GDDR version and maintenance level.

`Compat level` — GDDR compatibility level.

**Note:** GDDR compatibility level is intended for Dell EMC use.

`Licenses` — GDDR licenses installed on the system.

`Config type` — GDDR configuration type.

**Note:** "GDDRMAIN supported configuration types" on page 255 lists possible configuration types.

This field is displayed for C-systems only.

An asterisk (*) after the configuration type indicates a potential problem with the configuration. Issue the CONFIG command on that system for more information. If the CONFIG command shows that there is no longer a problem, the asterisk will be cleared from the configuration type.

`System events` — Lists active system-level events, both planned and unplanned. This field is displayed for C-systems only.

*Example 2: CHECKUP SUMMARY report* — The following example shows the CHECKUP SUMMARY command output:

**Note:** The example illustrates an SRDF/Star configuration.

```
GDDUC01I Health check information
 ------------------------------------------------------------------------
  Active events   : None
  Expected events : None
  Degraded mode   : No      Star-HA   : 0
  Consistency     : Yes     DLM abort : 0
 ------------------------------------------------------------------------
GDDUC02I Command queue
 ------------------------------------------------------------------------
  Status Owner    Count  Active Max
  ------ -------- ------ ------ ------
  Empty  n/a      0      0      8
 ------------------------------------------------------------------------
GDDUC03I System status
 ------------------------------------------------------------------------
  Site System   Type Status                          Last communicated
  ---- -------- ---- ------------------------------ -----------------
  DC1 *X117     Csys Active                         06/23/17 14:13:47
  DC2  X99      Csys Active                         06/23/17 14:13:47
  DC3  X98      Csys Active                         06/23/17 14:13:47
  DC1  HERCL887 Psys GDDR not active                Never
 ------------------------------------------------------------------------
  * Master C-system
GDDUG48W 1 P-systems are inaccessible
```

## Output fields

The meaning of the fields is as listed in "Example 1: CHECKUP with no options" on page 184.

*Example 3: CHECKUP DETAIL report*

The following example shows the CHECKUP DETAIL command output:

**Note:** The example illustrates an SRDF/Star with AutoSwap configuration.

```
GDDUC01I Health check information
 ------------------------------------------------------------------------
  Active events   : None
  Expected events : None
  Degraded mode   : No      Star-HA   : 0
  Consistency     : Yes     DLM abort : 0
 ------------------------------------------------------------------------
GDDUC02I Command queue
 ------------------------------------------------------------------------
  Status Owner    Count  Active Max
  ------ -------- ------ ------ ------
  In use GDDRCMDQ 3      0      8

  Commands:

  SQ GLOBAL
   Act: N  Done: N  RC: n/a  Added: 08/14/17 23:28:28  Script: GDDRCMDQ
  SQ GLOBAL
   Act: N  Done: N  RC: n/a  Added: 08/14/17 23:28:28  Script: GDDRCMDQ
  SQ GLOBAL
   Act: N  Done: N  RC: n/a  Added: 08/14/17 23:28:28  Script: GDDRCMDQ
 ------------------------------------------------------------------------
GDDUC03I System status
 ------------------------------------------------------------------------
  Site: DC1  System: X117  Type: Csys (master)  Subsys: GDDR

   IP: nnn.nnn.nnn.nnn Port: 7474

   Status: Active  Last comm: 08/23/17 14:24:02

   Maint level: V5.2.0 BASE (06/23/17)  Compat level: 2

   Licenses: GDDR  Config type: Cas Star-HA (CAX)

   System events: None

   Main task: GDDRMAIN (S0357358) ASID 0049

   Subtask Description       Act
   ------- ----------------- ---
   COMM    Communications     Y
   CONSOLE Console commands   Y
   GVT     Global variable    Y
   EVM     Event monitor      Y
   HBM     Heartbeat monitor  Y
   MCSOPER MCSOPER console    Y
   MISC    Miscellaneous      Y
   WORKMGR Work manager       Y

   Worker Description         Min Max Act
   ------ ------------------- --- --- ---
   GDDWCM Console message       1   2   1
   GDDWCO Command               1   5   1
   GDDWCX User msg automation   1   1   1
   GDDWDV Device                1  50   1
   GDDWGV Global variable       1   5   1
   GDDWST Status                2   6   6
   GDDWXH HMC                   1   5   1
   GDDWXQ Command queue         1   1   1
   GDDWXR REXX                  1   5   1
 ------------------------------------------------------------------------
```

```
Site: DC2  System: X99  Type: Csys  Subsys: GDDR

 IP: nnn.nnn.nnn.nnn Port: 7474

 Status: Active  Last comm: 08/23/17 14:24:02

 Maint level: V5.2.0 BASE (06/23/17)  Compat level: 2

 Licenses: GDDR  Config type: Cas Star-HA (CAX)

 System events: None

 Main task: GDDRMAIN (S0358539) ASID 0083

 Subtask Description       Act
 ------- ----------------- ---
 COMM    Communications     Y
 CONSOLE Console commands   Y
 GVT     Global variable    Y
 EVM     Event monitor      Y
 HBM     Heartbeat monitor  Y
 MCSOPER MCSOPER console    Y
 MISC    Miscellaneous      Y
 WORKMGR Work manager       Y

 Worker Description         Min Max Act
 ------ ------------------- --- --- ---
 GDDWCM Console message       1   2   1
 GDDWCO Command               1   5   1
 GDDWCX User msg automation   1   1   1
 GDDWDV Device                1  50   1
 GDDWGV Global variable       1   5   1
 GDDWST Status                2   6   6
 GDDWXH HMC                   1   5   1
 GDDWXQ Command queue         1   1   1
 GDDWXR REXX                  1   5   1
-------------------------------------------------------------------------
 Site: DC3  System: X98  Type: Csys  Subsys: GDDR

 IP: nnn.nnn.nnn.nnn Port: 7474

 Status: Active  Last comm: 08/23/17 14:24:02

 Maint level: V5.2.0 BASE (06/23/17)  Compat level: 2

 Licenses: GDDR  Config type: Cas Star-HA (CAX)

 System events: None

 Main task: GDDRMAIN (S0357357) ASID 00A5

 Subtask Description       Act
 ------- ----------------- ---
 COMM    Communications     Y
 CONSOLE Console commands   Y
 GVT     Global variable    Y
 EVM     Event monitor      Y
 HBM     Heartbeat monitor  Y
 MCSOPER MCSOPER console    Y
 MISC    Miscellaneous      Y
 WORKMGR Work manager       Y

 Worker Description         Min Max Act
 ------ ------------------- --- --- ---
 GDDWCM Console message       1   2   1
 GDDWCO Command               1   5   1
```

```
   GDDWCX User msg automation   1   1   1
   GDDWDV Device                1  50   1
   GDDWGV Global variable       1   5   1
   GDDWST Status                2   6   6
   GDDWXH HMC                   1   5   1
   GDDWXQ Command queue         1   1   1
   GDDWXR REXX                  1   5   1
 ------------------------------------------------------------------------
  Site: DC1  System: HERCL887  Type: Psys

   IP: nnn.nnn.nnn.nnn Port: 7474

   Status: GDDR not active  Last comm: Never

   System events: None
 ------------------------------------------------------------------------
GDDUG48W 1 P-systems are inaccessible
```

## Output fields

### GDDUC01I Health check information

See "GDDUC01I Health check information" on page 185.

### GDDUC02I Command queue

See "GDDUC02I Command queue" on page 186 for the meaning of the following fields:

- Status
- Owner
- Count
- Active
- Max

Commands — Lists commands in the GDDR command queue.

Act — Indicates if the command is active (Y) or not (N).

Done — Indicates if the command execution has completed (Y) or not (N).

RC — The command return code.

Added — Date and time when the command was added to the queue.

Script — The job name.

### GDDUC03I System status

See "GDDUC03I System status" on page 186 for the meaning of the following fields:

- Site
- System
- Type
- Subsys
- IP
- Port
- Status
- Last comm
- Maint level
- Compat level
- Licenses
- Config type

- System events

Main task — The job name and ID.

ASID — The address space ID.

Subtask — The name of the subtask.

Description — A short description of the subtask.

Act — Indicates whether the subtask is active (Y) or not (N).

Worker — The worker name.

Description — A short description of the worker.

Min — The minimum number of worker tasks per system.

Max — The maximum number of worker tasks per system.

Act — Indicates whether the worker is active (Y) or not (N).

*Example 4: CHECKUP VERBOSE report* — The following example shows an extract from CHECKUP VERBOSE command output:

**Note:** The example illustrates an SRDF/Star configuration.

```
GDDUC01I Health check information
 ------------------------------------------------------------------------
  All events      : C>>(0) CAX(0) CFG(0) CGD(0) CGT(0) DLM(0) ECA(0)
                    LDR(0) LNK.DC1vDC2(0) LNK.DC1vDC3(0)
                    LNK.DC2vDC3(0) MHB(0) MSC(0) MSF(0) MST(0) MXS(0)
                    NCX(0) RDF.DC1.DC2(0) RDF.DC1.DC3(0)
                    RDF.DC2.DC3(0) RDR(0) SRA(0) STR(0) CGV(0)
                    DC1.DLM(0) DC2.DLM(0) DC3.DLM(0) DC1.DC3.MSC(0)
                    DC1.DC3.SRA(0)
  Expected events : None
  Degraded mode   : No      Star-HA    : 0
  Consistency     : Yes     DLM abort  : 0
 ------------------------------------------------------------------------
GDDUC02I Command queue
...
GDDUC03I System status
...
```

## Output fields

### GDDUC01I Health check information

All events — Lists all events and their values. See "Monitored events" on page 63 for a description of each event.

See "GDDUC01I Health check information" on page 185 for the meaning of the other fields.

### GDDUC02I Command queue

See "GDDUC02I Command queue" on page 191.

### GDDUC03I System status

See "GDDUC03I System status" on page 186.

# COMM

Use the COMM command to display the list of systems, and IP addresses and ports comprising the GDDR-plex.

**Note:** COMM statements in the GDDRPARM file define the GDDR systems' IP addresses and ports, as described in "COMM" on page 259.

## Syntax

```
F GDDRMAIN,COMM[,<system>]
```

## Parameters

*system*

Optional system name. If *system* is specified, all input IP addresses are shown along with any DRTCOMM definitions.

## Example

The following example shows COMM command output:

```
GDDM127I Communications Status
Sys PRD1 ,      IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.50
Sys PRD2 ,      IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.50
Sys PRD3 ,      IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.50
Sys PRD4 ,      IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.50
Sys SYS4 (DC4), IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.51
Sys SYS3 (DC3), IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.51
Sys SYS2 (DC2), IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.51
Sys SYS1 (DC1), IP nnn.nnn.nnn.nnn,9876 Last 12/04/09 17:27:44.51
```

# CONFIG

Use the CONFIG command to validate and display all GDDR-managed SRDF groups (internal, external, and DLm) and show their status. The CONFIG command also indicates the configuration type, configuration mode, and topology.

Requirements are as follows:

- SCF must be active.
- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.
- Supported only on C-systems.

The CONFIG command is issued automatically on GDDRMAIN startup. The CONFIG command is also automatically issued at the start and end of all GDDR scripts, excluding GDDRMCPC, GDDRPGVB, and GDDRPXMC, to validate and display the configuration.

The CONFIG command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,ConFiG
[,SITE={DC1|DC2|DC3|DC4}]
[,ConTRoLler=<serial#>]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write report to console (default for console).
- No — Do not write report to console (default for batch).

`ConTRoLler=<serial#>`

Limits the scope of the command to SRDF group pairs which include the specified storage system.

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SITE={DC1|DC2|DC3|DC4}`

Limits the scope of the command to SRDF group pairs which include the specified site.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

The following example shows CONFIG command output:

```
GDDUG37I Configuration information
-------------------------------------------------------------------------
 Config type: SQAR
 Config mode: SQAR

 GDDR-managed RDF groups:

  Site A (DC1)  SRDF/S  Site B (DC2)  Typ Status
  ============ ======== ============  --- --------------------------
  000197700319 03 -> 03 000197700116  DLM Online SQAR-Mode
  000195700794 09 -> 09 000195700987  INT Online SQAR-Mode

  Site A (DC1)  SRDF/A  Site C (DC3)
  ============ ======== ============
  000197700319 D2 -> D2 000192600455  DLM Online SQAR-Mode MSC-Actv
  000195700794 28 -> 28 000195700866  INT Online SQAR-Mode MSC-Actv

  Site B (DC2)  SRDF/A  Site D (DC4)
  ============ ======== ============
  000197700116 D3 -> D3 000197700135  DLM Online SQAR-Mode MSC-Actv
  000195700987 38 -> 38 000197700135  INT Online SQAR-Mode MSC-Actv

  Site C (DC3) Recovery Site D (DC4)
  ============ ======== ============
  000195700866 D6 -> D6 000197700135  INT Online SQAR-Rcvy
  000192600455 D4 -> D4 000197700135  DLM Online SQAR-Rcvy
-------------------------------------------------------------------------
```

### Output fields

`Config type` The GDDR configuration type.

**Note:** "GDDRMAIN supported configuration types" on page 255 lists possible configuration types.

`Config mode` The configuration mode:

- `Half-SQAR`—Only one SQAR leg is active and is running in SQAR mode.
- `MSC`—The configuration is running in MSC mode
- `SQAR`—The configuration is running in SQAR mode

- `SRDF/S`—The configuration is running synchronous SRDF
- `Star`—The configuration is running in Star mode
- `Star-A`—The configuration is running in Star-A mode

`GDDR-managed RDF groups` Lists SRDF groups managed by GDDR.

`Site x (DCn)` The serial number of the system at the indicated site.

`SRDF/A | SRDF/A | Recovery` The SRDF group replication type and SRDF group numbers.

`Typ` The SRDF group type:

- `INT`—Internal
- `EXT`—External
- `DLM`—DLm

`Status` For each SRDF group pair, its status consists of a combination of the following strings.

Valid:

- `Online`—Online
- `MSC-Mode`—MSC mode
- `Star-Mode`—SRDF/Star mode
- `Star-Rcvy`—SRDF/Star recovery mode
- `StarA-Mode`—SRDF/Star-A mode
- `StarA-Rcvy`—SRDF/Star-A recovery mode
- `SQAR-Mode`—SRDF/SQAR mode
- `SQAR-Rcvy`—SRDF/SQAR recovery mode
- `MSC-Tag`—Cycle tag indicates MSC mode
- `Star-Tag`—Cycle tag indicates SRDF/Star mode
- `StarA-Tag`—Cycle tag indicates SRDF/Star-A mode
- `SQAR-Tag`—Cycle tag indicates SRDF/SQAR mode
- `MSC-Actv`—MSC active
- `SRA-Actv`—SRDF/A active

Undesirable, results in warning message(s) and RC=4:

- `C-Ahead*`—Site C ahead of site B
- `Int-Reqd*`—Intervention required
- `MSC-Actv*`—MSC active (not expected)
- `Xmit-Idle*`—The group is in the Transmit Idle state.

The SRDF group is in an invalid state, results in error message(s) and RC=8:

- `CycDelay*`—SRDF/A cycle delay
- `Gbl-Incon*`—Globally inconsistent
- `MSC-Inact*`—MSC not active
- `MSC-Mode*`—MSC mode (not expected)
- `MSC-Tag*`—Cycle tag indicates MSC mode (not expected)
- `Nonexist*`—The SRDF group does not exist.
- `No-Rcvy*`—Recovery not available
- `Not-SQAR*`—Not in SRDF/SQAR mode (SRDF/SQAR mode expected)
- `Not-Star*`—Not in SRDF/Star mode (SRDF/Star mode expected)
- `Not-StarA*`—Not in SRDF/Star-A mode (SRDF/Star-A mode expected)
- `Offline*`—The SRDF group is offline.
- `Rmt-Incon*`—Remote inconsistent
- `SQAR-Mode*`—SRDF/SQAR mode (not expected)
- `SQAR-Rcvy*`—SRDF/SQAR recovery mode (not expected)
- `SQAR-Tag*`—Cycle tag indicates SRDF/SQAR mode (not expected)
- `SRA-Actv*`—SRDF/A active (not expected)
- `SRA-Inact*`—SRDF/A not active
- `Star-Mode*`—SRDF/Star mode (not expected)
- `Star-Rcvy*`—SRDF/Star recovery mode (not expected)
- `Star-Tag*`—Cycle tag indicates SRDF/Star mode (not expected)
- `StarA-Mode*`—SRDF/Star-A mode (not expected)
- `StarA-Rcvy*`—SRDF/Star-A recovery mode (not expected)
- `StarA-Tag*`—Cycle tag indicates SRDF/Star-A mode (not expected)

The SRDF group state could not be determined, results in RC=12:

- `??? (Controller not found)`—The storage system is inaccessible or not defined to GDDR.
- `??? (Site not found)`—The site is not defined to GDDR.

# DLM

The DLM command enables you to do the following:

- Determine the status of DLm systems managed by GDDR (the DLM command issued with no parameters).

  The DLM command issues a heartbeat request to every DLm system defined at the Define DLm Systems panel (M,P,D,D) described in "Define DLm systems (M,P,D,D)" on page 314, and reports whether the DLm is alive or dead.
- Set or reset the DLm Abort flag.
- Set the active control station for Stateless DLm.

## Syntax

```
F GDDRMAIN,DLM
F GDDRMAIN,DLM,{SET|RESET},ABORT
F GDDRMAIN,DLM,{CS0|CS1}
```

## Parameters

`{CS0|CS1}`

Sets the active control station for Stateless DLm.

`{SET|RESET}, ABORT`

Sets or resets the DLm Abort flag.

# ENQ

Use the ENQ command to display GDDR ENQs that are currently held or waiting to be held.

GDDR ENQs are typically short-lived. If there are no GDDR ENQs currently held, the display will indicate: `*** No ENQs found ***`. This is normal.

The ENQ command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

The ENQ command is issued automatically for diagnostic purposes in case GDDR fails to acquire the command queue processing ENQ.

---

**Note:** The command queue processing ENQ prevents a GDDR script or other process from clearing the command queue while it is still in use, which can lead to abends. This can occur if a script is canceled, and the command queue worker is still processing commands, and then a new script is submitted.

---

## Syntax

```
F GDDRMAIN,ENQ
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

---

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

---

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

The following example shows ENQ command output:

```
GDDUG27I GDDR ENQs
 ------------------------------------------------------------------------
  Resource name              System   Job name ASID Control   Status
  -------------------------- -------- -------- ---- --------- ------
  GDDR.GDDRGD2COMMQ          X117     GDDRMAIN 01AF Exclusive Own
  GDDR.GDDRGD2COMMQ          X117     GDDRMAIN 01AF Exclusive Wait
  GDDR.GDDRGD2COMMQ          X117     GDDRMAIN 01AF Exclusive Wait
 ------------------------------------------------------------------------
```

### Output fields

`Resource name` The resource name in format *major*.*minor*.

`System` The system name.

`Job name` The job name.

`ASID` The address space ID.

`Control` Possible control type values are `Exclusive` or `Share`.

`Status` Possible status values are `Own` and `Wait`.

# GATEK

Use the GATEK command to validate and display all paths (both local and remote) to GDDR-managed systems that should be accessible to the site where the command is run. The GATEK command shows the path status.

**Note:** Remote paths are only validated once per SRDF group, not once per CUU per SRDF group, unless the ALL option is specified.

In addition, the GATEK command can optionally unbox boxed devices.

Requirements are as follows:

- SCF must be active.
- GDDRMAIN must be active.
- Supported only on C-systems.

The GATEK command is issued automatically on GDDRMAIN startup.

The GATEK command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,GateK
[,SITE={DC1|DC2|DC3|DC4}]
[,ConTRoLler=<serial#>]
[,Auto_UnBoX[={Yes|No}]]
[,PRompt[={Yes|No}]]
[,SIMulate[={Yes|No}]]
[,FoRCe[={Yes|No}]]
[,ALL[={Yes|No}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`ALL[={Yes|No}]`

- Yes — Validate and display all paths using GDDR gatekeepers, including those to systems not managed by GDDR. Validate remote paths once per CUU per SRDF group, instead of just once per SRDF group, to ensure every possible path is tested.

- No (**default**) — Validate and display only paths between GDDR-managed systems. Validate remote paths only once per SRDF group, as opposed to once per CUU per SRDF group, to avoid unnecessary overhead.

`Auto_UnBoX[={Yes|No}]`

- Yes — Automatically unbox devices.

  **Note:** GDDRMCMD must be APF-authorized in order to use the Auto_UnBoX option and should be added to the IKJTSO parameter file under AUTHPGM.

- No (**default**) — Do not unbox devices.

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`ConTRoLler=<serial#>`

Limits the scope of the command to gatekeepers (or remote paths) to the specified storage system.

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`FoRCe[={Yes|No}]`

- Yes — Force unboxing of devices when it normally would not be allowed (for example, the device is online or pending offline). There is no guarantee a device can be unboxed, but the usual validation will be bypassed.

  **Note:** Valid with Auto_UnBoX=Yes only.

- No (**default**) — Do not forcefully unbox devices.

`PRompt[={Yes|No}]`

- Yes — Prompt before unboxing a device.

  **Note:** Valid with Auto_UnBoX=Yes only.

- No (**default**) — Suppress all prompts.

`SIMulate[={Yes|No}]`

- Yes — Simulate unboxing of devices without actually unboxing.

  **Note:** Valid with Auto_UnBoX=Yes only.

- No (**default**) — Disable simulation mode.

`SITE={DC1|DC2|DC3|DC4}`

Limits the scope of the command to gatekeepers (or remote paths) to the specified site.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

The following example shows GATEK command output:

```
GK
GDDUG32I Gatekeeper information  775
------------------------------------------------------------------------
 Site Controller     Path     Status
 ---- -------------  -------  --------------------------------------
 DC1  0001957-00794  7D40     functional  dv-online
 DC1  0001957-00794  7D41     functional  dv-online
...
 DC2  0001957-00987  8A80     functional  dv-online
 DC2  0001957-00987  8A81     functional  dv-online
...
 DC3  0001926-00455  2710.D2  functional
 DC3  0001957-00866  7D40.02  functional  dv-online
...
 DC4  0001977-00135  1730.D3  functional
 DC4  0001977-00135  8A80.38  functional  dv-online
------------------------------------------------------------------------
GDDUG70W 31 online gatekeeper devices found
```

### Output fields

`Site` The site ID.

`Controller` The storage system serial number.

`Path` The path in format *gatekeeper*[.*srdfgrp*].

`Status` For each path, its status consists of a combination of the following strings.

Valid:

- `functional`
- `(unBOXed)`

Undesirable, results in warning message(s) and RC=4:

- `dv-allocated`
- `dv-online`

Path is not functional, results in error message(s) and RC=8:

- `API error (rc/rs/rcx)`
- `BOXed`
- `defer BOXed`
- `IO error`

- IO stopped by IOACTION STOP cmd
- IO timeout
- hot IO
- not connected
- no link available
- no logical paths
- no paths
- permanent error
- remote syscall timeout
- swap in progress
- syscall error (*syscall_rc*)
- syscall timeout
- wrong controller (*serial#*)
- UCB not found
- UCB not valid

# FIXPERT

Enables or disables the Resiliency Expert utility.

**Note:** "Recovering from script errors" on page 527 describes the Resiliency Expert utility.

When run without the ON or OFF parameter, the FIXPERT command shows the current state of the utility.

## Syntax

```
F GDDRMAIN,FIXPERT[,{ON|OFF}]
```

## Parameters

`ON`

Enables the Resiliency Expert.

`OFF`

Disables the Resiliency Expert.

# GVB

Use the GVB command to schedule a regular backup of global variables or to perform a backup on demand.

**Note:** "Creating GDDR parameter backups" on page 291 provides information on backups.

When the scheduled time comes, the backup will be postponed if a script is in progress.

The default value for GVB is NONE, meaning that no backups are scheduled. This default can be overridden via the GDDRMAIN EXEC parameter GVB described in "GVB" on page 171.

When the GVB command is run without any parameters, it returns the currently scheduled backup day and time.

## Syntax

```
F GDDRMAIN,GVB

F GDDRMAIN,GVB=NONE|NOW

F GDDRMAIN,GVB={dw|*}(hhmm)
```

## Parameters

{dw|*}(*hhmm*)

Schedules the backup at the specified time:

- *dw* is a day of the week (first two letters).
- * specifies every day.
- *hhmm* is a time (hour and minutes, 24 hour clock).

NONE

(**Default**) Causes backups to not be scheduled.

NOW

Causes an immediate backup.

# LICENSE

Use the LICENSE command to validate and display GDDR license information across all systems in the GDDR-plex where GDDRMAIN is active.

Requirements are as follows:

- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.

If the license check fails for a remote system, the failure reason is reported, resulting in a warning message(s) and RC=4.

The LICENSE command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,LICense
[,TiMeOut=<seconds>]
[,SITE={DC1|DC2|DC3|DC4}]
[,SYStem=<system-name>]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SITE={DC1|DC2|DC3|DC4}`

Limits license check to systems at the specified site.

The site for a given C-system is defined using the CSYSSITE statement in the GDDRPARM file (as described in "CSYSSITE" on page 263). The site for a given production system is defined using the Define Managed Systems panel (M,P,H,S) described in "Define managed systems (M,P,H,S)" on page 318.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

`SYStem=<`*`system-name`*`>`

Limits license check to the specified system.

`TiMeOut=<`*`seconds`*`>`

Number of seconds to wait for license check per system. Valid range is 0-60. By default, timeout is variable depending on last known status of the remote system.

**Note:** Specifying an explicit timeout is not recommended.

## Example

**Note:** The example illustrates a 2-site SRDF/S configuration.

```
GDDUG30I License information
---------------------------------------------
  T Site System   Licenses
  - ---- -------- --------------------------
  C DC1  DVTL1161 GDDR
  C DC2  DVTL149  GDDR
  P DC1  CLAYB033 ???  (GDDR not active)
  P DC2  CLAYB034 ???  (GDDR not active)
 ---------------------------------------------
GDDUG48W 2 P-systems are inaccessible
```

### Output fields

`T` The system type:

- `C`—Control system (C-system)
- `P`—Production system (managed system)

`Site` The site ID.

`System` The system name.

`Licenses` Lists available GDDR licenses.

**Note:** See "Install GDDR Licensed Feature Code" on page 144 for information about licenses.

If the license check fails for a remote system, the licenses are displayed as `???` followed by one of the following reasons in parentheses, resulting in warning message(s) and RC=4:

- `Connection timed out`—TCP/IP CONNECT failed with error code 60 (connection timed out). Ensure the remote system is available and GDDR is active.
- `GDDR not active`—TCP/IP CONNECT failed with error code 61 (connection refused). Ensure GDDR is active on the remote system and available on the port specified in GDDRPARM.
- `Host down`—TCP/IP CONNECT failed with error code 64 (host is down). Ensure the remote system is available.
- `Host unreachable`—TCP/IP CONNECT failed with error code 65 (host is unreachable). Ensure the network path to the remote system exists.
- `Internal error` *`rc/rs/work-status/request-status`*—Internal error occurred. Contact Dell EMC Technical Support providing the indicated return code, reason code, work status, and request status.
- `IP addr unreachable`—TCP/IP CONNECT failed with error code 49 (IP address is unreachable). Ensure the IP address for the remote system is available.
- `Network down`—TCP/IP CONNECT failed with error code 50 (network is down). Ensure the network path to the remote system is available.
- `Network unreachable`—TCP/IP CONNECT failed with error code 51 (network is unreachable). Ensure the remote system is available and GDDR is active.
- `Request timed out`—Work request timed out waiting for response from the system. Ensure that the system is available. Verify that GDDRMAIN and required subtasks and workers (COMM, WORKMGR, GDDWST) are active on the system.
- *`tcpip_service`* `error` *`rc`*—Other TCP/IP error occurred. The named TCP/IP service failed with the specified error code.

# LOCK

Use LOCK to display or clear the index, or update locks, or both.

**Note:** See "GDDR locks" on page 277 for lock descriptions, status definitions, and administrative considerations of the locks.

## Syntax

```
F GDDRMAIN,LOCK[,<lock-name>]
F GDDRMAIN,LOCK,<lock-name>,<cmd>
```

## Parameters

*cmd*

Is DISPLAY or CLEAR. The default command is DISPLAY.

The status of the requested lock will be displayed (CLEAR, SET, IN USE, PENDING, or BROKEN).

- If the status is SET, the job name and ID of the locker is displayed along with the number of seconds since the lock was set.
- If the status is IN USE or PENDING, the number of seconds since the shared lock was changed is displayed.

*lock-name*

Is INDEX, UPDATE, or BOTH. The default value for *lock-name* is BOTH.

# MAINTENANCE

Use the MAINTENANCE command to display GDDR maintenance information (including the GDDR version, highest PTF number, latest build date, and cumulative maintenance level CRC) for the address space where the command is issued and validate the reported maintenance level against that of the GDDRMAIN address space, if applicable.

The MAINTENANCE command is issued automatically on startup of GDDRMAIN, GDDREVM, GDDRWORK, and GDDR scripts, as well as by the GDDRECHK utility.

The MAINTENANCE command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,MAINTenance
[,SUMmary[={Yes|No}]]
[,COMposite[={Yes|No}]]
[,DETail[={Yes|No}]]
[,LoadMOD=<module>]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`COMposite[={Yes|No}]`

- Yes (**default**) — Display composite load module report, including the following:
  - For each load module, the GDDR version, highest PTF number, and latest build date amongst all CSECTs.
  - The cumulative maintenance level amongst all load modules, including the GDDR version, highest PTF number, and latest build date.

  **Note:** "Example 2: MAINTENANCE COMPOSITE report" on page 213 shows the MAINTENANCE COMposite report.
- No — Do not display composite load module report.

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`DETail[={Yes|No}]`

- Yes — Display CSECT detail report, including the following:
  - For each CSECT, the SCLM project, PTF number, build date and time.
  - For each load module, the GDDR version, highest PTF number, and latest build date amongst all CSECTs.
  - The cumulative maintenance level amongst all load modules, including the GDDR version, highest PTF number, and latest build date.

  **Note:** "Example 3: MAINTENANCE DETAIL report" on page 214 shows the MAINTENANCE DETail report.

- No (**default**) — Do not display CSECT detail report.

`LoadMOD=<module>`

Display CSECT detail report for specified load module, including the following:

- For each CSECT, the SCLM project, PTF number, build date and time.
- The cumulative maintenance level amongst all CSECTs, including the GDDR version, highest PTF number, and latest build date.

**Note:** "Example 4 MAINTENANCE LOADMOD report" on page 215 shows the MAINTENANCE LoadMOD report.

`SUMmary[={Yes|No}]`

- Yes — Show cumulative maintenance level only, including the GDDR version, highest PTF number, and latest build date amongst all load modules.

  **Note:** "Example 1: MAINTENANCE SUMMARY report" on page 213 shows the MAINTENANCE SUMmary report.

- No (**default**) — Do not limit report to cumulative maintenance level only.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Examples

*Example 1: MAINTENANCE SUMMARY report*

The following example shows the MAINTENANCE SUMmary report:

```
MAINT SUM
...
GDDM072I GDD5 GDDR maintenance level: V5.2.0 BASE (08/13/18)  [CRC 3086BBF0]
GDDM074I GDD5 GDDR maintenance level matches GDDRMAIN address space
```

The report shows the maintenance level of the address space where the MAINTENANCE command was issued, including the following:

- GDDR version
- The highest PTF number amongst all load modules
- The latest build date amongst all load modules
- CRC of maintenance level of all load modules

*Example 2: MAINTENANCE COMPOSITE report*

The following example shows the MAINTENANCE COMposite report:

```
MAINT COM
...
GDDM072I GDD5 GDDR maintenance level: V5.2.0 BASE (08/13/18)  [CRC 3086BBF0]
GDDM074I GDD5 GDDR maintenance level matches GDDRMAIN address space

GDDUG33I GDD5 Module-level maintenance
 ------------------------------------
  Appl  Date      Ver  Level
  ----  --------  ---  -------
  GDDR  08/13/18 520  BASE

    Module    Date      Ver  Level
    --------  --------  ---  -------
    GDDBCPCM  08/13/18 520  BASE
    GDDBCPCO  08/13/18 520  BASE
....
```

### Output fields

#### Application information

`Appl` The application name.

`Date` The latest build date amongst all load modules.

`Ver` The application version.

`Level` The highest PTF number.

#### Module information

`Module` The module name.

`Date` The latest build date amongst all CSECTs.

`Ver` The application version.

`Level` The highest PTF number.

*Example 3: MAINTENANCE DETAIL report*

The following example shows the MAINTENANCE DETail report:

```
MAINT DET
...
GDDM072I GDD5 GDDR maintenance level: V5.2.0 BASE (08/13/18)  [CRC 3086BBF0]
GDDM074I GDD5 GDDR maintenance level matches GDDRMAIN address space

GDDUG38I GDD5 CSECT-level maintenance
 ------------------------------------------------
  Appl  Date      Ver  Level
  ----  --------  ---  -------
  GDDR  08/13/18 520  BASE

    Module    Date      Ver  Level
    --------  --------  ---  -------
    GDDBCPCM  08/13/18 520  BASE

      CSECT     Date      Time   Project  Level
      --------  --------  -----  -------  -------
      GDDBCPCM  08/13/18 16.06  GDDR520  BASE
      GDDACPCM  08/13/18 16.05  GDDR520  BASE
...

    Module    Date      Ver  Level
    --------  --------  ---  -------
    GDDBCPCO  08/13/18 520  BASE

      CSECT     Date      Time   Project  Level
      --------  --------  -----  -------  -------
      GDDBCPCO  08/13/18 16.06  GDDR520  BASE
      GDDBCPC2  08/13/18 16.06  GDDR520  BASE
...
```

## Output fields

### Application information

See "Application information" on page 213.

### Module information

See "Module information" on page 213.

### CSECT information

`CSECT` The CSECT name.

`Date` The built date.

`Time` The built time.

`Project` The SCLM project.

`Level` The PTF number.

*Example 4 MAINTENANCE LOADMOD report*

The following example shows the MAINTENANCE LoadMOD report:

```
MAINT LMOD=GDDRMAIN
...
GDDUG39I GDD5 Selected module maintenance
 ---------------------------------------------
  Module    Date      Ver  Level
  --------  --------  ---  -------
  GDDRMAIN  08/13/18 520  BASE

    CSECT     Date      Time   Project  Level
    --------  --------  -----  -------  -------
    GDDRMAIN  08/13/18 16.08  GDDR520  BASE
    TRMEXIT   08/13/18 16.08  GDDR520  BASE
    GDDMAIN1  08/13/18 16.08  GDDR520  BASE
    GDDRDELW  08/13/18 16.07  GDDR520  BASE
    GDDRGETW  08/13/18 16.08  GDDR520  BASE
    GDDRGETD  08/13/18 16.08  GDDR520  BASE
    GDDRSSVL  08/13/18 16.11  GDDR520  BASE
    GDDRSSYS  08/13/18 16.11  GDDR520  BASE
    GDDRMSG   08/13/18 16.09  GDDR520  BASE
    GDDRMSGT  08/13/18 16.09  GDDR520  BASE
...
```

## Output fields

### Module information

See “Module information” on page 213.

### CSECT information

See “CSECT information” on page 214.

# MPARM

Use the MPARM command to display GDDRPARM file checksums which indicate when the copies of the GDDRPARM file referenced by GDDRMAIN on each system are consistent.

The MPARM command output lists each system with which GDDRMAIN can possibly communicate. The display shows each system name, the checksum of the GDDRPARM file data in use, the checksum of the contents of the GDDRPARM file, and whether or not the system has found different GDDRPARM file data from other systems. If GDDRMAIN cannot communicate with the system, the "unable to communicate" status is displayed.

Use the MPARM CHECK command to force a return to synchronization if the GDDRPARM file data in use everywhere is synchronized. The MPARM CHECK command retrieves the GDDRPARM file in-use checksum from each system and compares them all. If all are equal, consistency is restored and degraded status is turned off. MPARM CHECK will tolerate no response from managed systems, but must get a response from each C-system in order to declare consistency. If any system (C-system or managed system) responds with a different in-use checksum value, consistency will not be declared.

## Syntax

```
F GDDRMAIN,MPARM
```

```
F GDDRMAIN,MPARM,CHECK
```

## Example

- The following example shows MPARM command output when all GDDRMAIN tasks are consistent and capable of communicating:

```
GDDM140I GDDRPARM Status
Sys PRD1 : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
Sys PRD2 : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
Sys PRD2 : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
Sys SYS3 : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
Sys SYS2 : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
Sys SYS1 : In-use 0C63D749, Dataset 0F3F7372, Consistency Y
```

- Sample output of MPARM CHECK is as follows:

```
GDDM142I Consistency set globally
GDDM142I Consistency set locally
```

  The 'Consistency set locally' version of the message will appear in the system log of each system with successful communication.

  If any C-system cannot be reached, the output is:

```
GDDM141E GDDRPARM Inconsistency detected, C-System SYS1 is down
```

  If any system has a different in-use checksum, the output is:

```
GDDM141E GDDRPARM Inconsistency detected, System SYS1 is
inconsistent
```

# MSGS

Use the MSGS command to enable or disable GDDR message rules.

GDDR message rules list message IDs that indicate changes in storage or system status, delivered in *hlq*.GDDR*vrm*.LINKLIB(GDDRMMSG).

## Syntax

```
F GDDRMAIN,MSGS[,<msgid>]
F GDDRMAIN,MSGS,<msgid>,{ENABLE|DISABLE}
```

## Example

Sample MSGS command output is as follows:

```
GDDM130I Message Status
+GDDR721 (E) I:1 01/24/17 17:19:12.01, P:1 01/24/17 17:19:12.01
+GDDR786 (D) I:0 *NEVER*, P:0 *NEVER*
```

- `(E)` or `(D)` indicates if interception of the message is enabled (E) or disabled (D).
- `I:`*n* followed by a timestamp indicates the number of times and the last date and time (GMT) when the message was intercepted on this system.
- `P:`*n* followed by a timestamp indicates the number of times and the last date and time (GMT) when the message was processed on this system.

To display a particular message, for example message EMCGM9EE, issue MSGS,EMCGM9EE:

```
GDDM130I Message Status
 EMCGM9EE, E/D E, Int 1 01/07/10 20:44:53.61, Proc 1 01/07/10 20:44:53.
```

To enable or disable interception of a particular message, for example message EMCGM9EE, use MSGS,EMCGM9EE,DISABLE:

```
GDDM130I Message Status
 EMCGM9EE, E/D D, Int 1 01/07/10 20:44:53.61, Proc 1 01/07/10 20:44:53.
```

# PARM_REFRESH

Use the PARM_REFRESH command to reread the GDDRPARM file and perform the following actions:

- Refresh GDDR licenses

  **Note:** "Updating GDDR licenses" on page 572 instructs on how to update GDDR license information.

- Flag any unlicensed GDDRPARM file statements
- Process MSCGROUP statements (MSC group names)
- Process MULTI statements (used to enable or disable the GDDR C-System Multi-Tenancy feature)
- Process SITE statements (configured sites)

**Note:** See "Updating GDDRPARM file statements" on page 573 for actions that should be taken after you update statements in the GDDRPARM file.

Requirements are as follows:

- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.

The PARM_REFRESH command is issued automatically on GDDRMAIN startup with the ECHO option. The ECHO option is used to ensure the GDDRPARM statements are included in the GDDRMAIN log. If no valid GDDR licenses are found, GDDRMAIN terminates.

The PARM_REFRESH command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

The PARM_REFRESH command should be followed by the MPARM,CHECK command to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.

## Syntax

```
F GDDRMAIN,PaRM_REFresh
[,ECHO[={Yes|No}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

### Parameters

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`ECHO[={Yes|No}]`

- Yes — Echo all GDDRPARM statements.
- No (**default**) — Do not echo GDDRPARM statements.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

### Example

**Note:** The example illustrates an SRDF/Star configuration.

```
GDDM604I GDD2 Licenses set: GDDR
GDDM620I GDD2 Configuration type detected: Con Star-HA (CAX)
GDDM621I GDD2 Sites defined: DC1 (region 1), DC2 (region 1), DC3
   (region 2)
GDDM622I GDD2 MSC groups defined: L5GD2DC1, L5GD2DC2, L5GD2DC3,
   L5GD2DC3
GDDM610I GDD2 Multi-GDDR is enabled for subsystem GDD2
   Other GDDR instances active on LB06M38:
    - GDD1
GDDM699I GDD2 GDDR PARM_REFRESH complete
```

**Note:** "GDDRMAIN supported configuration types" on page 255 lists possible configuration types.

## RDFREFR

GDDRMAIN creates a map of SRDF links between storage systems when the WORKMGR subtask starts. If an SRDF group is added or removed, this map can be out of sync with reality. The RDFREFR command reconstructs this map. Previously, it was necessary to restart WORKMGR to reconstruct the SRDF link map.

# REGION

Use the REGION command to display above and below the line REGION limits and allocation information for the address space where the command is issued.

The REGION command is issued automatically on startup of GDDRMAIN, GDDREVM, GDDRWORK, and GDDR scripts.

The REGION command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,ReGioN
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes—Write report to console (default for console).
- No—Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes—Enable debugging to SYSPRINT DD.
- No (**default**)—Disable debugging.

`SysMSG[={Yes|No}]`

- Yes—Write body of report to JESYSMSG.
- No (**default**)—Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes—Write report to SYSPRINT (default for batch).
- No—Do not write report to SYSPRINT (default for console).

## Example

The following example shows REGION command output:

```
REGN
GDDUG29I REGION limits and usage
 ---------------------------------------------
                   Limit   Allocated  % Alloc
                 -------  ---------  -------
  Above the line    1.6G      3.1M        0
  Below the line   10.0M    208896        2
 ---------------------------------------------
```

## Output fields

The output fields are as follows:

| | |
|---|---|
| `Limit` | Shows the above the line and below the line REGION limit. |
| `Allocated` | Shows the above the line and below the line REGION allocation. |
| `% Alloc` | Shows the above the line and below the line REGION percent allocation. |

# RELOAD

Use RELOAD to reload the GDDRMAIN interface modules into dynamic LPA. This allows select GDDR fixes to be applied without restarting GDDRMAIN.

**⚠WARNING**

**Use RELOAD only under direction of Dell EMC Customer Support.**

The RELOAD command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

If RELOAD is issued as a GDDRMAIN modify command, GDDRMAIN automatically issues the MAINTENANCE command to update stored maintenance information. Otherwise, issue the MAINTENANCE command manually. Failure to do so could result in maintenance mismatches, preventing GDDR address spaces from initializing, including scripts.

The requirements are as follows:

- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.

## Syntax

```
F GDDRMAIN,ReLoad
[,FoRCe[={Yes|No}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

FoRCe[={Yes|No}]

- Yes — Force reloading of GDDRMAIN interface modules even if they appear to still be in use.

  **Note:** FORCE(Yes) can result in abends if in fact the previously loaded modules are still in use.

- No (**default**) — Do not forcefully reload modules.

SysMSG[={Yes|No}]

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

SysPRinT[={Yes|No}]

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

The following example shows RELOAD command output:

```
RELOAD
GDDM218I Module GDDRPCR loaded into dynamic LPA from STEPLIB
GDDM072I GDDR maintenance level: V5.2.0 GD52001 (08/04/18) [CRC
   58C9F833]
GDDM073I GDDR maintenance level upgraded from V5.2.0 BASE (08/03/18)
   to V5.2.0 GD52001 (08/04/18)
```

# RESTART

Use RESTART to stop a subtask which is running and restart it immediately.

RESTART will post the subtask to end, but if it does not end, you can issue the CANCEL command to stop it. If you use CANCEL, the automatic restart will not occur, but you can issue START to restart the subtask.

Table 27 shows GDDRPARM statements that are reread and processed when a subtask is restarted:

**Table 27** Reread and processed GDDRPARM statements

| Subtask type | RESTART command example | Reread and processed GDDRPARM statements |
|---|---|---|
| Work manager subtask | `F GDDRMAIN,RESTART WORKMGR` | CMDQMAXT, SYMM, and WORKER statements |
| Communications subtask | `F GDDRMAIN,RESTART COMM` | COMM and DRTCOMM statements |
| Miscellaneous subtask | `F GDDRMAIN,RESTART MISC` | CPC and VCPC statements |
| Communications subtask and miscellaneous subtask | `F GDDRMAIN,RESTART COMM and F GDDRMAIN,RESTART MISC` | CSYSSITE statements |
| Global variable manager subtask | `F GDDRMAIN,RESTART GVT` | GVDIVDSN statements |
| MCSOPER console subtask | `F GDDRMAIN,RESTART MCSOPER` | MSG statements |

## Syntax

```
F GDDRMAIN,RESTART <subtask-name>
```

# SCRIPT

Use SCRIPT to submit GDDR scripts for execution.

IMPORTANT

Do not run GDDR scripts during DLm configuration changes.

## SCRIPT command process flow

Use of the SCRIPT command results in work being sent to the GDDWXR worker to run module GDDRMCSS under ISPF under the TMP. This is so that file tailoring is available. No JCL modifications are required for any GDDR procs. File tailoring (if done) uses the ISPSLIB DD in the GDDRWORK proc. The skeleton to be used (GDDRXSUB) is very similar to GDDRXACT. It can be in a different dataset (still in the ISPSLIB DD concatenation) so that the contents can be user-controlled. However, it does not have to be in a different dataset and is provided ISPSLIB.

The general flow (which can be modified by various keywords) is as follows:

```
F GDDRMAIN,SCRIPT,script_id
GDDM251I Script command work submitted
```

(Where *script_id* is the script ID, for example, GDDRPA42.)

This is followed by a WTOR issued by a GDDRWORK address space:

```
GDDM252A Enter script parameters
```

The response to this is a comma-separated string of keywords as shown below. If CANCEL is entered, the process ends with no script generation or submission. The message is repeated until either END or CANCEL is entered or implied.

When END is entered, the following message is displayed showing all the options in effect:

```
GDDM253I Script script_id parameters:
   Site   - DC1
   System - *
   ...
   Submit - YES
```

This is followed by the following message:

```
GDDM255A Confirm script generation (Y/N/CANCEL)
```

- If N is entered, return to the GDDM252A message.
- If CANCEL is entered, the process ends at this point with no further action.
- If Y is entered, the script is generated (file tailored). The resultant JCL is saved if JCLSAVE was specified, and is submitted if SUBMIT was specified.

If a job is submitted, the following message occurs:

```
GDDM266I Script script_id submitted, JOB jobname (jobid)
```

### Authorization

Skeleton member GDDRXSUB supports secure access to GDDR scripts run through the SCRIPT console command. You may concatenate a PDS containing customized JCL skeleton GDDRXSUB to the GDDRMAIN ISPSLIB DD statement. This skeleton must provide a job card.

As scripts will be submitted by GDDRMAIN, they would normally run with GDDRMAIN authorizations. The job card provided in skeleton GDDRXSUB may have a USER= keyword. The named user would then be the only one with all the authorizations that are required to run a GDDR script. GDDRMAIN would only need SURROGAT authority to that user ID.

## Syntax

```
F GDDRMAIN,SCRIPT,<script-name>,
[,1SIDE={Y|N}]
[,ALLOWHMC={Y|N}{Y|N}{Y|N}{Y|N}] (PRIMARY, SECONDARY,
WTOR, LOADCLEAR)
[,ALTPARMS={Y|N}]
[,AUTO]
[,BCVGROUPS={Y|N}]
[,BGCLONE={Y|N}]
[,CALLOVER=<call-overrides>]
[,DRTPARMS={Y|N}]
[,EXTBCV={Y|N}]
[,EXTDEV={Y|N}]
[,LOAD=<script-parm-file>]
[,PRISWAP=<primary-swap-group>]
[,RRSKIP={Y|N}]
[,SDDFCLN={Y|N}]
[,SECSWAP=<secondary-swap-group>]
[,SELSYS=<lpar>|(<lpar>,<lpar>,...)]
[,SITE=DCn]
[,SUBMIT={Y|N}]
[,SYSTEM=<system-name>]
[,TESTBCV1={Y|N}]
[,TESTBCV2={Y|N}]
[,TESTBCV3={Y|N}]
[,TESTBCV4={Y|N}]
[,USER=<userid>]
[,ZDPPAUSE={Y|N}]
[,ZDPRESUME={Y|N}]
```

```
[,ZDPUPLA={Y|N}]
[,ZDPUPLB={Y|N}]
[,ZDPUPLC={Y|N}]
[,ZDPUPLD={Y|N}]
```

All keywords after SCRIPT are optional. However, if options are specified, then *script-name* must first be specified. If the script name is going to be supplied via a LOAD keyword, the script name can be specified as '*'.

## Parameters

*script-name*

The name of a GDDR script supported through this interface (such as GDDRPGVB). Chapter 8, "Running GDDR Scripts," lists valid script names.

If the script name is going to be supplied via a LOAD keyword, the script name can be specified as '*'.

1SIDE={Y|N}

Recover replication on the SQAR A-C leg in MSC mode.

When this option is specified, replication is not resumed on the A-B and B-D legs of the SQAR configuration, which retains the R21 and R22 devices at sites B and D as a Gold Copy of the data. Replication is resumed on the A-C leg of the SQAR, at which time production workloads may be restarted.

The full SQAR configuration may be restored by running script GDDRPA23 any time after script GDD2P18A has completed successfully.

---

**Note:** "Single-Side Restart" on page 533 provides additional information about this capability.

---

ALLOWHMC={Y|N}{Y|N}{Y|N}{Y|N]

The value is 4 characters where you specify 'Y' or 'N' for each character (for example, ALLOWHMC=NNYN).

The 4 bytes have the following meaning, in sequence:

- 1 — Allow HMC actions for the Primary DASD Site
- 2 — Allow HMC actions for the Secondary DASD Site
- 3 — Require WTOR confirmation for HMC Actions
- 4 — Allow LOAD CLEAR actions

Alternatively, you can specify these settings using the following separate keywords:

```
HMCPRIM={Y|N}
HMCSECN={Y|N}
HMCWTOR={Y|N}
HMCLOADC={Y|N}
```

ALTPARMS={Y|N}

Determines whether alternate IPL parameters are allowed (Y) or not (N) in HMC actions.

AUTO

Indicates that no further WTORs should be issued in this script submission process and the process should proceed as the parameters are currently set.

BCVGROUPS={Y|N}

Indicates whether multi-tenancy is to be used (Y) or not (N) for BCVs.

Specify Y to have the script identify BCV devices using the external data set named by the BCVGDEVS DD statement.

BGCLONE={Y|N}

Whether to use (Y) or not (N) BACKGROUNDCOPY for TimeFinder/Clone operations.

CALLOVER=<*call-overrides*>

Allows specification of a call override string, a 20-character string of 0 and 1.

If not specified, the value of GLOBAL.GDDR.GDDR.CALL_OVERRIDE is used. If that variable is not defined, the value 01111110000010011001 is used.

DRTPARMS={Y|N}

Determines whether to use (Y) or not (N) the Disaster Recovery Test (DRT) IPL parameters in HMC actions.

EXTBCV={Y|N}

Whether to use (Y) or not (N) external BCV devices.

LOAD=<*script-parm-file*>

Allows you to specify a file that contains the parameters for the SCRIPT command. Instead of specifying the parameters on the command manually, you can put them in a file and point the SCRIPT command at the file using the LOAD parameter.

*script-parm-file* can be either a sequential dataset (LOAD=*dataset*) or PDS member (LOAD=*dataset*(*member*)).

EXTDEV={Y|N}

Determines whether to use (Y) or not (N) external BCV devices.

PRISWAP=<*primary-swap-group*>

Specifies the name of the primary AutoSwap group.

RRSKIP={Y|N}

Indicates, on a script rerun, whether or not the failed step should be skipped or not. The value is YES or NO (any substring of YES or NO beginning with the first letter is allowed). No value implies YES. Default is NO.

SDDFCLN={Y|N}

Determines whether to perform SDDF cleanup. The value is YES or NO (any substring of YES or NO beginning with the first letter is allowed).

SECSWAP=<*secondary-swap-group*>

Specifies the name of the secondary AutoSwap group.

`SELSYS=<lpar>|(<lpar>,<lpar>,...)`

Specifies the names of the LPARs to be included in this execution of the script. SELSYS applies to scripts that execute test IPLs from R2 devices or BCVs, or the scripts that reset the environment to its normal state after a test IPL.

If BCVGROUPS is set to Y, then SELSYS is required.

`SITE=DC`*n*

Specifies the site name in the form of DC*n*; for example, DC1. Default is *.

`SUBMIT={Y|N}`

Determines whether to submit the JCL.

The value is YES or NO (any substring of YES or NO beginning with the first letter is allowed). Default is NO.

`SYSTEM=<system-name>`

Specifies the system name. Default is *.

`TESTBCV1={Y|N}`

Whether to use (Y) or not (N) the TEST-set BCVs at DC1.

`TESTBCV2={Y|N}`

Whether to use (Y) or not (N) the TEST-set BCVs at DC2.

`TESTBCV3={Y|N}`

Whether to use (Y) or not (N) the TEST-set BCVs at DC3.

`TESTBCV4={Y|N}`

Whether to use (Y) or not (N) the TEST-set BCVs at DC4.

`USER=<userid>`

Specifies the user ID to be set in the script job.

This parameter can have no value, implying no user ID is to be set in script job. Default is blank (no value).

`ZDPPAUSE={Y|N}`

Allows (Y) or prohibits (N, default) pausing of zDP processes during script execution.

`ZDPRESUME={Y|N}`

Allows (Y) or prohibits (N, default) resumption of zDP processes during script execution.

`ZDPUPLA={Y|N}`

Allows (Y, default) or prohibits (N) resumption of zDP processes for the VGDs at Site A during an unplanned script. This option is ignored when ZDPRESUME is set to N.

`ZDPUPLB={Y|N}`

Allows (Y, default) or prohibits (N) resumption of zDP processes for the VGDs at Site B during an unplanned script. This option is ignored when ZDPRESUME is set to N.

ZDPUPLC={Y|N}

Allows (Y, default) or prohibits (N) resumption of zDP processes for the VGDs at Site C during an unplanned script. This option is ignored when ZDPRESUME is set to N.

ZDPUPLD={Y|N}

Allows (Y, default) or prohibits (N) resumption of zDP processes for the VGDs at Site D during an unplanned script. This option is ignored when ZDPRESUME is set to N.

## Examples

The following examples show a non-existent script ID GDDRPA99 for demonstration purposes. User input is in bold.

**Note:** See "SCRIPT command process flow" on page 225 for an explanation of the SCRIPT command workflow.

*Example 1*

```
F GDDRMAIN,SCRIPT,GDDRPA99,JCLSUB=USERID1.WORK.CNTL(PA21)
GDDM251I Script command work submitted
GDDM266I Script GDDRPA99 submitted, JOB GDDRPA21 (J1234567)
GDDM267I GDDR Script command for GDDRPA99 ended
```

*Example 2*

```
F GDDRMAIN,SCRIPT,GDDRPA99
GDDM251I Script command work submitted
GDDM252A Enter script parameters
SITE=DC1,CALLOVER=01010101010101010101,SUBMIT=YES,USER=USERID1
GDDM252A Enter script parameters
JCLSAVE=USERID1.WORK.CNTL(PA21),NONSTAR=YES,ALLOWHMC=YYN,END
GDDM253I Script GDDRPA99 Parameters
   SITE=DC1
   SYSTEM=*
   PRISWAP=SWAP1
   SECSWAP=SWAP2
   CALLOVER=01010101010101010101
   USER=USERID1
   SDDFCLN=NO
   NONSTAR=YES
   ALLOWHMC=YYN
   RRSKIP=NO
   SUBMIT=YES
   JCLSAVE=USERID1.WORK.CNTL(PA21)
GDDM255A Confirm script generation (Y/N/CANCEL)
Y
GDDM266I Script GDDRPA99 submitted, JOB GDDRPA99 (J1234567)
GDDM267I GDDR Script command for GDDRPA99 ended
```

*Example 3*

```
F GDDRMAIN,SCRIPT,GDDRPA99
GDDM251I Script command work submitted
GDDM252A Enter script parameters
LOAD=USERID1.WORK.CNTL(PA21),SITE=DC2,END
GDDM253I Script GDDRPA99 Parameters
   SITE=DC2
   SYSTEM=*
   PRISWAP=SWAP1
   SECSWAP=SWAP2
   CALLOVER=01010101010101010101
```

```
   USER=USERID1
   SDDFCLN=NO
   NONSTAR=YES
   ALLOWHMC=YYN
   RRSKIP=NO
   SUBMIT=YES
GDDM255A Confirm script generation (Y/N/CANCEL)
Y
GDDM266I Script GDDRPA99 submitted, JOB GDDRPA21 (J1234567)
GDDM267I GDDR Script command for GDDRPA99 ended
```

*Example 4*

```
F GDDRMAIN,SCRIPT,GDDRPA99,LOAD=USERID1.WORK.CNTL(PA21),SITE=DC2,AUTO
GDDM251I Script command work submitted
GDDM253I Script GDDRPA99 Parameters
   SITE=DC2
   SYSTEM=*
   PRISWAP=SWAP1
   SECSWAP=SWAP2
   CALLOVER=01010101010101010101
   USER=USERID1
   SDDFCLN=NO
   NONSTAR=YES
   ALLOWHMC=YYN
   RRSKIP=NO
   SUBMIT=YES
GDDM266I Script GDDRPA99 submitted, JOB GDDRPA21 (J1234567)
GDDM267I GDDR Script command for GDDRPA99 ended
```

*Example 5*

```
F GDDRMAIN,SCRIPT,GDDRPA99
GDDM251I Script command work submitted
GDDM252A Enter script parameters
SITE=DC1,CALLOVER=01010101010101010101,SUBMIT=YES,USER=USERID1
GDDM252A Enter script parameters
CANCEL
GDDM257I GDDR Script command for GDDRPA99 cancelled
```

# SET

Use the SET command to enable or disable GDDR debugging, tracing, and verbose messaging globally.

**Note:** Debugging and verbose messaging can also be controlled using GDDRMAIN EXEC parameters DeBuG and VERBose.

Requirements are as follows:

- GDDRMAIN must be active.

The SET command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,seT
[,DeBuG[={ON|OFF}]]
[,TRaCe[={ON|OFF}]]
[,VERBose[={ON|OFF}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
```

**Note:** All yes/no and on/off parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=ON'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={ON|OFF}]`

- ON — Enable debugging in GDDRMAIN.
- OFF — Disable debugging in GDDRMAIN.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

`TRaCe[={ON|OFF}]`

- ON — Enable tracing in GDDRMAIN.

- OFF — Disable tracing in GDDRMAIN.

`VERBose[={ON|OFF}]`

- ON — Enable verbose messaging in GDDRMAIN.
- OFF — Disable verbose messaging in GDDRMAIN.

**Note:** This setting can be modified for a specific command using the VERBose parameter on that command.

## Examples

```
SET DBG=ON
GDDM081I Debugging set to ON

SET DBG=OFF
GDDM081I Debugging set to OFF

SET VERB=ON
GDDM083I Verbose messaging set to ON

SET VERB=OFF
GDDM083I Verbose messaging set to OFF
```

## START

Use START to start a subtask which is not currently running.

### Syntax

```
F GDDRMAIN,START <subtask-name>
```

**Note:** Issue the TASKS command to see the names of the subtasks.

## STOP

Use STOP to stop a subtask which is currently running, including GDDR worker tasks.

**Note:** See "Stopping workers" on page 167 for directions on stopping GDDR workers.

STOP will post the subtask to end, but if it does not end, you can use CANCEL to stop it.

### Syntax

```
F GDDRMAIN,STOP <subtask-name> [,FORCE]
```

### Parameters

`FORCE`

Forces shut down of the subtask.

**⚠WARNING**

**Use under direction of Dell EMC Customer Support.**

*`subtask-name`*

Specify the subtask name. Issue the TASKS command to see the names of the subtasks.

For GDDR workers whose task names are appended with numbers (*nn*), you can specify a worker type instead of a specific worker name to stop all workers of that type. For example, specify **`GDDWXR`** to stop all REXX workers (GDDWXR00, GDDWXR01, GDDWXR02, and so on), or **`GDDWXR02`** to stop REXX worker GDDWXR02 only.

# SUBSYS

Use SUBSYS to display active and inactive GDDR subsystems on the system where the command is run.

Requirements are as follows:

- At least one GDDR instance must have been started since the last IPL on the system where the command is run.

The SUBSYS command can also be issued from the Perform Health Check panel (C), or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,SubSys
[,ACTive[={Yes|No}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`ACTive[={Yes|No}]`

- Yes — Display only active GDDR subsystems.
- No (**default**) — Display both active and inactive GDDR subsystems.

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

```
SSYS
GDDUG28I GDDR subsystem information
 ------------------------------------------------------------------------
  Ssys Type Act Mul Com Maint level    Config type     Licenses
  ---- ---- --- --- --- -------------- --------------- ---------------
  GDD1 Csys  Y   Y    3 V5.2.0 BASE    Con Star-HA CAX GDDR
  GDD2 Csys  Y   Y    3 V5.2.0 BASE    SQAR            GDDR
 ------------------------------------------------------------------------
```

### Output fields

`Ssys` The GDDR subsystem name.

`Type` The system type:

- `Csys`—Control system (C-system)
- `Psys`—Production system (managed system)

`Act` Indicates whether the GDDR instance is active (Y) or not (N).

`Mul` Indicates if the GDDR C-System Multi-Tenancy feature is enabled (Y) or disabled (N).

**Note:** "GDDR C-system multi-tenancy" on page 51 describes this feature.

`Com` The compatibility level.

`Maint level` The GDDR maintenance level (version and highest PTF number).

`Config type` The configuration type.

**Note:** "GDDRMAIN supported configuration types" on page 255 lists possible configuration types.

`Licenses` Lists available GDDR licenses.

**Note:** See "Install GDDR Licensed Feature Code" on page 144 for information about licenses.

# SUMMARY

Use SUMMARY to display summary information for the system where the command is run, including the following details:

- System name
- Site
- GDDR version
- GDDR subsystem name
- SCF version
- SymmAPI version
- SCF subsystem name
- GDDR licenses

Requirements are as follows:

- SCF must be active.
- GDDRMAIN must be active.

The SUMMARY command is issued automatically on GDDRMAIN startup.

The SUMMARY command can also be issued from the Perform Health Check panel (C), or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,SUMmary
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.

- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

```
SUM
GDDUG31I GDDR Summary information
 ------------------------------------------------
  System               : X117 (master C-system)
  Site                 : DC1 (primary DASD site)
  GDDR version         : 5.2.0 BASE (02/01/18)
  GDDR subsystem name  : GDDR
  SCF version          : 8.3.0
  API version          : 8.3.0 (12/18/17)
  SCF subsystem name   : EMC
  Licenses             : GDDR
 ------------------------------------------------
```

### Output fields

| | |
|---|---|
| `System` | The system name and description. |
| `Site` | The site ID and description. |
| `GDDR version` | The GDDR version, maintenance level (highest PTF number), and build date. |
| `GDDR subsystem name` | The GDDR subsystem name. |
| `SCF version` | The SCF version and maintenance level (highest PTF number). |
| `API version` | The SCF API version, maintenance level (highest PTF number), and build date. |
| `SCF subsystem name` | The SCF subsystem name. |
| `Licenses` | Lists available GDDR licenses. |

**Note:** See "Install GDDR Licensed Feature Code" on page 144 for information about licenses.

# SVCDUMP

The SVCDUMP command issues an SVC dump for diagnostic use by Dell EMC.

The dump includes the GDDRMAIN address space, the address space from which the command was issued (if not GDDRMAIN), the GDDR data spaces, GDDR data spaces, the general purpose registers and access registers via PSWREGS, and SDATA = (ALLNUC, ALLPSA, CSA, GRSQ, LPA, LSQA, RGN, SQA, SUM, SWA, TRT). The dump title is "GDDR SVC Dump".

Requirements are as follows:

- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.

The SVCDUMP command can also be issued from the Perform Health Check panel (C), or using the GDDRMCMD batch interface. However, it is recommended to run this command as a GDDRMAIN modify command.

## Syntax

```
F GDDRMAIN,SVCDUMP
```

## Example

The following example shows the SVCDUMP command output:

```
SVCD
IEA794I SVC DUMP HAS CAPTURED:
DUMPID=006 REQUESTED BY JOB (GDDRMAIN)
DUMP TITLE=GDDR SVC Dump
```

# SYSTEMS

Use the SYSTEMS command to display all systems (both C-systems and production systems) defined in the GDDR-plex and their status.

**Note:** For more detailed system status information, use the CHECKUP command. CHECKUP SUMMARY produces a similar report to the SYSTEMS command but with more granular status information.

Requirements are as follows:

- GDDRMAIN must be active.

The SYSTEMS command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,SYStems
[,SITE={DC1|DC2|DC3|DC4}]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,SYStem=<system-name>]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SITE={DC1|DC2|DC3|DC4}`

Limits the report to systems at the specified site.

The site for a given C-system is defined using the CSYSSITE statement in the GDDRPARM file (as described in "CSYSSITE" on page 263). The site for a given production system is defined using the Define Managed Systems panel (M,P,H,S).

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG.

- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

`SYStem=<`*system-name*`>`

Limits the report to the specified system.

## Example

```
 SYS
GDDUG35I GDDR System information
------------------------------------------------------------------------
  T Site System   IP Address                              Port  Status
  - ---- -------- ---------------------------------------- ----- ------
  C DC1  DVTL1161 nnn.nnn.nnn.nnn                         9876  Avail
  C DC2 *GDDRDEVL nnn.nnn.nnn.nnn                         9876  Avail
  C DC3  DVTL149  nnn.nnn.nnn.nnn                         9876  Avail
  C DC4  DVTL150  nnn.nnn.nnn.nnn                         9876  Avail
  P DC1  CLAYB033 nnn.nnn.nnn.nnn                         9876  Err61
  P DC2  CLAYB034 nnn.nnn.nnn.nnn                         9876  Err61
  P DC3  CLAYB035 nnn.nnn.nnn.nnn                         9876  Err61
  P DC4  CLAYB036 nnn.nnn.nnn.nnn                         9876  Err61
------------------------------------------------------------------------
  *Master C-system
GDDUG48E GDDR 3 P-systems are inaccessible
```

### Output fields

`T` The system type:

- `C`—Control system (C-system)
- `P`—Production system (managed system)

`Site` The site ID.

`System` The system name. An asterisk (`*`) indicates the master C-system.

`IP Address` The IP address.

`Port` The port number.

`Status` The system status. The status consists of one of the following strings:

- Valid:
  - `Avail`—System is accessible.
  - `???`—Communication not yet attempted.
- System is not accessible, results in error message(s) and RC=8:
  - `NetDown`—TCP/IP CONNECT failed with error code 50 (network is down). Ensure the network path to the remote system is available.
  - `NoGDDR`—TCP/IP CONNECT failed with error code 61 (connection refused). Ensure GDDR is active on the remote system and available on the port specified in GDDRPARM.

- `SysDown`—TCP/IP CONNECT failed with error code 64 (host is down). Ensure the remote system is available.
- `Timeout`—TCP/IP CONNECT failed with error code 60 (connection timed out). Ensure the remote system is available and GDDR is active.
- `Unreach`—TCP/IP CONNECT failed with error code 49 (IP address is unreachable), 51 (network is unreachable), or 65 (host is unreachable). Ensure the remote system is available and GDDR is active.
- `Err<`*`tcpip-error-code`*`>`—Other TCP/IP CONNECT error occurred.

## TASKS

Use TASKS to display the status of each subtask.

For each subtask, a line is displayed with the subtask name and its status, ACTIVE or ENDED. If the status is ENDED, the last return code from the subtask is displayed in parentheses after ENDED.

### Syntax

```
F GDDRMAIN,TASKS
```

There are no operands.

# TOPOLOGY

Use the TOPOLOGY command to display all storage systems (both local and remote) discovered by GDDR on the system where the command is run, their SRDF groups, and the SRDF groups and storage systems on the other side.

Requirements and restrictions are as follows:

- GDDRMAIN must be active.
- Supported only on C-systems.

The TOPOLOGY command is issued automatically on GDDRMAIN startup.

The TOPOLOGY command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,TOPOlogy
[,SITE={DC1|DC2|DC3|DC4}]
[,ConTRoLler=<serial#>]
[,ALL[={Yes|No}]]
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`ALL[={Yes|No}]`

- Yes — Show all SRDF groups on GDDR-managed storage systems, including those connecting to storage systems not managed by GDDR.
- No (**default**) — Show only SRDF groups connecting GDDR-managed storage systems.

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`ConTRoLler=<serial#>`

Limits the scope of the command to the specified storage system and its SRDF groups.

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SITE={DC1|DC2|DC3|DC4}`

Limits the scope of the command to storage systems at the specified site and their SRDF groups.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

```
TOPO
GDDUG34I GDDR Discovered topology
------------------------------------------------------------------------
 Site   Controller    uCode          ____________Other side____________
 ---- -------------  --------  RAG RAG Site   Controller    uCode
                                --- --- ---- -------------  --------
 DC1  0001957-00794  5876.309
                                01  01  DC2  0001957-00987  5876.309
                                09  09  DC2  0001957-00987  5876.309
...
 DC1  0001977-00319  5978.110
                                03  03  DC2  0001977-00116  5977.125
                                D2  D2  DC3  0001926-00455  5876.309
 DC2  0001957-00987  5876.309
                                01  01  DC1  0001957-00794  5876.309
                                09  09  DC1  0001957-00794  5876.309
...
 DC2  0001977-00116  5977.125
                                03  03  DC1  0001977-00319  5978.110
                                D3  D3  DC4  0001977-00135  5977.125
 DC3  0001926-00455  5876.309
 DC3  0001957-00866  5876.309
                                02  02  DC1  0001957-00794  5876.309
                                12  12  DC1  0001957-00794  5876.309
...
 DC4  0001977-00135  5977.125
                                38  38  DC2  0001957-00987  5876.309
                                D3  D3  DC2  0001977-00116  5977.125
...
------------------------------------------------------------------------
```

### Output fields

| | |
|---|---|
| `Site` | The local site ID. |
| `Controller` | The local storage system serial number. |
| `uCode` | The version of the operating environment on the local storage system. |
| `RAG` | The SRDF group number on the local side. |
| `Other side RAG` | The SRDF group number on the remote side. |
| `Other side Site` | The remote site ID. |
| `Other side Controller` | The remote storage system serial number. |
| `Other side uCode` | The version of the operating environment on the remote storage system. |

# TRACE RESET

Use the TRACE RESET command to reset the GDDR trace by initializing pointers in GDDR data space 2 to the beginning of the trace buffer.

Requirements are as follows:

- GDDRMAIN must be active.
- GDDRMCMD must be APF-authorized.

The TRACE RESET command can also be issued as a Perform Health Check panel (C) primary command, or using the GDDRMCMD batch interface.

## Syntax

```
F GDDRMAIN,TRaCe RESet
[,CONsole[={Yes|No}]]
[,SysMSG[={Yes|No}]]
[,SysPRinT[={Yes|No}]]
[,DeBuG[={Yes|No}]]
```

**Note:** All yes/no parameters can be specified as either *keyword* or *keyword=value*. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

## Parameters

`CONsole[={Yes|No}]`

- Yes — Write body of report to console (default for console).
- No — Do not write report to console (default for batch).

`DeBuG[={Yes|No}]`

- Yes — Enable debugging to SYSPRINT DD.
- No (**default**) — Disable debugging.

`SysMSG[={Yes|No}]`

- Yes — Write body of report to JESYSMSG
- No (**default**) — Do not write report to JESYSMSG.

`SysPRinT[={Yes|No}]`

- Yes — Write report to SYSPRINT (default for batch).
- No — Do not write report to SYSPRINT (default for console).

## Example

```
TRC RES
GDDM089I GDDR Trace buffer pointers reset (00D968C0/80D968DC ->
   00001000/00001000)
```

# UMA

Use the UMA command to control user message automation.

When issued without parameters, the UMA command displays current user message automation status (ON or OFF).

**Note:** The MSG ADD|MODIFY statement enables or disables processing of specific messages, as described in "MSG ADD|MODIFY" on page 268.

## Syntax

```
F GDDRMAIN,UMA[,{ON|OFF|RELOAD}]
```

## Parameters

`OFF`

Turn off user message automation (do not allow processing of user-defined actions).

`ON`

Turn on user message automation (allow processing of user-defined actions).

`RELOAD`

Turn on user message automation and restart the MCSOPER subtask of GDDRMAIN with indication that the message table should be reloaded and MSG parameters should be processed. Note that this reloads all messages, not just user-defined messages.

# WORKER

Use the WORKER command to view worker subtasks, dynamically change the minimum and maximum number of worker tasks of a particular type, or disable a worker task.

**Note:** "Worker task management" on page 165 discusses GDDR workers.

## Syntax

```
F GDDRMAIN,WORKER[,<worker-name>]
F GDDRMAIN,WORKER,<worker-name>,{MIN=<min>|MAX=<max>}
```

## Parameters

`MIN=<min>|MAX=<max>`

Sets the minimum or maximum number of the specified worker task names to start.

**Note:** See "MIN and MAX limits" on page 166 for information about the MIN and MAX limits of worker tasks.

Both *min* and *max* are three-digit values.

If you specify a MIN limit higher than the current MAX limit, the MAX limit will automatically be increased to match the new MIN limit. If you specify a MAX limit lower than the current MIN limit, the MIN limit will automatically be reduced to match the new MAX limit.

You can specify either a MIN limit or a MAX limit or both on a single WORKER command. If you specify both, the MIN limit cannot be higher than the MAX limit or the command will fail.

The MAX value of 0 (zero) shut downs any matching workers and disables the worker type. Before using this feature, see "Disabling workers" on page 168 for usage information and warnings.

*worker-name*

Specifies the worker type name.

**Note:** Table 18 on page 165 lists valid names.

## Examples

- To view worker status:

```
F GDDRMAIN,WORKER
GDDM128I Worker Status
 GDDWCO Type S, Act 1, Min 1, Max 1
  Name GDDWCO00, Status Waiting
 GDDWCM Type S, Act 1, Min 1, Max 1
  Name GDDWCM00, Status Waiting
 GDDWST Type S, Act 1, Min 1, Max 1
  Name GDDWST00, Status Waiting
 GDDWXR Type A, Act 1, Min 1, Max 1
  Name GDDWXR00, Status Waiting
```

```
 GDDWGV Type S, Act 1, Min 1, Max 1
  Name GDDWGV00, Status Waiting
```

- To restrict the display to the GDDWST worker type:

```
F GDDRMAIN,WORKER,GDDWST
GDDM128I Worker Status
 GDDWST Type S, Act 1, Min 1, Max 1
  Name GDDWST00, Status Waiting
```

- To set the minimum limit of 2 for the GDDWST worker type:

```
F GDDRMAIN,WORKER,GDDWST,MIN=2
GDDM128I Worker Status
 GDDWST Type S, Act 1, Min 2, Max 2
  Name GDDWST00, Status Waiting
```

(Note that maximum value was also set to 2.)

- To set the maximum limit of 1 for the GDDWST worker type:

```
F GDDRMAIN,WORKER,GDDWST,MAX=1
GDDM128I Worker Status
 GDDWST Type S, Act 2, Min 1, Max 1
  Name GDDWST01, Status Waiting
  Name GDDWST00, Status Waiting
```

(Note that the minimum value was also set to 1.)

# GDDRMAIN startup command sequence

GDDRMAIN issues the following commands during startup, in their respective order:

1. MAINTENANCE (page 211)—issued in order to determine the GDDR maintenance level for the GDDRMAIN address space. GDDR maintenance information is registered with SYR on all storage systems that are defined to SCF. Message GDDM072I is displayed indicating the maintenance level for the GDDRMAIN address space, including the GDDR version, highest PTF number, latest build date, and cumulative maintenance level CRC. If GDDRMAIN had previously been running since the system was last IPLed and a change in maintenance level is detected, message GDDM073I is displayed indicating an upgrade or downgrade. In addition, the module-level maintenance report is written to the SYSPRINT DD in the GDDRMAIN job log with multi-line message GDDUG33I.
2. PARM_REFRESH (page 218)—issued to read the GDDRPARM file and perform various processing. The ECHO option is used to echo all GDDRPARM statements in the GDDRMAIN log with message GDDM600I. Any unlicensed or otherwise invalid GDDRPARM statements are flagged. Message GDDM604I indicates the active GDDR licenses, GDDM620I indicates the detected configuration type, GDDM621I indicates the configured sites, and GDDM622I indicates the configured MSC groups (if applicable).GDDRMAIN terminates if there are no active GDDR licenses, an unsupported combination of GDDRPARM statements is found, or if the GDDR instance is otherwise prohibited from running. The completion of PARM_REFRESH processing is indicated by message GDDM699I.
3. SUMMARY (page 237)—issued to display summary information in multi-line message GDDUG31I, including the local system name, site, GDDR version, GDDR subsystem name, SCF version, SymmAPI version, SCF subsystem name, and GDDR licenses.
4. REGION (page 220)—issued to display above and below the line REGION limits and usage information for the GDDRMAIN address space in multi-line message GDDUG29I.
5. CONFIG (page 194)—issued to validate and display all GDDR-managed SRDF groups (internal, external, and DLm) in multi-line message GDDUG37I. If any SRDF groups are found to be in an unexpected state, an error or warning message is displayed, as appropriate. Error message GDDUG83E indicates a problem during GDDRMAIN initialization; check SYMM statements in GDDRPARM. See previous error messages in GDDRMAIN log for more information.
6. TOPOLOGY (page 243)—issued to display all storage systems (both local and remote) discovered by GDDR on the local system, their SRDF groups, and the SRDF groups and storage systems on the other side, in multi-line message GDDUG34I. Error message GDDUG83E indicates a problem during GDDRMAIN initialization; check SYMM statements in GDDRPARM. See previous error messages in GDDRMAIN log for more information.
7. GATEK (page 201)—issued to validate and display all paths (both local and remote) to GDDR-managed storage systems that should be accessible on the local system in multi-line message GDDUG32I. If any gatekeepers are found to be unusable, an error message is displayed. If any gatekeepers are found to be online

or allocated, a warning message is displayed. Error message GDDUG05E indicates a problem during GDDRMAIN initialization; check SYMM statements in GDDRPARM. See previous error messages in GDDRMAIN log for more information.

8. CHECKUP (page 182)—issued to display system information in multi-line message GDDUC03I and validate the GDDR licenses, configuration type, subsystem name, version, maintenance level, and compatibility level across all systems in the GDDR-plex where GDDRMAIN is active. If a mismatch is detected, an error message is displayed. If a license, subsystem name, version, or compatibility level mismatch is detected, a WTOR is issued. If any systems are found to be inaccessible, a warning message is issued. Error message GDDUG85E indicates a problem during GDDRMAIN initialization; check COMM statements in GDDRPARM. See previous error messages in GDDRMAIN log for more information.

# GDDRMAIN validation checks

## Cross-address space validation of maintenance level

GDDR validates the maintenance level of various GDDR address spaces against that of the GDDRMAIN address space on that system.

**Note:** "GDDRMAIN dependent address spaces" on page 165 discusses GDDR address spaces.

For the GDDREVM, GDDRWORK, and GDDR script address spaces, this is done automatically upon startup. In addition, you can run the GDDRECHK utility (described in "GDDR Environment Check utility (GDDRECHK)" on page 442) to test the PROC used by GDDR scripts (GDDRPROC) and to ensure the maintenance level of any future GDDR script matches that of the GDDRMAIN address space.

For TSO users, when accessing the GDDR ISPF interface for the first time after logging on, the MAINTENANCE command (described in "MAINTENANCE" on page 211) is issued to determine the maintenance level of the TSO user address space.

### GDDRMAIN

On startup of GDDRMAIN, the MAINTENANCE command (described in "MAINTENANCE" on page 211) is issued to determine the GDDR maintenance level of the GDDRMAIN address space. GDDR maintenance information is registered with SYR on all systems that are defined to SCF, including the application name (GDDR), GDDR version, highest PTF number, latest build date, API version, and system name.

Message GDDM072I is displayed indicating the maintenance level for the GDDRMAIN address space, including the GDDR version, highest PTF number, latest build date, and cumulative maintenance level CRC, for example:

```
GDDM072I GDDR maintenance level: V5.2.0 GD52001 (08/04/18)  [CRC
   58C9F833]
```

If GDDRMAIN had previously been running since the system was last IPLed and a maintenance level upgrade is detected, message GDDM073I is displayed indicating an upgrade, for example:

```
GDDM073I GDDR maintenance level upgraded from V5.2.0 GD52015
   (02/20/18) to V5.2.0 GD52026 (04/16/18)
```

If GDDRMAIN had previously been running since the system was last IPLed and a maintenance level downgrade is detected, message GDDM073I is displayed indicating a downgrade, for example:

```
GDDM073I GDDR maintenance level downgraded from V5.2.0 GD52015
   (02/20/18) to V5.0.0 GD50026 (09/16/16)
```

In addition, the composite load module report is written to the SYSPRINT DD in the GDDRMAIN job log.

## GDDREVM, GDDRWORK, and GDDR scripts

On startup of GDDREVM, GDDRWORK, as well as GDDR scripts, the MAINTENANCE command (described in "MAINTENANCE" on page 211) is issued to determine the GDDR maintenance level for that address space, which is validated against that of the GDDRMAIN address space using a CRC representing the cumulative maintenance level.

Message GDDM072I is displayed indicating the GDDREVM, GDDRWORK, or GDDR script maintenance level, including the GDDR version, highest PTF number, latest build date, and cumulative maintenance level CRC, for example:

```
GDDM072I GDDR maintenance level: V5.2.0 GD52001 (08/04/18)  [CRC
   58C9F833]
```

If the GDDREVM, GDDRWORK, or GDDR script maintenance level matches that of the GDDRMAIN address space, message GDDM074I is displayed, for example:

```
GDDM074I GDDR maintenance level matches GDDRMAIN address space
```

If the GDDR version of a GDDREVM, GDDRWORK, or GDDR script address space does not match that of the GDDRMAIN address space, messages GDDM076E and GDDO77E are displayed indicating the mismatching maintenance levels, and the address space is terminated, for example:

```
GDDM076E GDDR Version mismatch - V5.0.0 GD50059 (05/24/17) vs.
   GDDRMAIN: V5.2.0 GD52001 (08/04/18)
GDDM077E **** Results are unpredictable, resolve maintenance level
   mismatch and retry ****
GDDM012I EVM Shutdown complete
```

If the PTF level of a GDDREVM, GDDRWORK, or GDDR script address space does not match that of the GDDRMAIN address space, messages GDDM075E and GDDM077E are displayed indicating the mismatching maintenance levels, and the affected address space terminates. For example:

```
GDDM075E GDDR maintenance level mismatch - V5.2.0 GD52001 (06/29/18)
   vs. GDDRMAIN: V5.2.0 BASE (06/28/18)
GDDM077E **** Results are unpredictable, resolve maintenance level
   mismatch and retry ****
GDDM012I EVM Shutdown complete
```

In addition, the composite load module report is written to the SYSPRINT DD in the GDDREVM, GDDRWORK, or GDDR script job log.

## GDDR environment check

The GDDRECHK utility described in "GDDR Environment Check utility (GDDRECHK)" on page 442 can be used to validate APF authorization, RACF definitions, as well as test the PROC used by GDDR scripts (GDDRPROC) to ensure the maintenance level of any future GDDR script matches that of the GDDRMAIN address space. The messaging is the same as described in "GDDREVM, GDDRWORK, and GDDR scripts" on page 252. The composite load module report is written to the SYSPRINT DD in the GDDRECHK job log. The GDDRPROC procedure should be checked to ensure 'SYSPRINT DD SYSOUT=*' is included.

### TSO users

When a TSO user first accesses the GDDR interface after logging on, the MAINTENANCE command (described in "MAINTENANCE" on page 211) is issued to determine the maintenance level of the TSO user address space. If the maintenance level of the TSO user address space (the version and PTF level) does not exactly match that of the GDDRMAIN address space for the selected GDDR instance, a popup panel is displayed. From this panel, the REFRESH command can be issued to re-determine the maintenance level for the TSO user address space. If the GDDR version does not ultimately match, access to that GDDR instance is prohibited.

The maintenance level of a TSO user's address space can be validated against that of the GDDRMAIN address space at any time using the MAINTENANCE primary command on the Perform Health Check panel (C) described in "Perform GDDR health check (C)" on page 366. The messaging is the same as described in "GDDREVM, GDDRWORK, and GDDR scripts" on page 252.

## Cross-system validation

On startup of GDDRMAIN on each system, after the MAINTENANCE command (described in "MAINTENANCE" on page 211) is issued to determine the maintenance level for the GDDRMAIN address space, this maintenance level, along with the GDDR licenses, configuration type, subsystem name, version, and compatibility level, is validated across all systems in the GDDR-plex where GDDRMAIN is active. This is done using the CHECKUP command described in "CHECKUP" on page 182.

If a subsystem name, license, version, or compatibility level mismatch is detected, a WTOR is issued, for example:

```
GDDUC17E Compatibility level mismatch detected
*01 GDDM700A Compatibility level mismatch detected (reply CONTinue,
   Retry, or CANcel)
```

If the local system is in error, the operator can reply CANcel to terminate GDDRMAIN and resolve the mismatch before restarting, for example:

```
R 01,CAN
GDDM703I Operator replied CANcel - terminating
GDDM011I GDDRMAIN Shutdown beginning
```

If a remote system is in error, the operator can resolve the mismatch on that system and then reply Retry to redrive cross-system validation after the issue has been resolved, for example:

```
R 01,R
GDDM702I Operator replied Retry - redriving cross-system validation
GDDUC03I System status
...
GDDM010I GDDRMAIN Initialization complete
```

The operator can also reply CONTinue to proceed with GDDRMAIN initialization (not recommended), for example:

```
R 01,CONT
GDDM701I Operator replied CONTinue - proceeding
GDDM010I GDDRMAIN Initialization complete
```

## Validation of multi-instance environment

A P-instance cannot run concurrently with any other GDDR instance on the same z/OS system. If a P-instance attempts to start and one or more GDDR instances are already active on that z/OS system, error message GDDM290E is issued, and GDDRMAIN terminates, for example:

```
GDDM290E Multiple GDDR subsystems not allowed on managed systems
```

If a C-instance attempts to start and a P-instance is already active on that z/OS system, error message GDDM613E is issued, and GDDRMAIN terminates, for example:

```
GDDM613E P-system GDD1 is active on X117, multi-GDDR not supported
```

A C-instance that is not enabled for GDDR C-System Multi-Tenancy cannot run concurrently with any other GDDR instance on the same z/OS system. If a C-instance attempts to start and another C-instance that is not enabled for GDDR C-System Multi-Tenancy is already active on that z/OS system, error message GDDM614E is issued, and GDDRMAIN terminates, for example:

```
GDDM614E Multi-GDDR is disabled for subsystem GDD1 active on X117
```

If one or more C-instances that are enabled for GDDR C-System Multi-Tenancy are active and another C-instance that is not enabled for GDDR C-System Multi-Tenancy attempts to start on that z/OS system, error message GDDM611E is issued, and GDDRMAIN terminates, for example:

```
GDDM611E Multi-GDDR is disabled for subsystem GDD3
 Other GDDR instances active on X117:
 - GDD1
 - GDD2
```

If a C-instance that is enabled for GDDR C-System Multi-Tenancy is started and any other GDDR instances active on that z/OS system are also C-instances with GDDR C-System Multi-Tenancy enabled, informational message GDDM610I is issued, for example:

```
GDDM610I Multi-GDDR is enabled for subsystem GDD3
 Other GDDR instances active on X117:
 - GDD1
 - GDD2
```

If a C-instance that is enabled for GDDR C-System Multi-Tenancy is started and no other GDDR instance is active on that z/OS system, message GDDM610I shows 'None', for example:

```
GDDM610I Multi-GDDR is enabled for subsystem GDD1
 Other GDDR instances active on X117: None
```

If a GDDR instance that is not enabled for GDDR C-System Multi-Tenancy (for example, a C-instance without the MULTI ENABLE statement, or a P-instance) is started and no other GDDR instance is active on that z/OS system, no messages related to GDDR C-System Multi-Tenancy are issued.

# GDDRMAIN supported configuration types

Table 28 lists GDDR configuration types that can be displayed in various GDDRMAIN reports:

**Table 28** GDDRMAIN supported configuration types

| Short name(s) | Description |
|---|---|
| `2-Site /A+Tape (MSC)`<br>`SRDF/A+Tape (MSC)` | 2-site SRDF/A with MSC + Tape |
| `2-Site /S+Tape (CAX)`<br>`SRDF/S+Tape (CAX)` | 2-site SRDF/S with AutoSwap + Tape |
| `2-Site /S+Tape (CG)`<br>`SRDF/S+Tape (CG)` | 2-site SRDF/S with ConGroup + Tape |
| `2-Site Con Star`<br>`Con Star-2` | 2-site concurrent SRDF/Star |
| `2-Site SRDF/A (MSC)`<br>`SRDF/A (MSC)` | 2-site SRDF/A with MSC |
| `2-Site SRDF/S (CAX)`<br>`SRDF/S (CAX)` | 2-site SRDF/S with AutoSwap |
| `2-Site SRDF/S (CG)`<br>`SRDF/S (CG)` | 2-site SRDF/S with ConGroup |
| `Cas Star (CAX)` | Cascaded SRDF/Star with AutoSwap |
| `Cas Star (CG)` | Cascaded SRDF/Star with ConGroup |
| `Cas Star-A` | Cascaded SRDF/Star-A (asynchronous Star) |
| `Cas Star-HA (CAX)` | Cascaded SRDF/Star with High Availability and AutoSwap |
| `Con Star (CAX)` | Concurrent SRDF/Star with AutoSwap |
| `Con Star (CG)` | Concurrent SRDF/Star with ConGroup |
| `Con Star (CG)` | Concurrent SRDF/Star with ConGroup |
| `Con Star-A` | Concurrent SRDF/Star-A (asynchronous Star) |
| `Con Star-HA (CAX)`<br>`Con Star-HA CAX` | Concurrent SRDF/Star with High Availability and AutoSwap |
| `SQAR` | SRDF/SQAR |

# GDDRPARM statements

## Overview

GDDRPARM statements, also referred to as *GDDRMAIN parameters*, are defined in the GDDRPARM file and propagated to be available to C-systems and managed systems during GDDR installation.

"Customize GDDRMAIN parameters" on page 145 provides the procedure to customize GDDRPARM statements and verify GDDRPARM file consistency across all systems.

Table 29 summarizes GDDRPARM statements.

**Table 29** GDDRPARM statements

| Statement | Description and syntax | Page |
|---|---|---|
| CMDQMAXT | Specify maximum number of tasks GDDR will load on the SRDF Host Component command queue<br>`CMDQMAXT <system> <subsystem> <max-tasks>` | |
| COMM | Specify IP addresses and ports for GDDRMAIN communication<br>`COMM <system> <subsystem> <ip-addr>,<port>[,<recv-ip-addr>]` | 259 |
| CPC | Specify fully qualified CPC name in format *netid.nau* for a C-system with HMC connectivity<br>`CPC <system> <subsystem> <site>,<cpc-name>,NETID` | 261 |
| | Specify HMC interface (e.g., BCPii) for a C-system with HMC connectivity<br>`CPC <system> <subsystem> <site>,<cpc-name>,<hmc-interface>` | |
| CSYSSITE | Specify site location for C-system<br>`CSYSSITE <system> <subsystem> <site>` | 263 |
| DRTCOMM | Allow communication with an LPAR that is participating in a disaster recovery (DR) test<br>`DRTCOMM <system> <subsystem> <site>,<ip-addr>,<port> [,<recv-ip-addr>]` | 264 |
| GVDIVDSN | Define global variable data-in-virtual (DIV) dataset name<br>`GVDIVDSN <system> <subsystem> <div-dsn>` | 265 |
| MSCGROUP | Defines MSC group name for a site pair<br>`MSCGROUP DCn->DCm <subsystem> <mscgrp>` | 266 |
| MSG \| MESSAGE | Enable or disable message interception for specified message<br>`MSG <system> <subsystem> <msgid>,{ENABLE\|DISABLE}`<br>Define message automation rules<br>`MSG <system> <subsystem> <msgid>,{ADD\|MODify}[,<actions>]` | 267 |
| MULTI | Enables or disables the GDDR C-System Multi-Tenancy feature<br>`MULTI <site-or-system> <subsystem> {ENABLE\|DISABLE}` | 271 |
| SITE | Define a configured site and its region<br>`SITE <site> <subsystem> REGION({1\|2\|3})` | 272 |
| SYMM | Define storage systems to GDDRMAIN<br>`SYMM <site> <subsystem> <symm-serial#>,<gatekeepers>` | 273 |
| VCPC | Allow test lab with a single CPC to fake multiple CPCs | 275 |

**Table 29** GDDRPARM statements

| Statement | Description and syntax | Page |
|---|---|---|
| | `VCPC <system> <subsystem> <site>,<virtual-cpc-name>,<real-cpc-name>` | |
| WORKER | Set minimum and maximum number of worker tasks for a given worker type<br>Disable a worker type<br>`WORKER <system> <subsystem> <worker-name> <min> <max>` | 276 |

# CMDQMAXT

Specifies the maximum number of tasks GDDR can load on the SRDF Host Component command queue.

## Syntax

```
CMDQMAXT <system> <subsystem> <max-tasks>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | CMDQMAXT | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME= *system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *max-tasks* | Th maximum number of tasks. |

# COMM

GDDR inter-system communication is defined via COMM statements in the GDDRPARM file. The COMM parameters specify the IP addresses and ports for GDDRMAIN to use.

You can specify two IP addresses per statement and multiple COMM statements per system. This allows GDDRMAIN to communicate with LPARs that are not at their designated location.

Having multiple COMM statements for a system also allows for situations where the same system may present different IP addresses to other systems based on firewalls or other network considerations. For example, system A may appear to have IP 1.2.3.4 to system B, but may appear to have IP 5.6.7.8 to system C. Communication with systems not at their designated location is also a consideration of the DRTCOMM parameter statement.

The address used is based on incoming communication. When a message is received, it is matched to a "receive from" address of a system. If that is not the current "receive from" system address, communication with that system is switched to use this "receive from" address and the associated "send to" address. Thus IP address selection is dynamic and requires no intervention.

When the COMM subtask initializes, it sends a null message to each "send to" address of every other system. When received, this message causes the receiving system to send back a different null message, thus informing the original system of what IP address is in use for the system. If the IP address changes while the COMM subtask is up, it will recognize the change and update its control blocks when it receives a message from the system which changed. Communication can always be forced via operator command (MPARM or BC/BR commands).

---

**Note:** If you need to change any COMM parameters, see the instructions in "Changing C-system or managed system IP port" on page 578.

---

## Syntax

```
COMM      <system> <subsystem> <ip-addr>,<port>[,<recv-ip-addr>]
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | COMM | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME= *system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *ip-address* | The “send to” address. If *recv-ip-address* is not provided, this address is used for both send and receive. |
| | *,port* | The port number. You can select any port number from the site's unregistered ports, but the selected port number must be the same for all systems. |
| | *,recv-ip-address* | The “receive from” address (optional). |

## Example

```
COMM      DVTL1161 GDDR nnn.nnn.nnn.nnn,9876
```

# CPC

The CPC parameter specifies the fully qualified CPC name in the format *netid.nau* for a C-system with HMC connectivity. It also specifies the HMC interface (BCPii) to be used for the specified GDDR-managed CPCs and the GDDR-managed objects on those CPCs.

There are two types of CPC parameters:

- NETID parameters define the CPC name for the C-systems in the configuration.
- METHOD parameters are required if you want to configure GDDR for BCPii support for your systems. The METHOD parameters define a B-system for each CPC where either a C-system or a managed system can run.

## Syntax

```
CPC      <system> <subsystem> <site>,<cpc-name>,NETID
CPC      <system> <subsystem> <site>,<cpc-name>,<hmc-interface>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | CPC | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is a C-system for the NETID parameters, and by definition, a B-system for the METHOD parameters. The B-system has BCPii connectivity to the named CPC. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24-26 | *site* | This site is the physical location of the named CPC: DC1, DC2, DC3, or DC4. |
| 27 | , | |
| 28- | | CPC name. The full CPC name in the format *netid.nau*: *netid* (1-8 characters), *nau* (1-8 characters), separated by a '.'. The full CPC name is followed by a comma.<br>BCPII HMC Interface, literal 'BCPII'.<br>Add at least one CPC parameter for each CPC where C-systems or managed systems can run. |

## Example

```
CPC      LB01M3A  GDDR DC1,nnnnnnnn.nnnnn,NETID
CPC      LB01K157 GDDR DC2,nnnnnnnn.nnnnn,NETID
CPC      LB01M34  GDDR DC3,nnnnnnnn.nnnnn,NETID
CPC      LB01K148 GDDR DC4,nnnnnnnn.nnnnn,NETID
```

# CSYSSITE

CSYSSITE parameters define the system names of the C-systems and the site where they are located.

## Syntax

```
CSYSSITE <system> <subsystem> <site>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | CSYSSITE | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *site* | The site name. |

## Example

```
CSYSSITE SYS1     GDDR DC1
```

# DRTCOMM

The DRTCOMM statement allows communication with an LPAR that is participating in a disaster recovery (DR) test.

At least one COMM statement for the system must precede the DRTCOMM statement.

## Syntax

```
DRTCOMM  <system> <subsystem> <site>,<ip-addr>,<port>[,<recv-ip-addr>]
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | DRTCOMM | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *site* | The site at which this system would be running during a DR test. |
| | *,ip-address* | The “send to” address. If *recv-ip-address* is not provided, this address is used for both send and receive. |
| | *,port* | The port number.<br>You can select any port number from the site's unregistered ports, but the selected port number must be the same for all systems. |
| | *,recv-ip-address* | The “receive from” address (optional). |

## Example

```
DRTCOMM      DVTL149  GDDR nnn.nnn.nnn.nnn,9876
```

# GVDIVDSN

The GVDIVDSN statement conveys the names of DIV datasets to GDDRMAIN. The DIV datasets serve as a permanent store of GDDR global variables.

**Note:** You define DIV datasets during the procedure described in "Define global variable datasets" on page 139.

## Syntax

```
GVDIVDSN <system> <subsystem> <div-dsn>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | GVDIVDSN | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *div-dsn* | The left-justified name of the DIV dataset. |

## Example

```
GVDIVDSN SYS1     GDDR hlq.GDDRvrm.SYS1.DIV
GVDIVDSN SYS2     GDDR hlq.GDDRvrm.SYS2.DIV
GVDIVDSN SYS3     GDDR hlq.GDDRvrm.SYS3.DIV
```

# MSCGROUP

The MSCGROUP statement defines the MSC group name for a given site pair.

MSCGROUP parameters specify the MSCGROUP name used to control included SRDF/A sessions on the indicated site pair, in the indicated direction.

MSCGROUP names are tied exclusively to the SRDF/A active site pair and direction, and are unrelated to the active MSC mode or topology.

GDDR requires MSCGROUP parameters for DC1->DC3, DC3->DC1, DC2-DC4, and DC4-DC2.

GDDR requires unique MSCGROUP names for each site pair and each direction.

## Syntax

`MSCGROUP DC`*`n`*`->DC`*`m`* `<`*`subsystem`*`> <`*`mscgrp`*`>`

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | MSCGROUP | |
| 10-17 | DC*n*->DC*m* | The SRDF/A source site and the SRDF/A target site.. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *mscgrp* | The MSC group name. |

# MSG ENABLE|DISABLE

The MSG ENABLE|DISABLE statement enables or disables processing of specific messages.

This statement is processed when GDDRMAIN is restarted, thus keeping the desired enable/disable state over GDDRMAIN restarts.

## Syntax

```
MSG <system> <subsystem> <msgid>,{ENABLE|DISABLE}
```

**Note:** MESSAGE is an alias of MSG. ENABLE and DISABLE may be abbreviated to any length. For example, ENABLE can be specified as E, EN, or ENA.

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | MSG | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running.<br>Asterisk (*) indicates all systems. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *msgid* | The ID of the message to be processed. |
| | ,ENABLE<br>\|DISABLE | Enables or disables processing of the message specified with *msgid*. |

## Example

For example, you may want to enable processing of messages GDDX191I and GDDX291I to trigger action on your production systems (these messages are disabled by default):

```
MESSAGE  *        GDDR GDDX191I,ENABLE
MESSAGE  *        GDDR GDDX291I,ENABLE
```

# MSG ADD|MODIFY

The MSG ADD|MODIFY statement enables or disables processing of specific messages.

This statement is processed when GDDRMAIN is restarted, thus keeping the desired enable/disable state over GDDRMAIN restarts.

## Syntax

```
MSG <system> <subsystem> <msgid>,{ADD|MODify}[,actions]
```

**Note:** MESSAGE is an alias of MSG. ADD and MODIFY may be abbreviated to any length (for example, MODIFY can be specified as M, MO, or MOD).

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | MSG | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running.<br>Asterisk (*) indicates all systems. |
| 19-22 | *subsystem* | The 4-character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *msgid* | The ID of the message to be processed. |
| | ,ADD \|MODify | ADD indicates the message is being added to the table of messages that GDDRMAIN intercepts.<br>MODIFY indicates the message is already defined (either already present in the GDDRMAIN-supplied table or added by a previous ADD statement). |
| | ,*actions* | See "MODIFY actions" on page 268. |

## MODIFY actions

Available MODIFY *actions* are as follows. If you need more actions than will fit on one statement, you can use additional MODIFY statements to add more.

ACTION=REPEAT

Repeat the message. This is used in conjunction with FORWARD. The message is repeated on the system it is forwarded to. This may only be specified for user-defined messages.

ACTION=REXX=*name*

Execute the named REXX module. The REXX is taken from the GDDREXEC DD of the GDDRWORK proc. This action may be specified for both user-defined and GDDR-defined messages. If specified for a GDDR-defined message, the action is taken after all GDDR-defined actions have been taken.

FORWARD=TOCSYS | TOPSYS | TOMAST | NONE

Specifies any forwarding of the message to other systems for processing. This may only be specified for user-defined messages.

FORWARD and PROCESS statements must be consistent with each other. For example, you cannot forward a message to production systems and specify process on C-systems. PROCESS=LOCAL is not allowed with any value of FORWARD except NONE.

NONE

Do not forward the message. This is the default unless PROCESS is specified and implies a value above.

TOCSYS

If the message occurs on a production system, it will be forwarded to every C-system, including the master. If the message occurs on a C-system, it will not be forwarded.

TOMAST

If the message occurs on a C-system which is not the master, it will be forwarded to the master C-system. If the message occurs on a production system, it will be forwarded to every C-system, including the master.If the message occurs on the master C-system, it will not be forwarded.

TOPSYS

If the message occurs on a C-system or master C-system, it will be forwarded to every production system. If the message occurs on a production system, it will not be forwarded.

PROCESS=ONCSYS | ONPSYS | MASTONLY | LOCAL

Specifies where the message should be processed. This may only be specified for user-defined messages.

FORWARD and PROCESS must be consistent with each other. For example, you cannot forward a message to production systems and specify process on C-systems.

Consistency requirements are as follows:

- FORWARD=TOPSYS requires PROCESS=ONPSYS.
- FORWARD=TOCSYS requires PROCESS=ONCSYS.
- FORWARD=TOMAST requires PROCESS=ONCSYS or PROCESS=MASTONLY.

PROCESS=LOCAL is not allowed with any value of FORWARD except NONE.

LOCAL

Process the message wherever it occurs. This is the default if PROCESS is not specified or is implied by FORWARD.

MASTONLY

The message can be processed only on the master C-system.

ONCSYS

The message can be processed on any C-system.

ONPSYS

The message can be processed on any production system.

# MULTI

Enables or disables the GDDR C-System Multi-Tenancy feature described in "GDDR C-system multi-tenancy" on page 51.

MULTI statements are processed at GDDRMAIN startup and when the F GDDRMAIN,PARM_REFRESH command is issued.

## Syntax

```
MULTI <site-or-system> <subsystem> {ENABLE|DISABLE}
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | MULTI | |
| 10-17 | *site-or-system* | The site ID or the system name. An asterisk (*) matches all sites or systems. |
| 19-22 | *subsystem* | The 4-character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | ENABLE\|DISABLE | Indicates whether to enable or disable C-system multi-tenancy.<br>A MULTI statement with a value of DISABLE is equivalent to having no MULTI statement for that GDDR subsystem at all. |

## Example

To enable C-system multi-tenancy for subsystem GDD1 on all C-systems:

```
MULTI     *         GDD1 ENABLE
```

# SITE

The SITE statement defines a configured site and its region to GDDRMAIN.

## Syntax

```
SITE <site> <subsystem> REGION({1|2|3})
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | SITE | |
| 10-17 | *site* | The site ID. |
| 19-22 | *subsystem* | The 4-character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | REGION({1\|2\|3}) | The site region. Possible values are 1, 2, or 3. |

# SYMM

SYMM parameters define PowerMax/VMAX storage systems to GDDRMAIN.

The same storage system can be listed at different sites if it is channel connected to each of the sites. The (H) indicates the site at which that storage system is physically located. Each storage system must have only one home site.

If the list of gatekeepers is so long that it will not fit on one statement, you can repeat the statement up to the gatekeeper list and put more gatekeepers on subsequent lines.

For example:

```
SYMM      DC1(H)   GDDR 012345678910,4202,3456-346F,1234,9876-9877
SYMM      DC1(H)   GDDR 012345678910,8202,8456-846F,8234,8876-8877
```

There must be at least one SYMM parameter for each storage system to be managed by this GDDR instance. This SYMM parameter defines the physical location (home site) for the storage system.

All GDDR command gatekeepers used on GDDR RDF.DEVICES, DLM.DEVICES, and STDBCV parameters as defined using the GDDR Parameter Wizard must first be defined on a SYMM parameter. Do not specify MSC SQARgatekeepers in the population of GDDR command gatekeepers.

**Note:** "Restoring normal operations after MSC THAW" on page 590 provides instructions on changing SYMM parameters.

## Syntax

```
SYMM <site>[H] <subsystem> <serial#>,<gatekeepers>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-4 | SYMM | |
| 10-12 | *site* | This is the site from where the listed gatekeepers provide access to the listed storage system: DC1, DC2, DC3, or DC4. |
| 13-15 | H | (H) optional, indicates that the listed site is the home site or physical location of the listed storage system. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24-35 | *serial#* | 12-digit PowerMax/VMAX system serial number. |
| 36 | , | |
| 37 | | A list of gatekeepers specified as individual unit addresses, ranges of unit addresses, or a combination; for example: *aaaa,bbbb-cccc,dddd*. |

## Example

```
SYMM      DC1(H)   GDDR 012345678910,4202
SYMM      DC2(H)   GDDR 012345678911,3456
SYMM      DC3(H)   GDDR 012345678912,8202
SYMM      DC4(H)   GDDR 012345678913,8202
```

# VCPC

Define virtual CPC, allowing test lab with single CPC to fake multiple CPCs.

**Note:** CPC names are in *netid.nau* format.

## Syntax

```
VCPC <system> <subsystem> <site>,<virtual-cpc-name>,<real-cpc-name>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | VCPC | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME= *system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running.<br>Asterisk (*) indicates all systems. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24- | *site* | The site name. |
| | *virtual-cpc-name* | The name of the virtual CPC. |
| | *real-cpc-name* | The name of the real CPC. |

# WORKER

The WORKER statements specify the names of the worker tasks to be initially started and the minimum and maximum number of worker tasks per system. You can also disable a worker type so that worker tasks of the disabled type are not initially started.

**Note:** See "Worker task management" on page 165 for information about GDDR workers.

It is recommended that you do not specify the MIN and MAX parameters in your GDDRPARM file. This maintains the minimum and maximum numbers for all workers at their default values listed in Table 19 on page 166, which should be appropriate for most configurations.

## Syntax

```
WORKER   <system> <subsystem> <worker-name> <min> <max>
```

Where:

| Columns | Value | Description |
|---|---|---|
| 1-8 | WORKER | |
| 10-17 | *system* | The 8-character (left-justified, uppercase) z/OS system name of a C-system which is specified using the SYSNAME= *system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. Note that this is *not* the SMF ID of the system (although it could be the same). The system named here is usually a C-system, but can be any system where GDDRMAIN is running.<br>Asterisk (*) indicates all systems. |
| 19-22 | *subsystem* | The 1-4 character left-justified name of the GDDR instance to which this applies. It must match *nnnn* as specified on the GDD$*nnnn* DD card in the startup JCL for this GDDRMAIN instance. |
| 24-31 | *worker-name* | The name of the worker task as listed in Table 18 on page 165. |
| 33-35 | *min* | The minimum number of the specified worker task names to start (blank for no change). This is a three-digit value. |
| 37-39 | *max* | The maximum number of the specified worker task names to start (blank for no change). This is a three-digit value that must be greater than or equal to the MIN value.<br>The MAX value of 000 disables the worker type. Before using this feature, see "Disabling workers" on page 168 for usage information and warnings. |

## Example

- To set the minimum and maximum limit of 1 for the GDDWST worker type:

```
WORKER   GDDRSYS1 GDDR GDDWST 1 1
```

- To disable the GDDWST worker type:

```
WORKER   SYS1     GDDR GDDWXR   000 000
```

# GDDR locks

Two locks are used when managing GDDR system variable integrity, the index lock and the update lock. Both locks enable you to record, without serialization, when you are using the index or doing an update, and allow exclusive control by a single user when no use is in progress. If either lock becomes "stuck", it can have negative implications for global variable access.

The lock consists of two words. The first word is the exclusive lock, the second word is the shared lock. If both words are zero, the lock is completely open. This is the normal state. The lock words can have the combinations of values listed in Table 30.

**Table 30** Possible lock states

| First word - exclusive value | Second word - shared value | State |
|---|---|---|
| 0 | 0 | The lock is open. There is no shared or exclusive use of the resource. |
| 0 | Positive value | The resource is in use by one or more non-exclusive users. The shared value is the number of users. No-one is requesting exclusive use. |
| 1 | Positive value | The resource is in use by one or more non-exclusive users, but one user is waiting for exclusive use. If the shared word does not go to zero within a very short period of time, the lock is stuck. This is known as a *shared lock stuck*. |
| 1 | 1 | The lock is locked, or set. One task has exclusive use of the resource and any other task which wants to use it must wait or take alternative action. If the exclusive word does not go to 0 within a very short period of time, the lock is stuck. This is known as an *exclusive lock stuck*. |
| Other than 0 or 1 | Any value | The lock is broken. The exclusive word should only have the value 0 or 1. |
| 0 or 1 | Negative value | The lock is broken. The shared word should only have non-negative values. |

To display the locks, run the GDDRGVX utility with the DSPLIST parameter, as described in "GDDR DIV Management utility (GDDRGVX)" on page 513, or issue the LOCK command described in "LOCK" on page 210.

You can clear a stuck or broken lock by using the LOCK command.

## Index lock

The index lock indicates that the global variable index is in use or is being rebuilt. If the index is being rebuilt, it cannot be used by anyone.

If the exclusive lock becomes stuck, no-one is able to use the index and performance suffers greatly. However, no functions fail. If the exclusive lock is set for more than 5 seconds and a task tries to use the index, a message is issued.

If the shared lock becomes stuck, the index cannot be rebuilt and performance suffers, but not as severely as a stuck exclusive lock. Again, no functions fail. A message is issued if a reindex is needed but cannot be done because the shared lock is not zero. If this message occurs with any frequency, investigate and consider manually clearing the lock.

## Update lock

The update lock is used (shared) when updates to global variables are made which change the structure of the data space. The exclusive lock is used by the GVT subtask when it writes the data space out to the DIV.

If the exclusive lock becomes stuck, no updates to global variables which require structural changes can be made (adding a new variable, changing a value to a longer value). If this occurs, requests fail and messages are issued.

If the shared lock becomes stuck, the GVT subtask is not able to write the data space out to the DIV. However, if GVT is unable to obtain the update exclusive lock, it forces it (that is, set the lock words to 1,0) after some period of trying to obtain the lock. A stuck shared update lock is automatically unstuck if GVT is active and if a DIV write needs to be done. Note that there is no impact of a stuck shared update lock except in these conditions.

# CHAPTER 6
# Using GDDR ISPF Interface

This chapter includes the following topics:


- Overview ..... 280
- Update GDDR ISPF profile (P) ..... 284
- Set up and maintain GDDR environment (M) ..... 286
- View GDDR configuration (G) ..... 365
- Perform GDDR health check (C) ..... 366
- Run GDDR scripts (S) ..... 375
- View GDDR script statistics (T) ..... 381
- Perform GDDR actions (A) ..... 382
- Run Dell EMC Started Task Execution Manager (ST) ..... 395

# Overview

The GDDR ISPF interface enables z/OS system programmers to perform configuration and administration tasks and operators to perform operator functions. To use the GDDR ISPF interface, invoke the GDDR invocation REXX exec which was customized in "Customize GDDR ISPF interface invocation REXX exec" on page 151.

The GDDR ISPF interface is available only on C-systems.

**Note:** You must install the GDDRPARM file and run GDDRMAIN before attempting to use the GDDR ISPF interface.

## Checking local maintenance

On entering the GDDR Primary Options Menu for the first time after logging in, the following popup is displayed as the GDDR maintenance level is determined for your TSO user address space. This may take several seconds.

```
+-----------------------------------------+
|  Refreshing local Maintenance info      |
|                                         |
|           Please wait...                |
+-----------------------------------------+
```

**Figure 31** Refreshing local Maintenance info popup

## Selecting GDDR subsystem

If multiple GDDR instances are running, a pop-up is displayed to select a particular instance for the GDDR ISPF session to be started, as shown in Figure 32:

```
------------ GDDR - Select GDDR Subsystem -------------------- Row 1 to 2 of 2
Command ===>                                                   Scroll ===> CSR

 This system has more than one active GDDR instance.
 Please select a GDDR instance from the list below.

                                Userid: JABCDE1
                GDDR maintenance level: V5.2.0 BASE
                Previous GDDR instance: GDD1

Enter S next to subsystem to select
Press <F3> to exit

Sel GDDR GDDRPLEX GDDRMAIN GDDRMAIN SCF  Maintenance    Configuration
    SSys Name     Jobname  JobID    Sfx  Level          Type
 _  GDD1 GDD1-52 GDD1MAIN STC09654 GA1  V5.2.0 BASE    SRDF/A (MSC)
 _  GDD2 GDDRMA   GDD2MAIN STC09815 GA2  V5.2.0 BASE    SQAR
****************************** Bottom of data ********************************
```

**Figure 32** Select GDDR Subsystem panel

- Select the required GDDR subsystem from the list and press **Enter** to go to the GDDR Primary Options Menu.

  The selected GDDR subsystem name will be shown at the beginning of the panel title in most GDDR ISPF panels, for example:

  **GDD1** - Primary Options Menu

## Checking GDDR maintenance level

If the GDDR maintenance level of the TSO user's address space (version and PTF level) does not exactly match that of the GDDRMAIN address space for the selected C-instance, the following popup is displayed:

```
+-----------------------------------------------------------+
| Command ===>                                              |
|                                                           |
| The GDDR maintenance level for ABCDEF1  does not match    |
| that of GDDRMAIN for the selected GDDR instance GDD1      |
|                                                           |
|         Jobname           : ABCDEF1  | GDDRMAIN           |
|         GDDR Version      : V5.2.0   | V5.2.0             |
|         Maintenance Level : GD52001  | BASE               |
|         Maintenance Date  : 02/02/18 | 02/01/18           |
|         Maintenance CRC   : EE28574A | 7B9794BC           |
|                                                           |
| Enter                                                     |
|  REFRESH to re-determine maintenance level for ABCDEF1    |
|  CONTINUE to proceed to GDDR instance GDD1 ***            |
|  END or press PF3 to exit                                 |
|                                                           |
|   *** WARNING: Results are unpredictable.                 |
|                                                           |
|                                                           |
|                                                           |
+-----------------------------------------------------------+
```

**Figure 33** Maintenance level mismatch detected

From this panel, the REFRESH command can be used to re-determine the maintenance level for the TSO user's address space. If the GDDR version still does not match, you can enter CONTINUE to proceed although results would be unpredictable, as indicated by the displayed message: `*** WARNING: Results are unpredictable.`

## Primary options

The Primary Options Menu panel is displayed, which looks similar to the following:

```
------------------ GDDR - Primary Options Menu for GDDR_AB --------------------
Option ===>

P  Profile       Update personal GDDR ISPF Profile        This System: GDDRDEVL
M  Maintenance   GDDR Setup and Maintenance                This Site: DC2
G  GDDR Config   View GDDR configuration                  This region: 1
C  Checkup       Perform pre-script checkup                  Master-C: GDDRDEVL
S  Scripts       Run GDDR Scripts                         Primary Site: DC1
T  Timing        View GDDR Script Statistics              Primary DASD: DC1
A  Actions       Perform GDDR Actions                       Automation: ON
ST eSTEM         Run Dell EMC STC Execution Manager    Planned script: None
                                                     Unplanned script: None

                                                             Appl ID.: GDDR



Dell EMC Geographically Dispersed Disaster Restart V5.2.0 BASE (06/15/18)
© 2007-2018 Dell Inc. or its subsidiaries. All rights reserved,

Select an option and press <Enter>
Press <F3> to Terminate GDDR ISPF
```

**Figure 34** Primary Options Menu panel

At the left side of the Primary Options Menu panel, a series of options are listed. To perform one of the following actions, enter the appropriate option on the command line and press **Enter**. See corresponding task description for each option:

- Update GDDR ISPF profile (P)
- Set up and maintain GDDR environment (M)
- View GDDR configuration (G)
- Perform GDDR health check (C)
- Run GDDR scripts (S)
- View GDDR script statistics (T)
- Perform GDDR actions (A)
- Run Dell EMC Started Task Execution Manager (ST)

## ISPF menu path convention

ISPF panel names and tasks described herein can be followed by capital letters in brackets to show the ISPF path to the required panel.

For example, (M,P) means:

- Choose option **M** in the Primary Options Menu.
- In the Setup and Maintenance Menu panel that opens, choose option **P**.

## ISPF panel titles

GDDR ISPF panel titles can include the following information:

- The name of the GDDR subsystem selected for the current ISPF session on the Select GDDR Subsystem panel
- The panel name
- The name of the GDDR-plex to which the settings apply

For example, in the “GDD3 - Primary Options Menu for GDDR_AB” title, GDD3 is the GDDR subsystem and GDDR_AB is the GDDR-plex.

## GDDR dashboard

At the right side, the Primary Options Menu panel and many other menu-type panels in the GDDR ISPF interface display the current settings for several GDDR control values:

```
        This System: GDDRDEVL
          This Site: DC2
        This region: 1
           Master-C: GDDRDEVL
      Primary Site: DC1
      Primary DASD: DC1
        Automation: ON
    Planned script: None
  Unplanned script: None

           Appl ID.: GDDR
```

**Figure 35** GDDR dashboard

The fields are as follows:

- This System

  Specifies the current system.

- This Site

  Specifies the current site.

- This region

  Specifies the current region.

- Master-C

  Shows the name of the C-system that currently owns the GDDR master function.

  See "Master C-system" on page 47 for information about the master C-system and its location.

- Primary Site

  Indicates the site where the business applications are currently running.

- Primary DASD

  Indicates the site at which the R1 DASD currently reside. Generally, the sites indicated by Primary Site and Primary DASD should be the same.

- Automation

  Indicates the current availability of GDDR automation functionality. The automation state can be changed as described in "Changing GDDR automation state" on page 286.

- Planned script

  Indicates which planned script, if any, is currently in progress.

  A planned script is considered to be in progress if it has been started but has not yet completed successfully; in this case, it is eligible for restart. If no planned script is currently in progress, 'None' displays in this field.

- Unplanned script

  Indicates which unplanned script, if any, is currently in progress.

  An unplanned script is considered to be in progress if it has been started but has not yet completed successfully; in this case, it is eligible for restart. If no unplanned script is currently in progress, 'None' displays in this field.

- Appl ID

  Indicates the ISPF application ID. GDDR uses the selected GDDR subsystem name as the ISPF application ID. This field is displayed in the Primary Options Menu dashboard only.

# Update GDDR ISPF profile (P)

Specify option **P** in the Primary Options Menu panel (Figure 34 on page 281) to access the GDDR ISPF profiles. The Change GDDR ISPF Profile Variable Values panel is displayed:

```
------------- GDD5 - Change GDDR ISPF Profile Variable Values -----------------
 Command ===>

   GDDR Subsystem Name      ===> GDD5
   JCL dataset              ===>
   ISPF skeleton dataset    ===>
   Global variable backup   ===>



   Jobcards for your user:
   //GDD5PACT JOB (0),'P',MSGLEVEL=(1,1),CLASS=A,MSGCLASS=X,SYSAFF=*,
   // NOTIFY=&SYSUID
   //*
   ____________________________________________________________________________

   Press ENTER to apply updates
   Press <F3> when finished
   Enter CANCEL to return without changing any profile variable values
   Enter CLEAR to set all values to null and exit
   Enter RESET to restore the values as they were upon entry
```

**Figure 36** Change GDDR ISPF Profile Variable Values panel (P)

The panel shows the job cards field only if your user ID is authorized to activate parameter changes or to manage GDDR scripts.

## Updating personal GDDR ISPF profile

Use the Change GDDR ISPF Profile Variable Values panel to specify GDDR-related variables pertaining to your user ID.

**Note:** Whenever a dataset name is required, specify a fully-qualified dataset name. TSO prefixing does not apply to any dataset name specified within GDDR.

1. Review the GDDR subsystem name.

   The GDDR Subsystem Name field shows the GDDR subsystem name for your active GDDR-plex. If you have multiple GDDR instances sharing the same C-systems, this field shows the GDDR subsystem name for your selected GDDR-plex.

   **Note:** "GDDR C-system multi-tenancy" on page 51 discusses running multiple GDDR instances.

   The GDDR Subsystem Name field defaults to GDDR and is not enabled for editing. Subsystem names are specified during the procedure described in "Install GDDR started procedures" on page 141.

2. In the JCL dataset field, type the name of the PROCLIB dataset holding GDDRPROC and GDDRWORK members customized during the GDDR integration procedure "Customize PROCLIB member GDDRPROC" on page 150.

3. In the ISPF skeleton dataset field, specify the name of the GDDR ISPSLIB library that resulted from your SMP/E installation. This dataset is used to retrieve ISPF file tailoring skeletons for the parameter activation, when done in background mode and when scripts are submitted.
4. In the Global variable backup field, type the name of your personal parameter backup dataset to be used for parameter management functions.

   **Note:** You defined parameter backup datasets during the GDDR integration procedure "Define GDDR datasets" on page 139.

5. Enter job card information.

   This field is displayed conditionally for GDDR users who are authorized to activate parameter changes and to manage GDDR scripts. The job card information entered here applies only to the current user and is used for parameter activation and script submission tasks. Always include the REGION=0M job card parameter. For example:

   ```
   ===>//JOBNAME JOB (acct-number),gddr-job,CLASS=A,REGION=0M,
   ===>//           MSGCLASS=A,SYSAFF=*,USER=GDDR,NOTIFY=GDDR
   ===>/*JOBPARM LINES=999999
   ===>//*
   ```

6. Press **F3** to save the values and return to the Primary Options Menu panel.

# Set up and maintain GDDR environment (M)

Specify option **M** in the Primary Options Menu panel (Figure 34 on page 281) to access the GDDR setup and maintenance functions. The Setup and Maintenance Menu panel is displayed:

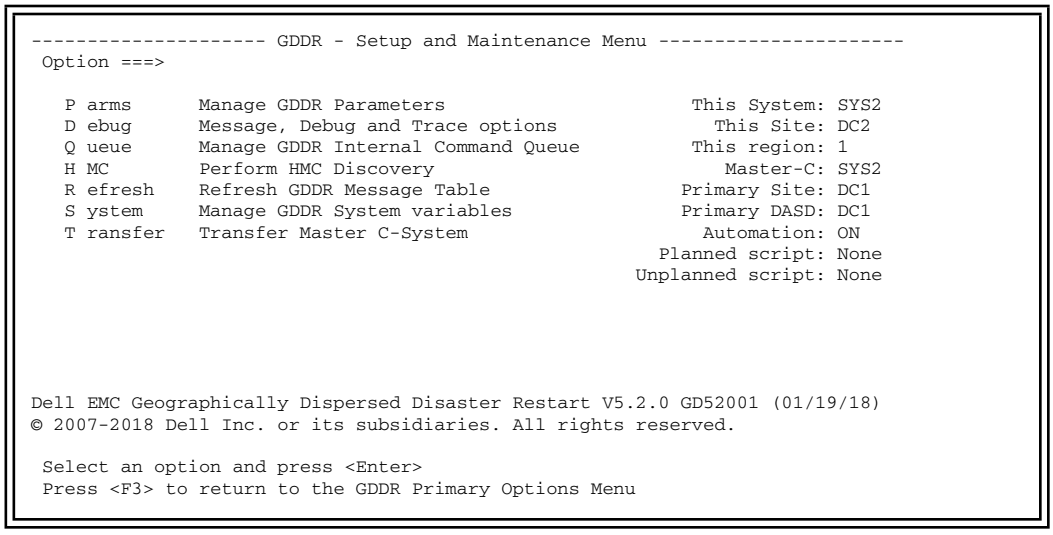



**Figure 37** Setup and Maintenance Menu panel (M)

The Setup and Maintenance Menu panel allows you to change the GDDR automation state, as described in "Changing GDDR automation state" on page 286.

The Setup and Maintenance Menu panel provides options required to complete the following tasks:

- Manage GDDR parameters (M,P)—GDDR Parameter Wizard
- Set message, debug and trace options (M,D)
- Manage GDDR internal command queue (M,Q)
- Perform HMC discovery (M,H)
- Refresh GDDR message table (M,R)
- Manage GDDR system variables (M,S)
- Transfer master C-system (M,T)

## Changing GDDR automation state

The Automation field displayed on the GDDR dashboard indicates the current availability of GDDR automation functionality.

To change the automation state:

- Type **GDDR ON** or **GDDR OFF** in the Option line of the Setup and Maintenance Menu panel.

The displayed value of the automation setting on the right of the panel reflects the change immediately.

**Note:** The GDDR ON/OFF command is valid only on the master C-system.

- When you specify **GDDR OFF** to turn GDDR automation off, the GDDR Event Monitor does not respond to events that would normally indicate a storage or system failure. Therefore, the OFF automation status should be used only when system availability may be impacted by scheduled maintenance activities.

  After you enter **GDDR OFF**, messages similar to the following appear:

  ```
  GDDR373I GDDR Broadcasting AUTOMATION.FLAG = OFF
  GDDR739I GDDR -> Set AUTOMATION.FLAG = OFF at DC1
  ```

- When you specify **GDDR ON** to turn GDDR automation on, GDDR operators will be able to run GDDR scripts.

## Manage GDDR parameters (M,P)—GDDR Parameter Wizard

Specify option **P** in the Setup and Maintenance Menu panel (M) to access the GDDR parameter management functions. The Parameter Management Options Menu panel is displayed:

```
------------------- GDDR - Parameter Management Options Menu ------------------
Option ===>

  B ackup       Manage GDDR Parameter backups             This System: LB01M34
  G DDRACDD     Manage GDDRACDD                             This Site: DC2
                                                          This region: 1
  P ARMLIB      Manage PARMLIB DSN backups                   Master-C: LB01M34
  .---------- Parameter Load functions --------.         Primary Site: DC1
 |  I nput       Select Parameter Input Dataset |        Primary DASD: DC1
 |  C onfig      Define Configuration Basics    |          Automation: ON
 |  D ata        Define Data Storage Objects    |      Planned script: None
 |  H ost        Define Data Host Objects       |    Unplanned script: None
 |  O ptions     Specify GDDR Options           |
 |  V alidate    Validate GDDR Parameter Set    |
 |  A ctivate    Activate GDDR Parameter Set    |
 |_____________________________________________|

Current work data set PCALLE1.GDDR.WORK

Select an option and press <Enter>
Press <F3> to return to the GDDR Setup and Maintenance Menu
```

**Figure 38** Parameter Management Options Menu panel (M,P)

The Parameter Management Options Menu options allow you to manage GDDR parameter backups and define and maintain GDDR parameters using the GDDR Parameter Wizard described in "GDDR Parameter Wizard" on page 54. The GDDR Parameter Wizard is accessible from this panel, using the Parameter Load functions.

For the initial setup of GDDR, it is strongly recommended that you go through the entire series of panels at least once to become familiar with all the required and optional features of GDDR, and to ensure that all defined elements are in agreement with the desired behavior of GDDR.

The Parameter Management Options Menu panel shows Parameter Load functions described in "Parameter Load functions" on page 288 or Parameter Review functions described in "Parameter Review functions" on page 289.

The options initially displayed in this panel may vary. The panel does not show options C(onfig), D(ata), H(ost), O(ptions), V(alidate), and A(ctivate) until the actions described in "Preparing to edit the current GDDR parameter set" on page 293 are completed.

The Parameter Management Options Menu panel provides options required to complete the following tasks:

- Manage GDDR parameter backups (M,P,B)
- Manage automated configuration discovery for DASD (M,P,G)
- Manage PARMLIB DSN backups (M,P,P)
- GDDR Parameter Wizard (Parameter Load) functions:
  - Select parameter input dataset (M,P,I)
  - Define configuration basics (M,P,C)
  - Define data storage objects (M,P,D)
  - Define host objects (M,P,H)
  - Specify GDDR options (M,P,O)
  - Validate GDDR parameter set (M,P,V)
  - Activate GDDR parameter set (M,P,A)

## Parameter Load functions

The series of Parameter Load function panels enable you to display or define parameter values organized by object category. The panels use the function keys as follows, so that fields are presented in data dependency order:

- F5 returns you to the preceding panel in the series.
- F6 takes you to the next panel in the series.
- F3 returns you to the menu panel for objects of the same type.

Each Parameter Load display and definition panel:

- Displays values for named parameters from the selected parameter dataset member, subject to the precedence rules that apply to data returned to the GDDR Parameter Wizard from the Auto-Discovery feature.
- Displays values which were saved on a previous Parameter Load panel.

If no value is supplied for a field by the selected parameter dataset member, the first time a value for that field is specified as panel input and saved to the GDDR Parameter Wizard work dataset, the value is propagated to all succeeding panels which contain the same field.

Changes to field values in parameter display and definition panels are saved to your GDDR Parameter Wizard work dataset when you type **SAVE** on the command line of the panel. After editing the values in the panel and before you use the SAVE command,

if you wish to restore the original values displayed in the panel, type **LOAD** in the panel. If you have 'Saved' interim changes for values on the current panel, then the Load command restores the last saved values.

**Note:** Each panel in which you use the SAVE command will be utilized for a partial parameter activation, whether or not you made changes on the panel.

## Parameter Review functions

GDDR makes the current parameters available for review through the Parameter Management Options Menu panel (M,P), without the validation and activation capabilities. For example, an audit of your disaster restart operations may include a review of GDDR parameters. The GDDR parameters review capability enables a third-party reviewer to access a current parameter backup with no exposure from unwanted changes to the active parameter values.

Authority to use the GDDR parameters review capability is granted to members of the GDDR$REV RACF group with READ access to the GDDRISPF.ACCESS, GDDRISPF.SETUP.ACCESS, and the GDDRISPF.SETUP.PARMS.REVIEW facility profiles, as described in Table 14 on page 133. When authority has been granted, the Parameter Management Options Menu panel (M,P) is displayed without the Validate and Activate Parameter Load functions.

The parameter review is performed by making a backup of the current parameters, as described in "Manage GDDR parameter backups (M,P,B)" on page 290. The reviewer selects the backup member to be used to prepare the reviewer's parameter work dataset in the Select Parameter Input Dataset panel (M,P,I).

Within the Select Parameter Input Dataset panel (M,P,I), after you select a backup member which will be used to populate the GDDR Parameter Wizard work dataset, the following message is displayed:

```
Ready to start preparation of the work dataset. Press <F3> to proceed.
```

To display the reviewer's version of the Parameter Management Options Menu panel (M,P), type **review** in the command line of the Select Parameter Input Dataset panel (M,P,I), rather than pressing <F3> to proceed, as the message directs.

The Parameter Review functions listed in the reviewer's version of the Parameter Management Options Menu panel display the parameter values of the current backup member.

## Manage GDDR parameter backups (M,P,B)

When you specify option **B** in the Parameter Management Options Menu panel (M,P), the Manage GDDR Parameter Backups panel is displayed:

```
------------------- GDDR - Manage GDDR Parameter Backups ---- Row 1 to 9 of 11
 Command ===>

 1 Backup       Create new GDDR Parameter Backup

     Backup Dataset ===> JABCDE1.GDDR520.BACKUP.PARMS
 Backup description ===> ____________________________________________
 Backup consistency ===> Y   (Y/N/1-9)

 Press <F3> to return to the GDDR Parameter Management Options Menu
 Line commands: B rowse, E dit, D elete, M odify description
                S elect for Parameter Load
 ------------------------------ Previous Backups -----------------------------
 Act  Member    Date    Time   Userid                  Description
 --- -------- -------- ----- -------- ------------------------------------------
  _  I5PE3025 18/05/25 14:30 PQRSTU1  A:testing
  _  I5PE3024 18/05/25 14:30 PQRSTU1  B:testing
  _  I5LG1928 18/05/21 16:19 OPQRST1  B:ptw test
  _  I5LG1852 18/05/21 16:18 OPQRST1  B:ptw test
  _  I5LG1613 18/05/21 16:16 OPQRST1  B:ptw test
  _  I5LG1152 18/05/21 16:11 OPQRST1  B:ptw test
  _  I5LG0448 18/05/21 16:04 OPQRST1  B:ptw test
  _  I5JF4346 18/05/19 15:43 RGHIJK1
 ****************************** Bottom of data ********************************
```

**Figure 39** Manage GDDR Parameter Backups panel (M,P,B)

The Manage GDDR Parameter Backups panel allows you to create new GDDR parameter backup as described in "Creating GDDR parameter backups" on page 291.

The Manage GDDR Parameter Backups panel provides the following fields:

- Backup Dataset

  The backup dataset name can be one of the following:

  - The GDDR variable backup dataset name specified in the Change GDDR ISPF Profile Variable Values panel (P).
  - The dataset name associated with the BKUPVARS dataset type in the Define GDDR Datasets panel (M,P,C,D).
  - Any PDS or PDSE dataset name you type in this field.

- Backup description

  Use this field to describe the circumstances regarding the backup.

- Backup consistency

  Use this field to enforce the consistency of the parameter set being backed up. GDDR global variable backup consistency is enforced by activating a dynamic exit which protects global variables from updates in the time between the start and the completion of the backup. See "Dynamic exits" on page 143 for more information about this exit.

  Valid values are Y, N, and numerals 1-9. Entries 1-9 indicate the number of retries to achieve exclusive control of global variables before backup processing begins, or the attempts to begin backup processing are halted.

It is recommended to perform all parameter backups with consistency. If you take a backup without consistency, you should manually quiesce GDDR activity to prevent global variable changes during the time it takes to perform the backup. If backups fail regularly due to concurrent activity, specify a number from 1 to 9 to increase the number of retries.

- Previous Backups

  Parameter backup dataset members are used as input by the parameter definition process. New members are created by the GDDR Heartbeat Monitor at initialization, by the user using the GDDR ISPF interface, and during parameter activation to capture a “Before” and “After” image of the existing parameters. In the example shown in Figure 39 on page 290, there is a list of dataset members that had been created by previous parameter activations, where "B" indicates Before and "A" indicates After.

- Member

  For either automatic or requested backups, the member name is generated automatically by the GDDR parameter backup processor. It encodes the time of the backup (including year, month, day, hour, minute, and second).

- Line commands

  Line commands enable the following actions in the Action (Act) column in the Previous Backups member list:

  - B—Browse

    Enter the ISPF browse mode for the dataset and member.

  - D—Delete

    Delete the dataset member.

  - E—Edit

    Enter the ISPF edit mode for the dataset and member.

  - M—Modify the member description

    Edit the parameter backup member description.

  - S—Select for Parameter Load

    Select a member. The selected member is displayed in the Selected Member field of the Select Parameter Input Dataset panel (M,P,I).

## Creating GDDR parameter backups

GDDR automatically creates GDDR parameter backups before and after parameters are activated and during GDDR Heartbeat Monitor initialization. Global variable backup is accomplished by running a script (GDDRPGVB) in a local REXX worker (GDDWXR). The backup is made into the dataset named in the global variable GLOBAL.GDDR.PARMS.GDDRVAR_BACKUP.<*system*>.

However, there may be other times when it is necessary to back up the GDDR parameters. For example, you can create a backup to be used as input to your next parameter definition. The Manage GDDR Parameter backups option (M,P,B) allows you to do this.

Backups of all parameters are automatically taken during GDDR Heartbeat Monitor initialization and as part of GDDR parameter activation. A parameter backup is also taken when you specifically request a backup using option **B** on the Parameter Management Options Menu panel (M,P), or when you use the GDDRMAIN GVB console command (or the GDDRMAIN EXEC parameter GVB) with the *dw*(*hhmm*), *(*hhmm*), or NOW operand as described in "GVB" on page 206.

For the automatic and GDDRMAIN GVB initiated backups, the dataset to which the backup is written is obtained from the Dataset Name field of the Define GDDR Datasets panel (M,P,C,D), dataset type of BKUPVARS. For requested backups, the dataset name of dataset type BKUPVARS is presented as a default, but another backup dataset may be specified.

**Creating a new GDDR parameter backup**

To create a new GDDR parameter backup:

1. Specify option **B** on the Parameter Management Options Menu panel (M,P).

   If the Global variable backup dataset name specified in the Change GDDR ISPF Profile Variable Values panel (P) differs from the dataset associated with BKUPVARS in the Define GDDR Datasets panel (M,P,C,D), then the Select Dataset for GDDR Parameter Backup panel shown in Figure 40 is first displayed. This panel presents you with both datasets so you may choose the one to use for parameter backup:

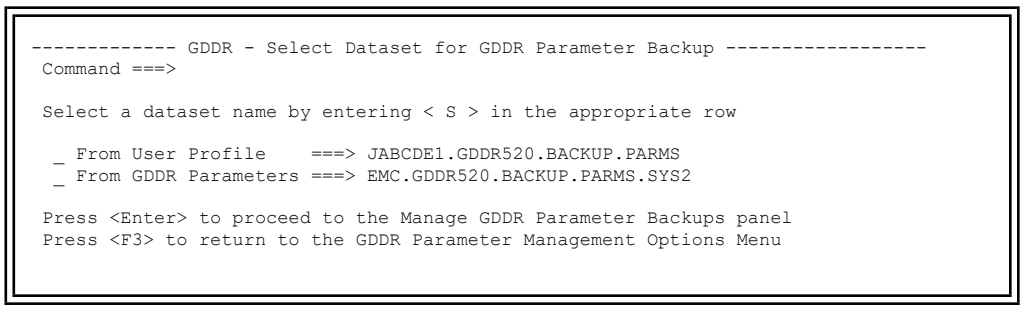



**Figure 40** Select Dataset for GDDR Parameter Backup panel

2. Select a dataset name and press **Enter** to go to the Manage GDDR Parameter Backups panel (M,P,B).
3. Specify a dataset name in the Backup Dataset field.
4. Type **B** in the command line in the Manage GDDR Parameter Backups panel (M,P,B) and press **Enter**.

   When the parameter backup processing completes, the following messages are displayed:

```
GDDR721I GDDR Starting GDDR Global Variable Backup
GDDP304I Backup successful, 503 lines written to PARMOUT
GDDR639I GDDR Completed GDDR Global Variable Backup with rc 0
```

### Preparing to edit the current GDDR parameter set

To prepare to edit the current GDDR parameter set:

The Manage GDDR Parameter Backups panel (M,P,B) also lets you prepare to edit the current GDDR parameter set. You do this when you select an existing member of the parameter backup dataset which is displayed.

1. Type **S** next to the desired member in the Previous Backups list displayed in the Manage GDDR Parameter Backups panel (M,P,B) and press **Enter**.
2. The Select Parameter Input Dataset panel displays. Specify the panel options as described in "Select parameter input dataset (M,P,I)".

## Manage automated configuration discovery for DASD (M,P,G)

When you specify option **G** in the Parameter Management Options Menu panel (M,P), the GDDRACDD - Automated Configuration Discovery for DASD panel is displayed:

```
------------ GDDRACDD - Automated Configuration Discovery for DASD ------------
  Command ===>

  Press <F3> to exit
  Enter the required information. Press <Enter> to run GDDRACDD
                                                                        More:     +
    Run where? ===> FORE  (FOREground or BACKground)
    Hold Class ===> X     SDSF Hold Output class.
    Range size ===> ___   (default: 256)
         ALIGN ===> _     (Y or N)
           DLM ===> _     (Y or N)
           BCV ===> _     (Y or N)

           RAID ===> _    (Y or N)
           META ===> _    (Y or N)
           SIZE ===> _    (Y or N)

        GLOBALS ===> _____ (CHECK or NONE)
           COMP ===> _     (Y or NO)
           SCAN ===> _     (Y or NO)
          FORCE ===> _     (Y or NO)

    CONFIG site ===> ___   (DC1/DC2/DC3/DC4)
       Topology ===> ____  (CONCurrent or CASCaded)

           Sort ===> _     (Y or N)

    Group break ===> _     (Y or N)

         Debug ===> 0      (0, 1, or 2)
         Trace ===> 0      (0, 1, 2, or 3)  Routine ===> ____________________
     SCF subsys ===> SCF_  (SCFx)
    GDDR subsys ===> GDDR  (GDDx)

   ACDDPARM dsn ===> ____________________________________________________
   GDDR JCLLIB  ===> ____________________________________________________

   Output files:
   E04SRDFD dsn ===> ____________________________________________________
   E05TFDEV dsn ===> ____________________________________________________

   ******************************** Bottom of Data ********************************
```

**Figure 41** GDDRACDD - Automated Configuration Discovery for DASD panel (M,P,G)

This panel lets you set up and run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412.

### Performing automated DASD discovery for active GDDR Parameter Wizard session

1. Ensure that the GDDR and prerequisite host software installation is complete on the C-system where you also have GDDRSCF running.
2. Point the GDDRACDD E04SRDFD and E05TFDEV DD cards to your GDDR Parameter Wizard work dataset.
3. Run GDDRACDD as a batch job or use the GDDRACDD panel interface (M,P,G).
4. When GDDRACDD completes, issue a LOAD command on the Define SRDF Device Ranges panel (M,P,D,S). This refreshes the panel with the generated RDF.Devices data.
5. Review and, where necessary, modify the RDF.DEVICES parameters created by GDDRACDD.
6. If you made any changes to the RDF.DEVICES parameters, issue a SAVE command in the Define SRDF Device Ranges panel (M,P,D,S).
7. Repeat steps 4-6 above for TimeFinder devices, using the Define TimeFinder Device Ranges panel (M,P,D,T). Note that SRDF and TimeFinder devices can be discovered with a single run of GDDRACDD.

## Manage PARMLIB DSN backups (M,P,P)

When you specify option **P** in the Parameter Management Options Menu panel (M,P), the Dataset name to Backup subpanel is displayed:

```
+----- GDDRIPDS - Dataset name to Backup. ------+
|                                                |
| SYSU.GDDR520.CUSTOM.LIB.K138                   |
|   Without Quotes                               |
| Enter to process       PF3 to Exit             |
+------------------------------------------------+
```

**Figure 42** Dataset name to Backup panel

This panel prompts you to enter the name of the dataset to back up. Type the desired PARMLIB name in the panel and press **Enter** to back up the PARMLIB or **PF3** to exit without a backup.

## Select parameter input dataset (M,P,I)

When you specify option **I** in the Parameter Management Options Menu panel (M,P), the Select Parameter Input Dataset panel is displayed:

```
------------------ GDD5 - Select Parameter Input Dataset ---- Row 1 to 1 of 1
 Command ===>

     Parameter input dataset ===> EMC.GDDR520.PARMS.BKUP.M5.M34
              Selected Member ===>                Unselect? ===> N  (Y/N)
 Parameter Load work-dataset ===> ____________________________________________
  Parameter Load description ===> ____________________________________________

   Currently activated GDDR Parameter source:
   EMC.GDDR20.PARMS.LAPD.M5.M3A  on LB01M3A
   (User: PQRSTU1 07/17/18 10:33:31)

 Select '--NONE--' or choose a different GDDR Parameter Load input dataset.
 Press <F3> to return to the Parameter Management Options Menu
 Line commands: S elect, U nselect, C leanup, B rowse, E dit
---------------------- Parameter Input Member Selection ----------------------
Act  Member    Date   Time   Userid                 Description
--- -------- -------- ----- -------- ----------------------------------------
 _   --NONE-- 18/03/08 06:44 PQRSTU1   GDDR parameter load from scratch
 _   J2R90530 18/02/27 09:05 PQRSTU1   A: Change control @100227
 _   J2R90501 18/02/27 09:05 PQRSTU1   B: Change control @100227
****************************** Bottom of data ********************************
```

**Figure 43** Select Parameter Input Dataset panel (M,P,I)

This panel is used to manage GDDR parameter datasets as described in the following sections:

- "Registered and unregistered backups" on page 296
- "Editing current GDDR parameter set" on page 297
- "Reviewing current GDDR parameter set" on page 298
- "Edit-in-progress serialization" on page 298
- "Releasing Edit-In-Progress serialization lock (same user)" on page 299

The following fields are provided on the Select Parameter Input Dataset panel (M,P,I):

- Parameter input dataset

  Specify the parameter backup dataset name. It can be one of the following:

  - The global variable backup dataset name specified for you in the Change GDDR ISPF Profile Variable Values panel (P).
  - The dataset name associated with the BKUPVARS dataset type in the Define GDDR Datasets panel (M,P,C,D).
  - Any PDS or PDSE dataset name you type in this field.

- Selected Member (display only)

  Indicates the currently selected backup member.

- Parameter Load work-dataset

  Enter the GDDR Parameter Wizard work dataset that was defined in "Define parameter management datasets" on page 139. This dataset contains your in-process work, enabling you to assemble a complete parameter set by saving your input data from each of the parameter definition panels, and returning to the task at a later time.

  ---

  **Note:** The parameter input dataset and the GDDR Parameter Wizard work dataset must be two different datasets. This is because the contents of the GDDR Parameter Wizard work dataset are overwritten when exiting the Select Parameter Input Dataset panel (M,P,I).
  The GDDR Parameter Wizard work dataset must also be different from the dataset that is defined to GDDR as the *last activated parameter dataset* defined in "Define parameter management datasets" on page 139.

  ---

- Parameter Load description

  Use this field to describe the circumstances for the parameter load or update.

- Currently activated GDDR Parameter source

  Indicates which work dataset or parameter backup dataset was used for the last activation, the system where the last activation was done, the timestamp for that activation, the user ID, and the circumstances of the update.

- Parameter Input Member Selection

  Displays the current parameter members.

- Line commands

  Line commands enable the following actions in the Action (Act) column in the Parameter Input Member Selection list:

  - S—Select

    Display the Select Parameter Input Dataset panel for the dataset and member.

  - U—Unselect

    Remove the member.

  - B—Browse

    Enter the ISPF browse mode for the member.

  - E—Edit

    Enter the ISPF edit mode for the member.

## Registered and unregistered backups

The parameter backup member information shown in the Select Parameter Input Dataset panel (M,P,I) is maintained in GDDR variables for all parameter backups created by GDDR—either as an impromptu user backup, as a backup created before and after a parameter activation, or as a backup created by the GDDR Heartbeat Monitor during initialization. These actions are referred to as *registered* backups.

Where the backup description starts with B: or A: this reflects parameter backups taken by a previous activation process, where B indicates “Before” and A indicates “After” the actual update of GDDR parameters.

*Unregistered* or outside parameter backup members may be copied into your backup dataset containing registered parameter backups to facilitate configuration of additional GDDR-plexes or reconfiguration of existing GDDR-managed environments. Unregistered parameter backup members are displayed when you enter the 'showall' command on the command line of the Select Parameter Input Dataset panel (M,P,I). The showall command updates the Parameter Input Member Selection list with members which you have previously copied to your specified parameter input dataset.

## Editing current GDDR parameter set

To edit the current GDDR parameter set:

1. In the Select Parameter Input Dataset panel (M,P,I), press **F3** to edit your GDDR parameter set by copying the contents of the selected parameter backup member to your GDDR Parameter Wizard work dataset.

   This operation overwrites the contents of your GDDR Parameter Wizard work dataset.

   A confirmation panel requests you to confirm the operation:

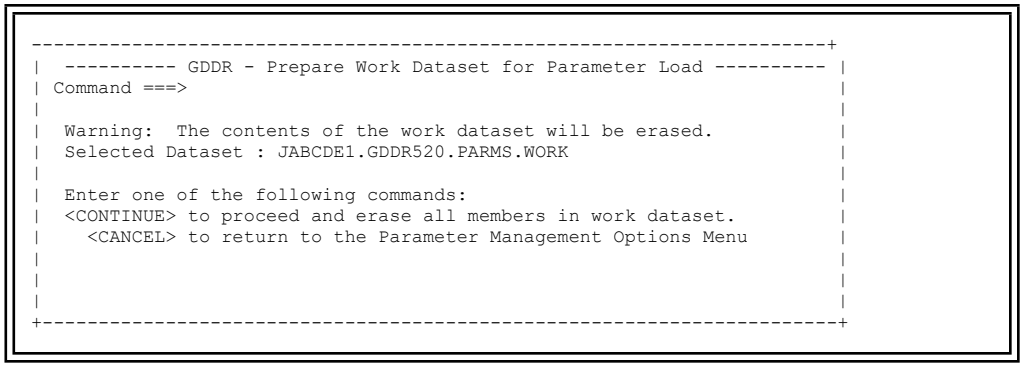



**Figure 44** Prepare Work Dataset for Parameter Load confirmation panel

2. Enter **CONTINUE** to proceed and overwrite the contents of the work dataset with the selected member of the parameter input dataset.

   A panel similar to the following displays the status of the operation:

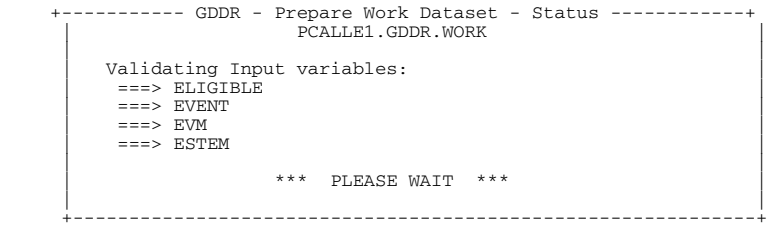



**Figure 45** Prepare Work Dataset status panels

On completion of the Prepare Work Dataset process, message GDDI172I confirms that your work dataset has been populated with the parameter dataset member you selected in the Select Parameter Input Dataset panel (M,P,I).

The Parameter Management Options Menu panel redisplays with the following message:

```
+-------------------------------------------------------------+
| GDDI172I Member I6L13905 selected as GDDR Parameter Load Input |
+-------------------------------------------------------------+
```

## Reviewing current GDDR parameter set

Select any of the Parameter Review functions to view or change values which were populated from the selected input parameter dataset member.

## Edit-in-progress serialization

GDDR Parameter Wizard work dataset serialization starts when you issue the PREPARE command on the Select Parameter Input Dataset panel (M,P,I), and ends after the successful activation of the parameter set, when the processing performed by the Activate GDDR Parameter Set panel (M,P,A) completes.

While editing of the parameter work dataset is in progress, the GDDI010I informational message is issued on entry to the Select Parameter Input Dataset panel (M,P,I), as shown in Figure 46.

```
+-------------------------------------------------------------------+-------+
| GDDI010I Edit in progress RGHIJK1.GDDR.V520.PARMS.WORK by RGHIJK1 on  |
| SYS2 at DC2 2018/01/27 16:42:32 EST                                   |
+-----------------------------------------------------------------------+
```

**Figure 46** GDDI010I message - User 1

The GDDI010I message indicates the name of the work dataset, the TSO ID of the user editing (if no user ID is shown in the message, the current TSO user is the one with the Edit-in-Progress serialization lock), the C-system name of the system on which the parameter edit is in progress, as well as the site (DC*n*) and the time and date that the edit started.

The serialization can be overridden by another user with proper authority, when the user issues another PREPARE command. Note that the other user must have 'READ' authority to facility profile GDDRISPF.SETUP.PARMS.FORCE, as described in "Set up GDDR security" on page 130.

When issuing a PREPARE command while an edit is in progress, the new user with 'FORCE' authority is presented with the FORCE option displayed on the popup panel shown in Figure 47.

```
+-------------------------------------------------------------------+
| -------------------- Parameter Edit Warning --------------------- |
| Command ===>                                                      |
|                                                                   |
|  Warning:  User RGHIJK1 is currently editing GDDR parameter       |
|        work data set RGHIJK1.GDDR.V520.PARMS.WORK                 |
|        on SYS2 at DC2 starting 2018/01/27 16:42:32 EST            |
|                                                                   |
|                                                                   |
|  Enter FORCE to abandon changes and start a new EDIT session      |
|  Enter CANCEL or press  <F3> to return to the previous panel.     |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

**Figure 47** Parameter Edit Warning panel - User 2

If the FORCE command is entered, the TSO user ID associated with the serialization lock is changed to the second TSO user's ID, and a new edit session effectively begins. The TSO user who originally held the serialization lock will then receive the GDDB010E message with the prompt to indicate they can no longer proceed with their edit session either with the message 'Reverting to REVIEW mode', as shown in Figure 48 on page 299 or with the message 'Reverting to initial state', as shown in Figure 49 on page 299. Following the display of message GDDB010E, the original TSO user's session displays the Parameter Management Options Menu panel (M,P) in review mode, showing the Parameter Review functions menu.

After a successful FORCE, the previous TSO user's session state will either:

- Revert to the initial state (where only the Backup and Input options are shown)
- Revert to the REVIEW mode state

If the second PREPARE command issued by the other user references the *same* parameter work dataset name that was being edited by the first user, then the first user reverts to the initial state. If the second PREPARE command issued by the other user references a *different* parameter work dataset name, then the first user reverts to REVIEW mode on the parameter work dataset. Message GDDI010E will appear on the GDDR Parameter Wizard panel currently displayed by first user's TSO session.

```
+-------------------------------------------------------------------------------+
| GDDI010E Edit in progress RGHIJK1.GDDR.V520.PARMS.WORK FORCEd by PQRSTU1 on |
| Q312 at DC2 2018/02/01 16:29:28. Reverting to REVIEW mode.                    |
+-------------------------------------------------------------------------------+
```

**Figure 48** Message GDDI010E in GDDR - Define Configuration Basics panel after FORCE of Edit-in-Progress serialization lock - User 1

```
+-------------------------------------------------------------------------------+
| GDDI010E Edit in progress RGHIJK1.GDDR.V520.PARMS.WORK FORCEd by PQRSTU1  |
| on SYS1 at DC1. 2018/02/01 13:12:50. Reverting to initial state.          |
+-------------------------------------------------------------------------------+
```

**Figure 49** Message GDDI010E in Parameter Management Options Menu panel after FORCE of Edit-in-Progress serialization lock - User 1

## Releasing Edit-In-Progress serialization lock (same user)

Complete the following steps to release the Edit-in-Progress serialization lock by the TSO user who started it, without activating the parameter set:

1. Enter the REVIEW line command on the Select Parameter Input Dataset panel (M,P,I).
2. The GDDR Parameter Wizard responds by prompting you for a FORCE command to abandon your changes.
3. Reply FORCE to abandon the EDIT session and start the REVIEW session, which ends the Edit-In-Progress serialization lock.

## Define configuration basics (M,P,C)

When you specify option **C** in the Parameter Management Options Menu panel (M,P), the Define Configuration Basics panel is displayed:

```
-------------------- GDDR - Define Configuration Basics -----------------------
Option ===> F

       F Define Configuration Features                  This System: SYS2
       C Define C-Systems                                 This Site: DC2
       D Define GDDR Datasets                           This region: 1
       R Define Site Roles and Groups                      Master-C: SYS2
                                                       Primary Site: DC1
                                                       Primary DASD: DC1
                                                         Automation: ON
                                                    Planned script: None
                                                  Unplanned script: None




Select an option and press <Enter>


<F5> Parameter Management Options Menu  <F6> Define Configuration Features
Press <F3> to return to the Parameter Management Options Menu
```

**Figure 50** Define Configuration Basics panel (M,P,C)

The Define Configuration Basics panel lists options to define your GDDR configuration, including the following tasks:

- Define configuration features (M,P,C,F)
- Define C-systems (M,P,C,C)
- Define GDDR datasets (M,P,C,D)
- Define site roles and groups (M,P,C,R)

IMPORTANT

Before completing the tasks, read the important information provided in "Define Configuration Basics options" on page 300.

### Define Configuration Basics options

This series of Define Configuration Basics panels is used once during initial installation of GDDR, and is unlikely to be used again. Most changes performed here will require a full parameter activation as well as edits to GDDRPARM statements. The following changes can be done with a partial parameter activation:

- Setting FBA Devices to Y/N after you select option **F**, Define Configuration Features
- Changing dataset names after you select option **D**: Define GDDR Datasets

Complete the tasks listed in the Define Configuration Basics panel (M,P,C) to define your GDDR configuration. As you complete each task, save your input to ensure that you can return to the task with no loss of data. If you attempt to exit a panel before a save has been completed, you are reminded with the following message:

```
---------------------- SAVE unsaved changes? --------------------
 Command ===>

  Warning:  You may have unsaved changes.

  Enter SAVE to save your work and continue to the next panel.
  Enter CANCEL to return to the panel you were on.
  Press <F3> key to continue and lose unsaved changes.
```

**Figure 51** Prompt to save unsaved changes

Note that the "Save unsaved changes?" confirmation pop-up may appear even when you have not manually entered data on the current panel. This is because the GDDR Parameter Wizard is a series of panels with a specific sequence. Panels earlier in the sequence may necessitate adding new information to panels later in the sequence. This occurs frequently in the subset of panels under "Define Host Objects". Another possibility is that the GDDR Parameter Wizard automatically added new data for the current panel, based on input from GDDRPARM or other sources of Auto-Discovery.

If the managed systems are changed (added/deleted/replaced), it is recommended to step through the entire sequence of Define Host Objects panels. Where GDDR expects additional information, the wizard will insert new lines on the panel. These are usually shown initialized with underscores, and sometimes with template information or with discovered information. This creates a difference between the panel content and the GDDR Parameter Wizard work dataset. When an attempt is then made to step to the next wizard panel, the "Save unsaved changes?" confirmation pop-up will be displayed as a signal that you need to confirm or complete the information added by the wizard.

## Define configuration features (M,P,C,F)

When you specify option **F** in the Define Configuration Basics panel (M,P,C), the Define GDDR Configuration features panel is displayed, similar to the one shown in Figure 52. To proceed, select a configuration and press **Enter**. The panel updates to display configuration-specific options.

```
------------------ GDDR - Define GDDR Configuration features --- Row 1 to 1 of 1
  Command ===>

  Site list: DC1 DC2 DC3 DC4                    FBA Devices: Y  (Y/N)
                                                DLm support: N  (Y/N)
                                       SRDF/SQAR HA level: 2  (0/2)
  SRDF Recovery Preference:             SYNC  (Sync/Async/None)

  GDDR zDP Management     :  DC1: _   DC2: _   DC3: _   DC4: _  (Y/N/blank)
  SNAP-VX SOFTLINK Support:  DC1: _   DC2: _   DC3: _   DC4: _  (Y/N/blank)
  SNAP-VX is TARGETLESS   :  DC1: _   DC2: _   DC3: _   DC4: _  (Y/N/blank)
  Legacy TimeFinder Method:  DC1: _   DC2: _   DC3: _   DC4: _  (M/C/blank)

  Select a configuration from the list below and Press <Enter> to Validate.
  <F3> Return to previous Menu        <F6> Define C-Systems
  Type SAVE to save, LOAD to restart from last saved values.

  Sel      Configuration
  ---      ------------------------------------------------------------
   S  ***  SRDF/SQAR with AutoSwap & R22 support                  <===Current
   ***End of configuration list***
```

**Figure 52** Define GDDR Configuration features panel (M,P,C,F)

The Define GDDR Configuration features panel displays the list of sites from the GDDRPARM file you installed as described in "Install GDDRPARM file" on page 145.

This panel displays the configurations and populates values that are valid selections for the Site list.

1. Specify the following values:

   - FBA Devices

     Specify **Y** or **N** to indicate whether Fixed Block Address (FBA) devices will also be managed by GDDR.

     CKD devices are always managed by GDDR by default.

   - DLm support

     Specify **Y** or **N** to indicate whether DLm virtual tape drives will be managed by GDDR.

   - SRDF/SQAR HA level

     Specify **0** or **2** to indicate the desired level of support for SRDF/SQAR High Availability:

     – 0

       Configuration does not support SRDF/SQAR HA operation.

     – 2

       (recommended) Full SRDF/SQAR HA support.

     See "SRDF/SQAR High Availability (HA)" on page 74 for more information about this feature.

- SRDF Recovery Preference

  Specify **SYNC** or **ASYNC** to set the desired preference for establishing SRDF recovery replication:

  – SYNC: The GDDRUP31 script will first try to recover SRDF/S before moving on to SRDF/A recovery given that the SRDF/S can be recovered based on the state of the environment.

  – ASYNC: The GDDRUP31 script will only recover the SRDF/A and leave the SRDF/S recovery to be performed by the GDDRPA0A script.

- GDDR zDP management

  Specify **Y or N** to indicate whether you want GDDR to manage zDP activity. See "TimeFinder and zDP field value combinations" on page 304 for additional information about how this field value is set.

  It is recommended to set this field to **Y** for every DASD site location protected by VDGs.

  "zDP integration" on page 78 provides details about GDDR zDP integration.

- SNAP-VX SOFTLINK Support

  Specify **Y** or **N** to indicate whether or not SnapVX softlinking support is required in GDDR for the listed site. See "TimeFinder and zDP field value combinations" on page 304 for additional information about how this field value is set.

  You must specify **Y** if there is any non-GDDR usage of SnapVX with softlinking (for example, zDP) using the GDDR-managed SRDF devices as source devices. You can also specify **Y** if you prefer GDDR to use SnapVX with softlinking.

  The default option is **N**. You must specify **N** if the listed site has no VMAX3 or later system.

  "SnapVX softlinking support" on page 77 provides details about this feature.

- SNAP-VX is TARGETLESS

  Specify **Y** or **N** to indicate whether you want GDDR to manage a SnapVX targetless configuration. See "TimeFinder and zDP field value combinations" on page 304 for additional information about how this field value is set.

  If the field is left blank, it defaults to **N**.

  "Targetless SnapVX support" on page 77 provides details about this feature.

- Legacy TimeFinder Method

  Specify **M** to have GDDR use TimeFinder/Mirror with Clone Emulation for the listed site (default), or specify **C** to have GDDR use TimeFinder/Clone for the listed site. See "TimeFinder and zDP field value combinations" on page 304 for additional information about how this field value is set.

- Configuration

  Specify **S** in the SEL column of one of the rows to select the name of the configuration which matches your site's storage replication environment.

2. Press **Enter** to validate the selected configuration.

3. Type **SAVE** to save your selection.
4. Press **F6** to go to the Define C-Systems panel (M,P,C,C).

#### TimeFinder and zDP field value combinations

The GDDR zDP Management, SNAP-VX SOFTLINK Support, SNAP-VX is TARGETLESS, and Legacy TimeFinder Method fields are site-specific, and affect each other as follows:

**Table 31** TimeFinder and zDP field value combinations

| Field value for a site | Result for the affected site |
|---|---|
| GDDR zDP Management = **Y** | SNAP-VX SOFTLINK Support is forced to **Y**.<br>Legacy TimeFinder Method field is forced to **C**. |
| SNAP-VX SOFTLINK Support = **Y** | Legacy TimeFinder Method is ignored.<br>GDDR uses SnapVX with softlinking. |
| SNAP-VX SOFTLINK Support = **N**<br>Legacy TimeFinder Method =**C** | SNAP-VX SOFTLINK Support is ignored.<br>SNAP-VX is TARGETLESS is forced to **N.**<br>GDDR uses:<br>• SnapVX with softlinking on PowerMax, VMAX All Flash and VMAX3 systems<br>• TF/Clone on VMAX V1 and V2 systems |
| SNAP-VX SOFTLINK Support = **N**<br>Legacy TimeFinder Method =**M** | GDDR uses TF/Mirror. |
| SNAP-VX is TARGETLESS = **Y** | GDDR uses targetless SnapVX.<br>SnapVX SOFTLINK Support must be set to **Y**. |

## Define C-systems (M,P,C,C)

When you specify option **C** in the Define Configuration Basics panel (M,P,C), the Define C-Systems panel is dislayed:

```
------------------------- GDDR - Define C-Systems --------------------------
 Command ===>

 Press <F3> to return to the Define Configuration Basics Menu
 <F5> Define Configuration Features              <F6> Define GDDR Data Sets
 Type SAVE to save, LOAD to restart from last saved values.

        System                IPL
  Site   Name    SMFID     Parameters     CPC (netid.nau)     LPAR
  ---- -------- -------- --------------- -----------------  --------
  DC1    SYS1   SYS1     7517 , 751799M1 IBM390PS.Q3        SYS1
  DC2    SYS2   SYS2     751E , 751E99M1 IBM390PS.O         SYS2
  DC3    SYS3   SYS3     7509 , 750999M1 IBM390PS.C         SYS3
  DC4    SYS4   SYS4     7509 , 750999M1 IBM390PS.F         SYS4

 *--------------------------- Auto-Discovery Legend ---------------------------*
 * Green  - Display only. Auto-Discovery had no conflict with input.           *
 * Yellow - Display only. Restricted by Auto-Discovery, conflict with input.   *
 *   Red  - User input allowed. Not restricted by Auto-Discovery,              *
 *          or impossible to discover at this time.                            *
 *-----------------------------------------------------------------------------*
```

**Figure 53** Define C-Systems panel (M,P,C,C)

1. The System Name, SMFID, IPL Parameters, CPC, and LPAR fields of the Define C-Systems panel are automatically discovered using CSYSSITE parameter values set during installation, as described in "Install GDDRPARM file" on page 145.

   The color font that displays in these fields indicates the precedence rules that apply to data returned to the GDDR Parameter Wizard from the Auto-Discovery feature when validated against the contents of your parameter work dataset.

   - Green—Display only. Auto-Discovery had no conflict with input.
   - Red—User input allowed. Not restricted by Auto-Discovery, or impossible to discover at this time.
   - Yellow—Display only. Restricted by Auto-Discovery, conflict with input.

   ---

   **Note:** You can only change values displayed in red font via the panel. Within the Define C-Systems panel, the only values you can change are the IPL parameters.

   ---

   The values that are displayed may be confirmed or changed using the information provided with each field described below, subject to the Auto-Discovery data precedence rules described above.

   - Site

     Indicates the ID of the site location being specified. Site values are populated from the Site List in the Define GDDR Configuration Features panel. Valid values are DC1, DC2, DC3, or DC4. This field reflects the CSYSSITE parameter for the listed C-system.

   - System Name

     The MVS system name of a C-system which is specified using the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or equivalent parameter file. This field reflects the CSYSSITE parameter for the listed C-system.

   - SMFID

     The SMFID of the C-system identified by System name. This field cannot be edited, unless Auto-Discovery fails for the listed remote C-system.

   - IPL Parameters

     Specifies the IPL parameters that GDDR may use to IPL the C-systems at sites DC1, DC2, DC3, or DC4 in the following format:

     *ssss*,*iiiixxmn*

     Where

     – *ssss* is the Sysres device address
     – *iiii* is the IODF device address
     – *xx* is the LOADxx member suffix
     – *m* is the IMSI Field
     – *n* is the IEANUC0n suffix

You can locate IPL parameters using the D IPLINFO MVS console command, as shown in the following sample output:

```
RESPONSE=SYSB
IEE254I 21.14.59 IPLINFO DISPLAY 860
SYSTEM IPLED AT 15.59.20 ON 02/02/2018
RELEASE z/OS 01.10.00 LICENSE = z/OSe
USED LOAD02 IN SYS1.IPLPARM ON 0A26
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = B0
IEASYS LIST = B0 (OP)
IODF DEVICE 0A26
IPL DEVICE 0A1D VOLUME RESB14
```

- CPC

  Specify the name of the central processor complex where the C-system z/OS system is defined. This field cannot be edited, as it is fully determined by the CPC parameter of type NETID for the listed C-system.

- LPAR

  Specify the name of the logical partition within which this system runs at the specified site. This field cannot be edited, unless Auto-Discovery fails for the listed remote C-system. You can find the CPC and LPAR names using the D M=CPU MVS console command, as shown in the following sample output:

```
RESPONSE=SYSB
IEE174I 20.59.34 DISPLAY M 781
PROCESSOR STATUS
ID  CPU                 SERIAL
0                       02F94E2096
1                       02F94E2096
2                       02F94E2096


 CPC ND = 002096.S07.IBM.83.00000008F94E
 CPC SI = 2096.N03.IBM.83.000000000008F94E
 CPC ID = 00
 CPC NAME = C
 LP NAME = ZOSESYSB LP ID = 2
 CSS ID = 0
 MIF ID = 2

   ONLINE - OFFLINE . DOES NOT EXIST W WLM-MANAGED
N NOT AVAILABLE

 CPC ND CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
 CPC SI SYSTEM INFORMATION FROM STSI INSTRUCTION
 CPC ID CENTRAL PROCESSING COMPLEX IDENTIFIER
 CPC NAME CENTRAL PROCESSING COMPLEX NAME
 LP NAME LOGICAL PARTITION NAME
```

2. Type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Define GDDR Datasets panel (M,P, C,D).

## Define GDDR datasets (M,P,C,D)

When you specify option **D** in the Define Configuration Basics panel (M,P,C), the Define GDDR Datasets panel is displayed:

```
------------------------ GDDR - Define GDDR Datasets ------ Row 1 to 16 of 16
 Command ===>

 Press <F3> to return to the Define Configuration Basics Menu
 <F5> Define C-Systems                  <F6> Define Site Roles
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat

 CMD  C-system    DS Type    Seq   Dataset Name
 ---  --------    --------   ---   -------------------------------------------
  _   SYS1        BKUPVARS         GDDR.GDDR520.PARMS.BKUP
  _   SYS2        BKUPVARS         GDDR.GDDR520.PARMS.BKUP
  _   SYS3        BKUPVARS         GDDR.GDDR520.PARMS.BKUP
  _   SYS4        BKUPVARS         GDDR.GDDR520.PARMS.BKUP
  _   SYS1        LAPD             GDDR.GDDR520.PARMS.LAPD
  _   SYS2        LAPD             GDDR.GDDR520.PARMS.LAPD
  _   SYS3        LAPD             GDDR.GDDR520.PARMS.LAPD
  _   SYS4        LAPD             GDDR.GDDR520.PARMS.LAPD
  _   SYS1        PROCLIB     1    GDDR.GDDR520.CUSTOM.LIB
  _   SYS2        PROCLIB     1    GDDR.GDDR520.CUSTOM.LIB
  _   SYS3        PROCLIB     1    GDDR.GDDR520.CUSTOM.LIB
  _   SYS4        PROCLIB     1    GDDR.GDDR520.CUSTOM.LIB
  _   SYS1        SKELETON    1    GDDR.GDDR520.ISPSLIB
  _   SYS2        SKELETON    1    GDDR.GDDR520.ISPSLIB
  _   SYS3        SKELETON    1    GDDR.GDDR520.ISPSLIB
  _   SYS4        SKELETON    1    GDDR.GDDR520.ISPSLIB
 ****************************** Bottom of data ********************************
```

**Figure 54** Define GDDR Datasets panel (M,P,C,D)

The Define GDDR Datasets panel lets you specify the GDDR datasets, as described in "Defining GDDR datasets" on page 307.

### Defining GDDR datasets

To define GDDR datasets:

1. Type **Add** in the Option field at the top of the panel to display a new line for entry of values.
2. Insert additional datasets by typing **R** (Repeat) in the CMD field to the left of an existing entry and overtyping the existing data.
3. Type **D** (Delete) in the CMD field to the left of an existing entry to remove it.
4. Complete the required information for each dataset type:
   - BKUPVARS

     Specify the name of the dataset into which the backups produced by GDDR Heartbeat Monitor initialization should be saved. This dataset was allocated during the procedure described in "Define parameter management datasets" on page 139.
   - LAPD

     Specify the name of the last activated parameter dataset. This dataset was allocated during the procedure described in "Define parameter management datasets" on page 139. It cannot be the same dataset as your current work dataset.

- PROCLIB

  Specify the name of the library on the C-systems which contains the GDDRPROC member. This library was selected during the procedure described in "Customize PROCLIB member GDDRPROC" on page 150.

- SKELETON

  Specify the name of the GDDR ISPSLIB library that resulted from your SMP/E installation.

- Seq

  Specify a sequence number for the dataset. The sequence number enables the PROCLIB and skeleton libraries to be assigned sequentially across C-systems.

5. If you made any changes, type **Save** on the command line and press **Enter**.
6. Press **F6** to go to the Define Site Roles and Groups panel (M,P,C,R).

## Define site roles and groups (M,P,C,R)

When you specify option **R** in the Define Configuration Basics panel (M,P,C), the Define Site Roles and Groups panel is displayed, similar to the following:

```
------------------- GDD1 - Define Site roles and groups -----------------------
Command ===>

Press <F3> to return to the Define Configuration Basics Menu
<F5> Define GDDR Data Sets              <F6> Define Data Storage Objects
Type SAVE to save, LOAD to restart from last saved values.

Enter the required information. Press <Enter> to Validate.
                                                                More:     +
                 Primary DASD Site:  DC1          Select DC1 DC2 DC3 or DC4
                      Primary Site:  DC1

        Consistency Group Name DC1:  SITEASQ1     Select DC1 DC2 DC3 or DC4
        Consistency Group Name DC2:  SITEBSQ1
        Consistency Group Name DC3:  SITECSQ1
        Consistency Group Name DC4:  SITEDSQ1

                MSC Group DC1->DC3:  SQA1_A_1
                MSC Group DC2->DC4:  SQA1_B_1
                MSC Group DC3->DC1:  SQA1_A_2
                MSC Group DC4->DC2:  SQA1_B_2
****************************** Bottom of Data ******************************
```

**Figure 55** Define Site roles and groups panel (M,P,C,R)

1. Complete the following fields in the panel:

   - Primary DASD Site

     Specify the site where the source (R1) DASD is located.

   - Primary Site

     Specify the site where the production workload is located.

   - Consistency Group Name DC1/DC2/DC3/DC4

     Specify the name of the managed consistency group when the Primary Site is the indicated site (DC1, DC2, DC3, or DC4).

In an SRDF/SQAR with AutoSwap environment, the consistency group and AutoSwap group share the same name when the consistency group is enabled with CAX protection.

- MSC Group DC*n*->DC*m*

  These fields show the MSC group names defined using the MSCGROUP parameter in the GDDRPARM file. The fields are read-only.

---

**Note:** If the RESET option is specified during GDDR parameter activation, GDDR dynamically determines the correct primary DASD site based on a discovery process of the managed configuration; this may override the value specified on this panel.

---

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Define Data Storage Objects panel (M,P,D).

## Define data storage objects (M,P,D)

When you specify option **D** in the Parameter Management Options Menu panel (M,P), the Define Data Storage Objects panel is displayed:

```
---------------------- GDDR - Define Data Storage Objects ---------------------
Option ===>

    S   Define SRDF Device ranges                          This System: LB01M34
    T   Define TimeFinder Device ranges                      This Site: DC2
    N   Define GDDR Snapshot Names                         This region: 1
    V   Define GDDR Managed VDG Names                         Master-C: LB01M34
    G   Define SDDF Gatekeepers                            Primary Site: DC1
    D   Define DLm Systems                                 Primary DASD: DC1
                                                             Automation: ON
                                                       Planned script: None
                                                     Unplanned script: None




Select an option and press <Enter>


<F5> Define Site Roles                  <F6> Define SRDF Device Ranges
Press <F3> to return to the Parameter Management Options Menu
```

**Figure 56** Define Data Storage Objects panel (M,P,D)

Complete the following tasks to define your data storage objects:

- Define SRDF device ranges (M,P,D,S)
- Define TimeFinder device ranges (M,P,D,T)
- Define GDDR snapshot names (M,P,D,N)
- Define GDDR-managed VDGs (M,P,D,V)
- Define SDDF gatekeepers (M,P,D,G)
- Define DLm systems (M,P,D,D)

## Define SRDF device ranges (M,P,D,S)

When you specify option **S** in the Define Data Storage Objects panel (M,P,D), the Define SRDF Device Ranges panel is displayed, similar to the following:

```
---------------------- GDDR - Define SRDF Device Ranges -------------------------
 Command ===>

 Press <F3> to return to the Define Data Storage Objects Menu
 <F5> Define Data Storage Objects        <F6> Define TimeFinder Device Ranges
 Type <ADD> to add row, <SAVE> to save, <LOAD> restart from last saved values.

 Line commands: A dd, D elete, R epeat

                                SRDF Device Range
            <===========SITE1===========> <===========SITE2===========>
Sel SitePair GK  RDFgrp   Start -  End      Start -  End    RDFgrp  GK  EXT DLM
--- ------- ---- ------ -------- -------- -------- -------- ------ ---- --- ---
 _  DC1-DC3 2715     D2      385      394     1435     14AD     D2 D345  NO YES
 _  DC1-DC3 2716     D2      395      3A4     14B5     152D     D2 D346  NO YES
 _  DC1-DC3 7D40     28      80A      82F      80A      82F     28 3C40  NO NO
 _  DC1-DC3 7D41     28      830      839      4B0      4B9     28 3C41  NO NO
 _  DC1-DC3 7D42     28     38A1     38CC     38A1     38CC     28 3C42  NO NO
 _  DC2-DC4 1730     D3     6EAD     6EAE     6B71     6B72     D3 18D0  NO YES
 _  DC2-DC4 1731     D3     6EB2     6EB3     6B76     6B77     D3 18D1  NO YES
 _
...
****************************** Bottom of data ********************************
```

**Figure 57** Define SRDF Device Ranges panel (M,P,D,S)

If the panel is not pre-populated with existing SRDF device range entries, run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 to discover all SRDF devices to be managed by GDDR, both consistency-protected and 'external' devices. Alternatively, complete the following steps to manually define the SRDF device configuration to GDDR.

1. Type **Add** in the command line at the top of the panel to display a new line for entry of SitePair and SRDF Device Range values.
2. Insert additional Site Pairs by typing **R** (Repeat) in the Sel field to the left of an existing SitePair.
3. Complete the fields as follows:
   - SitePair

     Specify the GDDR SRDF replication pairs in the format:

     DC*m*-DC*n*

     Where *m* is the lower site number, and *n* is the higher site number. Valid pairings of DC*m* and DC*n* are DC1-DC2, DC1-DC3, DC2-DC4 and DC3-DC4.
   - SRDF Device Range

     Specifies contiguous PowerMax/VMAX device ranges which are common to the specified SitePair for SITE1 and SITE2:

     – GK

       SRDF gatekeeper.

     – RDFgrp

       The SRDF group spanning the specified site pair and to which the defined device ranges belong.

– Start - End

A range of contiguous PowerMax/VMAX devices at the corresponding DC site.

The 4-character gatekeeper masks represent MVS volume addresses used to uniquely identify the site DC*m* or DC*n* storage system where the volume [gatekeeper] resides. The gatekeepers associated with the SRDF device ranges provide information used in the processing of commands: the site, the storage system at a site, and the gatekeeper address. All MVS addresses used here MUST be specified on SYMM parameters in the GDDRPARM file. The addresses must be unique within a site.

The PowerMax/VMAX device number ranges at DC*m* and DC*n* must contain the same number of devices.

If a site is not configured with MVS systems, then dummy MVS addresses must be specified as gatekeeper devices for that site.

- EXT (External)

  YES or NO indicates whether these devices are external to the consistency-protected environment.

- DLM

  **YES** in the DLM column indicates that a given SRDF device range is also a DLm backend device range.

4. If you made any changes, type **Save** on the command line and press **Enter**.
5. Press **F6** to go to the Define TimeFinder Device Ranges panel (M,P,D,T).

## Define TimeFinder device ranges (M,P,D,T)

When you specify option **T** in the Define Data Storage Objects panel (M,P,D), the Define TimeFinder Device Ranges panel is displayed.

The Define TimeFinder Device Ranges panel is pre-populated with values obtained during a GDDRACDD run. The GDDRACDD utility generates parameters for all BCVs found in selected storage systems and generates member E05TFDEV in gatekeeper order.

```
-------------------- GDDR - Define TimeFinder Device Ranges  Row 1 to 13 of 20
Command ===>

Press <F3> to return to the GDDR Define Data Storage Objects Menu
<F5> Define SRDF Device Ranges          <F6> Define GDDR Snapshot Names
Type <LOAD> to restart from last saved values.

            TimeFinder Device Range  STD Device Range
      Site   GKDN   Start -  End       Start -  End      SET   TYPE  HOST
      ----   ---- -------- --------  -------- --------  ----  ----  ----
      DC1    7010     29B0     29EF      016B     01AA  GOLD   INT     C
      DC2    8010     2A40     2A7F      0170     01AF  GOLD   INT     C
      DC2    8010     2BAA     2BB9      033A     0349  GOLD   INT     C
      DC2    8010     2CAA     2CB9      093A     0949  GOLD   INT     C
      DC3    5410     29F0     2A2F      0170     01AF  GOLD   INT     C
      DC3    5410     2F7F     307E      0F20     101F  GOLD   INT     C
      DC3    5410     307F     3084      1020     1025  GOLD   INT     C
      DC3    5410     30BB     3142      1A0E     1A95  GOLD   INT     C
      DC3    5410     3166     31CF      1A96     1AFF  GOLD   INT     C
      DC3    5410     2A30     2A3F      093A     0949  GOLD   INT     C
      DC3    5410     2B85     2B94      033A     0349  GOLD   INT     C
      DC3    5410     2BB1     2C84      193A     1A0D  GOLD   INT     C
      DC3    5410     2C85     2D4A      0ADA     0B9F  GOLD   INT     C
      DC3    5410     2D4B     2E4A      0BBA     0CB9  GOLD   INT     C
      DC3    5410     2E4B     2E50      0CBA     0CBF  GOLD   INT     C
      DC3    5410     2E51     2F50      0DD2     0ED1  GOLD   INT     C
      DC3    5410     2F51     2F7E      0ED2     0EFF  GOLD   INT     C
      DC3    5410     2BA5     2BB0      036A     0375  GOLD   EXT     C
****************************** Bottom of data ******************************
```

**Figure 58** Define TimeFinder Device Ranges panel (M,P,D,T)

1. If the panel is not pre-populated, run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 to discover all TimeFinder devices to be managed by GDDR, both consistency protected and 'external' devices.

   If only a subset of BCVs are GDDR-managed, use the standard ISPF edit to modify the GDDR Parameter Wizard work dataset, member E05TFDEV. Complete the change with the GDDR Parameter Wizard validation and activation steps.

   ---

   **Note:** Editing of the GDDR Parameter Wizard work dataset outside of the GDDR Parameter Wizard is best done under guidance of the GDDR Solution Support team.

   ---

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Define GDDR Snapshot Names panel (M,P,D,N).

## Define GDDR snapshot names (M,P,D,N)

When you specify option **N** in the Define Data Storage Objects panel (M,P,D), the Define GDDR Snapshot Names panel is displayed:

```
-------------------- GDDR - Define GDDR Snapshot Names ------------------------
Command ===>

Press <F3> to return to the Define Data Storage Objects Menu
<F5> Define TimeFinder Device Ranges    <F6> Define GDDR Managed VDG Names
Type <SAVE> to save, <LOAD> to restart from last saved values.

 Specify a 1-23 character name for each of the following sets,
 so that, in combination with the predefined GDDR prefix of
 GDDR_INT_ or GDDR_EXT_ a unique identifier is created for each
 Snapshot

   GOLD-Internal set: GDDR_INT_ GOLDABCD_______________
   GOLD-External set: GDDR_EXT_ GOLDABCD_______________

   TEST-Internal set: GDDR_INT_ TESTABCD_______________
   TEST-External set: GDDR_EXT_ TESTABCD_______________
```

**Figure 59** Define GDDR Snapshot Names panel (M,P,D,N)

Specify the snapshot names to be used by GDDR, one per set.

- Use GDDR_INT_* *snapshotname* for devices in or paired with consistency-protected SRDF groups (ConGroup and/or MSC).
- Use GDDR_EXT_* *snapshotname* for devices in or paired with ADCOPY-DISK SRDF groups.

## Define GDDR-managed VDGs (M,P,D,V)

When you specify option **V** in the Define Data Storage Objects panel (M,P,D), the Define GDDR Managed VDG Names panel is displayed:

```
---------------------- GDDR - Define GDDR Managed VDG Names --- Row 1 to 1 of 1
Command ===>

Press <F3> to return to the Define Data Storage Objects Menu
<F5> Define GDDR Snapshot Names         <F6> Define SDDF GateKeepers
Type SAVE to save, LOAD to restart from last saved values.

Line commands: A dd, D elete, R epeat

CMD  Site      VDG Name
---  ----   ---------------
_    DCX    ENTER_VDG_NAME
****************************** Bottom of data ******************************
```

**Figure 60** Define GDDR Managed VDG Names panel (M,P,D,V)

The Define GDDR Managed VDG Names panel allows you to define VDG names for each DASD site in the configuration. These groups will be STARTed at the end of scripts, even if they were not PAUSEd by the script. It is recommended to define all VDGs usually active at your installation on this panel.

- Site

  Identify the storage location of the GDDR-managed SRDF devices protected by zDP.

- VDG Name

  Specify the VDG names that you want GDDR to bring in an Active state at the end of GDDR scripts, even if the script did not pause these groups.

## Define SDDF gatekeepers (M,P,D,G)

When you specify option **G** in the Define Data Storage Objects panel (M,P,D), the Define SDDF GateKeepers panel is displayed:

```
----------------- GDDR - Define SDDF GateKeepers --------------- Row 1 to 3 of 3
Command ===>

Press <F3> to return to the Define Data Storage Objects Menu
 <F5> Define GDDR Managed VDG Names      <F6> Define DLm Systems
 Type <SAVE> to save, <LOAD> to restart from last saved values,
      <ADD> to add a new entry.

Line commands: A dd, D elete, R epeat

CMD   Site     GK
---   ----     ----
 _     DC1      AA1F
 _     DC2      B21F
 _     DC3      BA1F
 _     DC4      DA1F
****************************** Bottom of data ********************************
```

**Figure 61** Define SDDF GateKeepers panel (M,P,D,G)

The SDDF gatekeeper parameters are not used in the SRDF/SQAR with AutoSwap configuration. Press **F6** to go to the Define DLm Systems panel (M,P,D,D).

## Define DLm systems (M,P,D,D)

When you specify option **D** in the Define Data Storage Objects panel (M,P,D), the Define DLm Systems panel is displayed:

```
------------------------ GDDR - Define DLm Systems ----------- Row 1 to 4 of 4
 Command ===>

 Press <F3> to return to the Define Data Storage Objects Menu
 <F5> Define SDDF GateKeepers           <F6> Define Host Objects
 Type <SAVE> to save, <LOAD> to restart from last saved values.

 Line commands: A dd, D elete, R epeat, S erialNos

    Suspend DLm heartbeat ==> 0240 (In minutes, 1440 max value)

 Valid Types are: NAS NAS/RDF, SLV NAS/RDF Slave, STLS Stateless

 Sel Site DLm Name  Type    IP address (IPv4 or IPv6)       Port
 --- ---- --------  ------  ------------------------------  -----
  _   DC1  DLMSITE1  NAS     nnn.nnn.nnn.nnn                 9050
  _   DC2  DLMSITE2  NAS     nnn.nnn.nnn.nnn                 9050
  _   DC3  DLMSITE3  NAS     nnn.nnn.nnn.nnn                 9050
  _   DC4  DLMSITE4  NAS     nnn.nnn.nnn.nnn                 9050
 ****************************** Bottom of data ********************************
```

**Figure 62** Define DLm Systems panel (M,P,D,D)

The Define DLm Systems panel allows you to specify DLm systems to be managed by GDDR.

**Note:** See "DLm integration using DLMDR" on page 72 for information about DLm integration.

Each entry on the Define DLm Systems panel represents a single DLm ACP (Access Control Point). Each ACP can control multiple VTECs (virtual tape emulation controllers).

Use **A**, **D**, **R**, or **S** in the Sel column to add, delete, repeat ACP entries, or to select an ACP for FICON adapter serial number auto-discovery.

**Note:** "Discovering ACPs" on page 316 and "Setting DEVSTATUS QUIESCE and VARY options" on page 316 provide additional information.

The Suspend DLm heartbeat field shows after how many minutes the DLm heartbeat function will be automatically resumed after it has been suspended.

**Adding or editing ACPs**

When adding or editing ACP entries, use the following fields:

- Site

  This is the physical site location of this DLm system.

- DLm Name

  A name you assign to this DLm system.

- Type

  The DLm type:

  - `NAS`—DLm uses PowerMax/VMAX devices and is the only GDDR-managed DLm system or, if a site has multiple GDDR-managed DLm systems, it is the last DLm system quiesced and the first DLm system started during a site swap.
  - `SLV`—DLm uses PowerMax/VMAX devices and, when a site has more than one GDDR-managed DLm systems, it is the among the first DLm system or systems quiesced or the last DLm system or systems started during a site swap.
  - `STLS`—DLm uses PowerMax/VMAX devices and is configured for Stateless switchover and recovery.

  For the listed DLM types, there must be a DLm defined at a minimum of two sites in the configuration.

- IP address and Port

  This is the IP address and port GDDR will use to communicate with this DLm system.

### Discovering ACPs

When selecting an ACP using the S line command, GDDR attempts to contact the DLm ACP station using TCP/IP at the IP address and IP port shown, to query the ACP for the serial numbers of all FICON adapters on all attached VTEs. The ACP Details panel is displayed:

```
+---------------------- ACP Details ----------------------+
| Command ===>                                            |
|                                                         |
| Site    IP address (IPv4 or IPv6)       Port    Type    |
| ----  ------------------------------   -----   ------   |
|  DC1  nnn.nnn.nnn.nnn                  9060     STLS     |
|                                                         |
| ............. FICON adapter serial numbers ............ |
|  28230851 28230852                                      |
|                                                         |
|                                                         |
|                                                         |
|                                                         |
|                                                         |
|  Press <F3> to return                                   |
|                                                         |
+---------------------------------------------------------+
+-------------------------------------------------------------------------+
| DLm auto-discovery found 2 FICON adapter serial numbers. List refreshed. |
+-------------------------------------------------------------------------+
```

**Figure 63** ACP Details panel

DLm FICON adapter serial number auto-discovery usually takes a few seconds to a minute. However, if the ACP is not responding or incorrect TCP/IP information was given (IP address and/or port number), this could take several minutes before a (non) response is detected. If the auto-discovery is successful, the list of discovered FICON adapter serial numbers is displayed in the ACP Details panel shown in Figure 63.

If the auto-discovery fails, the S line command also presents the ACP Details panel shown in Figure 63.

### Setting DEVSTATUS QUIESCE and VARY options

When you choose the SAVE command in the Define DLm Systems panel (M,P,D,D) and any of the DLm have a type of NAS, SLV or STLS, the DLm QUIESCE ands VARY panel is displayed:

```
+------------- GDDR - DLm QUIESCE and VARY --------------+
|     NAS-RDF/Stateless QUIESCE and VARY parameters      |
| Issue DEVSTATUS QUIESCE: N    (Y or N)                 |
|    Vary online applies to Autoswitched drives only     |
| Vary online : NONE    (PRIMARY, BOTH or NONE)          |
|       Vary offline applies to any online drives.       |
| Vary offline: NONE    (PRIMARY, BOTH or NONE)          |
| ENTER to Continue            <PF3> to Cancel           |
+---------------------------------------------------------+
```

**Figure 64** DLm QUIESCE and VARY panel

The DLm QUIESCE and VARY panel provides the following options:

- Issue DEVSTATUS QUIESCE

  The DEVSTATUS QUIESCE field specifies whether GDDR scripts will issue a DEVSTATUS QUIESCE to the DLm ACP (Y) or (N).

- Vary online

  PRIMARY causes GDDR scripts to vary online only the managed tape drives set to AUTOSWITCH at the primary site. You must manually vary online GDDR-managed tape drives at the secondary site, if so desired.

  BOTH causes GDDR scripts to vary online managed tape drives set to AUTOSWITCH at the primary and the secondary site.

  AUTOSWITCH causes GDDR scripts to vary online only those managed tape devices that have been set to AUTOSWITCH in z/OS.

- Vary offline

  PRIMARY causes GDDR scripts to vary offline only the managed tape drives at the primary site. You must manually vary offline GDDR-managed tape drives at the secondary site, if so desired.

  BOTH causes GDDR scripts to vary offline managed tape drives at the primary and the secondary site.

  NONE causes GDDR scripts to bypass vary online AUTOSWITCH and vary offline the managed tape drives. These operations are expected to be performed outside of GDDR control.

## Define host objects (M,P,H)

When you specify option **H** in the Parameter Management Options Menu panel (M,P), the Define Host Objects panel is displayed:

```
------------------------- GDDR - Define Host Objects -------------------------
Option ===>

    S   Define Managed Systems                            This System: LB01M34
    SP  Define System IPL Priorities                        This Site: DC2
    L   Define Managed LPARs                               This region: 1
    R   Define System Recovery Attributess                    Master-C: LB01M34
    P   Define Managed CPC                                Primary Site: DC1
    I   Define IPL Parameters                             Primary DASD: DC1
    A   Define HMC Load Activation Profiles                 Automation: ON
                                                         Planned script: GDDRPA21
    D   Define Managed Couple Datasets                 Unplanned script: None
    CF  Define Managed CF Structures
    W   Define External Workloads
    E   Define Dell EMC Mainframe Enabler STCs

Select an option and press <Enter>

<F5> Define DLm Systems                <F6> Define Managed Systems
Press <F3> to return to the Parameter Management Options Menu
```

**Figure 65** Define Host Objects panel (M,P,H)

Complete the following tasks to define your host objects configuration:

- Define managed systems (M,P,H,S)
- Define system IPL priorities (M,P,H,SP)
- Define managed LPARs (M,P,H,L)
- Define system recovery attributes (M,P,H,R)

- Define managed CPCs (M,P,H,P)
- Define IPL parameters (M,P,H,I)
- Define HMC load activation profiles (M,P,H,A)
- Define managed coupled datasets (M,P,H,D)
- Define managed CF structures (M,P,H,CF)
- Define external workloads (M,P,H,W)
- Define Mainframe Enablers STCs (M,P,H,E)

## Define managed systems (M,P,H,S)

When you specify option **S** in the Define Host Objects panel (M,P,H), the Define Managed Systems panel is displayed:

```
----------------------- GDDR - Define Managed Systems ------- Row 1 to 14 of 14
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define Host Objects                <F6> Define System IPL Priorities
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat

                                                 Manage     HMC      CF
 CMD  Site    System      Sysplex              Workload    Only    LPAR
 ---   ---   --------   -------------          --------    ----    ----
  _     DC1   PRD1       PROD,PLEX               YES         NO      NO
  _     DC1   TST1       _____________           YES         NO      NO
  _     DC1   TST2       _____________           YES         NO      NO
  _     DC2   CNT1       PROD,PLEX               YES         NO      NO
  _     DC1   TST5       _____________            NO         NO      NO
  _     DC1   TST6       _____________            NO         NO      NO
  _     DC2   TST3       N/A                      NO        YES      NO
  _     DC3   PRD3       PROD,PLEX               YES         NO      NO
  _     DC3   TST3       _____________           YES         NO      NO
  _     DC3   TST7       _____________           YES         NO      NO
  _     DC4   CNT2       PROD,PLEX               YES         NO      NO
  _     DC3   TST8       _____________            NO         NO      NO
  _     DC3   TST9       _____________            NO         NO      NO
  _     DC4   TSTA       N/A                      NO        YES      NO
 ****************************** Bottom of data ********************************
```

**Figure 66** Define Managed Systems panel (M,P,H,S)

IMPORTANT

Changes to this panel are likely to require changes to COMM or CPC parameters in the GDDRPARM file. If necessary, such changes must be completed before using the GDDR Parameter Wizard.

Use the Define Managed Systems panel to define mainframe systems to GDDR and indicate the extent to which they will be managed by GDDR, either using the online interface or during automation sequences. Do not include C-systems here, as those have been defined to GDDR earlier as described in "Define C-systems (M,P,C,C)" on page 304 and are defined in GDDRPARM. Managed system parameters are defined for every managed system to enable HMC functions and workload management.

The first time you use this panel (when you have never issued a SAVE command), the Auto-Discovery feature will populate this panel with managed system names along with related sysplexes, if available. The auto-discovered list of systems is determined

by the presence of GDDRPARM COMM parameters for non C-systems. After you have issued a SAVE command, the previously saved information will be displayed on the panel whenever you return to it.

If changes to GDDRPARM parameters are made as a result of the various maintenance procedures described in Chapter 10, "Maintaining GDDR Environment," those changes will be displayed on re-entry to this panel.

1. If the panel is not pre-populated with existing entries, type **Add** in the command line at the top of the panel to display a new line for field entry.
2. Insert systems by entering **R** (Repeat) in the CMD field to the left of an existing entry and overtype the existing data.
3. Confirm or change the values that are displayed by using the information provided with each field described below, subject to the precedence rules that apply to data returned to the GDDR Parameter Wizard by the Auto-Discovery feature:

   - Site

     Specify the ID of the site location being specified. It can have the value DC1, DC2, DC3, or DC4.

   - System

     Specify the z/OS system name of a managed system which resides at the specified site. You can find the system name on the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or the equivalent parameter file.

   - Sysplex (optional)

     When defining managed systems to GDDR, specify a sysplex name for those systems where coupling facility structure and coupling facility dataset management actions are to be performed by GDDR.

     Coupling facility dataset and structure management is controlled by the Rebuild CF Dataset, Rebuild CF Structure, and the CF Rebuild Timeout fields in the Script Sysplex Options panel (M,P,O,S), by selections made in the Define Managed Couple Datasets panel (M,P,H,D), and in the Define Managed CF Structures panel (M,P,H,CF).

     The format of this field is: <*name*>,<*type*>, where:

     – *name* is either the name of the Sysplex to which the defined system belongs or "NONE".

     – If a name is specified then *type* is either "PLEX", "MONO", or "NONE".

     This entry is used to simplify data entry on the Sysplex Object management related panels of the GDDR Parameter Wizard.

     ---

     **Note:** The Sysplex Object Management features are enabled only for systems which have been defined here with <*name*>,"PLEX".

     ---

- Manage Workload

  Indicates whether GDDR will trigger start/stop of the workload for the listed system. Type **YES** to cause GDDR to trigger the stop and start of applications when GDDR takes planned or unplanned actions that impact the managed systems.

- HMC Only

  Indicates whether the specified system is external to the population of systems managed by GDDR scripts. HMC Only 'YES' Systems may be managed using the Perform HMC LPAR Actions panel (A,L). Type **YES** to bypass GDDR parameter validation for the specified system. Type **NO** to specify that the system is to be managed by GDDR.

- CF LPAR

  The CF LPAR setting is used to designate a dummy system name that will be associated with a coupling facility LPAR on the Define Managed LPARs panel (M,P,H,L). Standalone coupling facilities are not managed by GDDR.

  **Note:** CF LPARs are not required at any site, and if present at any site, are not necessarily present at the other sites, as standalone coupling facilities may be defined at the other sites.

4. If you made any changes, type **Save** on the command line and press **Enter**.
5. Press **F6** to go to the Define Managed LPARs panel (M,P,H,L).

## Define system IPL priorities (M,P,H,SP)

When you specify option **SP** in the Define Host Objects panel shown in Figure 65 on page 317, the following panel is displayed:

```
---------------------- GDDR - Define System IPL Priorities ----- Row 1 to 14 of 14
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define Managed Systems             <F6> Define Managed LPARs
 Type SAVE to save, LOAD to restart from last saved values.

 Enter the required information. Press <Enter> to Validate.

                         IPL
      Site    System    Priority       IPL Group Description
       ---    --------  --------    ----------------------------
       DC1    PRD1         00       Prod Group 1 - Top Priority
       DC1    TST1         90       Test Group 1
       DC1    TST2         90       Test Group 1
       DC1    TST6         98       Test Group 3
       DC1    TST5         98       Test Group 3
       DC2    CNT1         02       Prod Group 2
       DC2    TST3         95       Test Group 2
       DC3    PRD3         00       Prod Group 1 - Top Priority
       DC3    TST3         98       Test Group 3
       DC3    TST7         98       Test Group 3
       DC3    TST8         99       Test Group 4 Lowest Priority
       DC3    TST9         99       Test Group 4 Lowest Priority
       DC4    CNT2         00       Prod Group 1 - Top Priority
       DC4    TSTA         90       Test Group 1
 ****************************** Bottom of data ******************************
```

**Figure 67** Define System IPL Priorities panel (M,P,H,SP)

This panel lists systems eligible for IPL and their home site. This panel is populated from the systems on the Define Managed Systems panel (M,P,H,S), excluding:

- Excluded systems (MANAGE WORKLOAD column=NO)
- HMC ONLY systems (HMC ONLY column=YES)
- CF LPAR systems (CF LPAR column=YES).

The managed systems shown on the Define System IPL Priorities panel (M,P,H,SP) are maintained in the Define Managed Systems panel (M,P,H,S). Each system can be assigned a priority from 00 to 99, with 00 being the highest IPL priority (systems that will be IPLed or ACTIVATEd first), and 99 being the lowest IPL priority (systems that will be IPLed or ACTIVATEd last).

During scripts that perform HMC actions, the list of systems will be divided into groups by priority, and groups are processed in the following manner:

1. Inform the user with a list of target systems.
2. Confirm all actions with a single WTOR.

   Depending on the Confirm GDDR Script HMC actions by WTOR call override setting, other WTORs may be issued.
3. ACTIVATE all systems, then LOAD CLEAR them (or RESET CLEAR all systems, then DEACTIVATE them).

The following fields are provided on the Define System IPL Priorities panel (M,P,H,SP):

- Site

  The home site ID for this system.
- System

  The name of the system as defined on the Define Managed Systems panel (M,P,H,S).
- IPL Priority

  A number between 00 (highest) and 99 (lowest) indicating the priority of system during HMC actions.

  - For ACTIVATEs (IPLs), CF LPARs come first, followed by managed systems in order from 00 to 99.
  - For DEACTIVATEs (shutdowns), systems are shutdown in order from 99 to 00, CF LPARs are last.
  - The order of IPL/shutdown for systems with the SAME priority is undetermined.

  You can (re)assign a priority simply by typing a new number into this column, and clearing the Description column.
- IPL Group Description

  A 1-30 character description for a given priority group (all systems with the same priority number).

To change an IPL Group Description, make sure you have entered the correct number in the IPL Priority column, then simply type a new description and press the **Enter** key.

Clearing the Description column and pressing the **Enter** key causes the Description field to be filled in with any existing description for the given IPL Priority value. If you do not clear the Description column when assigning a new priority, you will also be changing the IPL Group Description for the new priority.

**Sorting the display**

You can sort the display by typing **SORT** in the command line at the top of the panel:

SORT [ASC|DESC]*colname* ...

where *colname* is the name of the column to sort. Valid column names are:

- SITE
- SYSTEM
- PRIORITY
- DESC

The optional prefixes ASC (default) or DESC cause sorting in ascending or descending order, respectively. For example, the following commands sort the display by ascending site and descending priority and ascending system name:

```
SORT ASC SITE DESC PRIORITY ASC SYSTEM
```

or

```
SORT SITE DESC PRIORITY SYSTEM
```

## Define managed LPARs (M,P,H,L)

When you specify option **L** in the Define Host Objects panel (M,P,H), the Define Managed LPARs panel is displayed:

```
------------------------- GDDR - Define Managed LPARs ------- Row 1 to 11 of 11
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define System IPL Priorities      <F6> Define System Recovery Attributes
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat
                                                                    Bypass
 CMD   Site    System        CPC (netid.nau)  LPAR        HMC
 ---    ---    --------     -----------------  --------    ---
  _      DC1   PRD1         IBM390PS.PRD       LPARPRD1     NO
  _      DC1   TST1         IBM390PS.TST1      LPARTST1     NO
  _      DC1   TST2         IBM390PS.TST1      LPARTST2    YES
  _      DC1   TST5         IBM390PS.TST1      LPARTST5    YES
  _      DC1   TST6         IBM390PS.TST6      LPARTST6    YES
  _      DC2   CNT1         IBM390PS.CNT1      LPARCNT1     NO
  _      DC2   TST3         IBM390PS.TST3      LPARTST3     NO
  _      DC3   PRD3         IBM390PS.CP3       LPARPRD3     NO
  _      DC3   TST7         IBM390PS.CP3       LPARTST7     NO
  _      DC4   CNT2         IBM390PS.CNT       LPARCNT2     NO
  _      DC4   TSTA         IBM390PS.TST       LPARTSTA     NO

 ****************************** Bottom of data ********************************
```

**Figure 68** Define Managed LPARs panel (M,P,H,L)

**Note:** Changes to this panel are likely to require changes to COMM or CPC parameters in the GDDRPARM file. If necessary, such changes must be completed before using the GDDR Parameter Wizard.

GDDR HMC command requests are directed to HMC using the LPAR name and the processor name where the LPAR is defined. To enable HMC functions (IPL, CBU, and so forth), these variables must be defined for every managed system. Auto-Discovery populates this panel with a list of LPARs, which has at least one LPAR for each system defined on the Define Managed Systems panel (M,P,H,S), for the home site of that system.

1. If the panel is not pre-populated with existing entries, type **Add** in the command line at the top of the panel to display a new line for field entry.
2. Insert additional systems by entering **R** (Repeat) in the CMD field to the left of an existing entry and overtype the existing data.
3. Confirm or change the values that are displayed using the information provided with each field described below, subject to the precedence rules that apply to data returned to the GDDR Parameter Wizard by the Auto-Discovery feature.

   - Site

     Indicates the site location of the LPAR being defined. Site values are initially populated from the Define Managed Systems panel. Valid values are DC1, DC2, DC3, or DC4. For each system, you can add one LPAR per site in the configuration, except for systems protected with LPAR_RECOVERY as described in "Define system recovery attributes (M,P,H,R)" on page 325. LPAR_RECOVERY protected systems cannot have an LPAR defined at the site

where they have a Recovery LPAR defined. They cannot have an alternate LPAR defined, neither at the home site, nor at the site of the recovery LPAR location.

- System

  This field is populated from the System name supplied in the Define Managed Systems panel (M,P,H,S), one row per system, for the site which is the home site for that system.

  **Note:** If the field has not been pre-populated, this is the z/OS system name of a managed system which resides at the specified site. The name may be found on the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or the equivalent parameter file.

  Add rows for each additional site housing a CPC where a populated system can be IPL'd.

- Central Processor Complex (CPC)

  Specify the name of the central processor where the managed LPAR is defined. Auto-Discovery cannot determine the NETID portion of the CPC name, therefore you must provide this information on this panel.

- Logical Partition (LPAR)

  Specify the name of the LPAR within which this system runs at the specified site.

  You can find the CPC and LPAR names using the D M=CPU MVS console command, as shown in the following sample output:

```
RESPONSE=SYSB
IEE174I 20.59.34 DISPLAY M 781
PROCESSOR STATUS
ID  CPU                 SERIAL
0                       02F94E2096
1                       02F94E2096
2                       02F94E2096


 CPC ND = 002096.S07.IBM.83.00000008F94E
 CPC SI = 2096.N03.IBM.83.000000000008F94E
 CPC ID = 00
 CPC NAME = C
 LP NAME = ZOSESYSB LP ID = 2
 CSS ID = 0
 MIF ID = 2

   ONLINE - OFFLINE . DOES NOT EXIST W WLM-MANAGED
N NOT AVAILABLE

CPC ND CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC SI SYSTEM INFORMATION FROM STSI INSTRUCTION
CPC ID CENTRAL PROCESSING COMPLEX IDENTIFIER
CPC NAME CENTRAL PROCESSING COMPLEX NAME
LP NAME LOGICAL PARTITION NAME
```

- Bypass HMC

  Identifies the LPARs where hardware management console actions should be bypassed. When YES, this bypass affects all HMC actions for the specified LPARs. This includes Load, Reset, Activate, Deactivate, manual and automatic CBU Activate and Undo, Couple DS Realignment, and CF Structure Rebuild. Set this field to NO for GDDR-managed LPARs.

  If you set this field to * for any LPAR at site DC*m,* then on the next Save command, HMC Bypass will be in effect for ALL LPARs at site DC*m*. This is visible once you issue a LOAD after that Save.

4. If you made any changes, type **Save** on the command line and press **Enter**.
5. Press **F6** to go to the Define System Recovery Attributes panel (M,P,H,R).

## Define system recovery attributes (M,P,H,R)

When you specify option **R** in the Define Host Objects panel (M,P,H), the Define System Recovery Attributes panel is displayed:

```
------------------ GDDR - Define System Recovery Attributes - Row 1 to 8 of 8
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define Managed LPARs              <F6> Define Managed CPCs
 Type SAVE to save, LOAD to restart from last saved values.

                  Home  Contingency   Recovery-----------------
 System    Type   Site   System       Site    CPC (netid.nau)     LPAR
 --------   -     ---   --------      ---    -----------------  --------
 SYS1       C     DC1   N/A                  IBM390PS.Q3        SYS1
 SYS2       C     DC2   N/A                  IBM390PS.O         SYS2
 PRD1       P     DC1                  DC2    IBM390PS.PRD       LPARPRD1
 CNT1       P     DC2                  DC1    IBM390PS.CNT1      LPARCNT1
 SYS3       C     DC3   N/A                  IBM390PS.C         SYS3
 PRD3       P     DC3                  DC4    IBM390PS.CP3       LPARPRD3
 SYS4       C     DC4   N/A                  IBM390PS.F         SYS4
 CNT2       P     DC4                  DC3    IBM390PS.CNT       LPARCNT2
 ******************************* Bottom of data ********************************
```

**Figure 69** Define System Recovery Attributes panel (M,P,H,R)

1. If the panel is not pre-populated with existing entries, type **Add** in the command line at the top of the panel to display a new line for field entry.
2. Insert additional sites by entering **R** (Repeat) in the CMD field to the left of an existing entry and overtype the existing data.
3. Complete the fields as follows:

   - System

     The System name is populated from the System name supplied in the Define Managed Systems panel (M,P,H,S), one row per system.

   - Type

     C indicates C-system, P indicates production system.

   - Home Site

     The Site ID of the site where the system normally runs.

- Contingency System

  A system at a remote site on which applications can run if a primary managed system fails.

  The managed system and contingency system pairing is exclusive, meaning that the home LPAR location of a contingency system cannot also be a target LPAR for CPC/LPAR Recovery purposes.

  The field may be left blank, as contingency systems are not required.

The following Recovery Site, CPC and LPAR values must either all be specified or must all be omitted. They indicate the Site, CPC, and LPAR where the failing system should be IPL'd if it fails in its normal location. If another system is running in the recovery LPAR, it will be reset-cleared and the failing system will be IPL'd in its place. LPARs which are the home of a system with LPAR recovery defined may not be the target of recovery for a different system.

- Recovery Site

  The site at which the recovery CPC is located. If this is different from the Home site, the configuration must include AutoSwap.

- CPC (netid.nau)

  The name of the central processor where the LPAR is defined.

- LPAR

  The name of the logical partition (LPAR) that this system runs in at the specified site. The CPC and LPAR names can be found using the D M=CPU MVS console command, as shown in the sample output on page 324.

4. If you made any changes, type **Save** on the command line and press **Enter**.
5. Press **F6** to go to the Define Managed CPCs panel (M,P,H,P).

## Define managed CPCs (M,P,H,P)

When you specify option **P** in the Define Host Objects panel (M,P,H), the Define Managed CPCs panel is displayed:

```
-------------------------- GDDR - Define Managed CPCs -------- Row 1 to 6 of 6
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define System Recovery Attributes  <F6> Define IPL Parameters
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat
                                                 ----- AUTOCBU ------
                                                 (REAL, TEST or NONE)
 CMD   Site      CPC (netid.nau)       CBU   Planned   Unplanned
 ---    ---     -----------------      ---   -------   ---------
  _      DC1     IBM390PS.PRD           NO    TEST       NONE
  _      DC1     IBM390PS.TST1         YES    NONE       TEST
  _      DC2     IBM390PS.CNT1         YES    TEST       REAL
  _      DC2     IBM390PS.TST3          NO    NONE       NONE
  _      DC3     IBM390PS.CP3           NO    TEST       TEST
  _      DC3     IBM390PS.G3            NO    NONE       TEST
  _      DC4     IBM390PS.CNT          YES    TEST       REAL
  _      DC4     IBM390PS.CP4           NO    NONE       TEST
 ****************************** Bottom of data ********************************
```

**Figure 70** Define Managed CPCs panel (M,P,H,P)

Auto-Discovery prepopulates this panel with a list of unique CPC names resulting from:

- CPCs for C-system LPARs
- CPCs for Managed LPARs
- CPCs for LPAR_RECOVERY LPARs

The AUTOCBU Options allow script control of capacity backup activation by cpcname and site. The presence of this parameter invokes script automation to activate licensed processing capacity on specified central processing complexes (CPCs) at the recovery site specified by site in preparation for restart of workload.

Operator control of capacity backup activation is performed using the Perform GDDR Actions menu. The option to cancel capacity backup following an activation is only available from the Perform CBU Actions (A,CBU), using the Undo CBU option.

1. If the panel is not pre-populated with existing entries, type **Add** in the command line at the top of the panel to display a new line for field entry.
2. Insert additional CPCs by entering **R** (Repeat) in the CMD field to the left of an existing entry and overtype the existing data.
3. Confirm or change the values that are displayed using the information provided with each field described below, subject to the precedence rules that apply to data returned to the GDDR Parameter Wizard by the Auto-Discovery feature.

   - Site

     Indicates the ID of the site location being specified. Valid values are DC1, DC2, DC3, or DC4.

   - Central Processor Complex (CPC)

     Indicates the name of the central processor where the managed LPAR is defined. There are no limits on the number of CPC name entries per site.

   - Capacity Backup (CBU)

     Defines whether Capacity Backup Upgrade is enabled from the GDDR ISPF interface for the named CPC. Specify **YES** or **NO**. If YES is specified, CBU actions will be allowed using the GDDR ISPF interface.

   - AUTOCBU options for Planned and Unplanned scripts

     – Planned: Valid values are **REAL**, **TEST**, or **NONE**.

       When **REAL**, capacity backup licenses will be activated as a part of planned script processing.

       When **TEST**, capacity backup capacity will be simulated as part of planned script processing.

       When **NONE**, no capacity backup automation actions are performed. This is the default value.

     – Unplanned: Valid values are **REAL**, **TEST**, or **NONE**.

       When **REAL**, capacity backup licenses will be activated as a part of unplanned script processing.

When **TEST**, capacity backup capacity will be simulated as part of unplanned script processing.

When **NONE**, no capacity backup automation actions are performed. This is the default value.

4. If you made any changes, type **Save** on the command line and press **Enter**.
5. Press **F6** to go to the Define IPL Parameters panel (M,P,H,I).

## Define IPL parameters (M,P,H,I)

When you specify option **I** in the Define Host Objects panel (M,P,H), the Define IPL Parameters panel is displayed:

**Note:** The example illustrates an SRDF/Star-A configuration.

```
----------------------- GDDR - Define IPL Parameters --------- Row 1 to 3 of 3
Command ===>

Press <F3> to return to the Define Host Objects Menu
<F5> Define Managed CPCs              <F6> Define HMC Load Activation Profiles
Type SAVE to save, LOAD to restart from last saved values.
     When HMC Only=YES, only RES and ALT STD IPL Parameters are accepted
Line commands: A dd, D elete, R epeat

     DASD                        STD IPL      Gold BCV IPL    Test BCV IPL   HMC
CMD  Site   System   Typ    Parameters      Parameters      Parameters     Only
---  ----  --------  ---  --------------  --------------  --------------  ----
 _   DC3   LB01K148  RES  0953B,953BM100  0953B,952BB101  0953B,952BB101  NO
                     ALT  0953B,953BM100  0953B,952BB101  0953B,952BB101
                     DRT  _____,________

 _   DC3   LB09M3B   RES  0947D,0670MGM1  0947D,0670MGM1  0947D,0670MGM1  NO
                     ALT  0947D,0670MGM1  0947D,0670MGM1  0947D,0670MGM1
                     DRT  _____,________

 _   DC4   LB09M3B   RES  0947D,0670MGM1  0947D,0670MGM1  0947D,0670MGM1  NO
                     ALT  0947D,0670MGM1  0947D,0670MGM1  0947D,0670MGM1
                     DRT  _____,________
```

**Figure 71** Define IPL Parameters panel (M,P,H,I)

The Define IPL parameters panel specifies the IPL parameters that GDDR may use to IPL a managed system at the primary DASD site. All IPL parameters are optional.

The following statements are recommendations for production systems where GDDR is expected to perform HMC LPAR LOAD actions for the affected systems.

- Add an STD IPL parameter for each site where the listed system has an LPAR defined.
- Add an STD IPL parameter for the other site in the same region as the listed DASD site.
- Add a Gold BCV IPL and/or a Test IPL BCV parameter for systems for which GDDR is expected to perform HMC LPAR LOAD actions during GDDR Test IPL from BCV scripts.
  - For the IPL parameter-types listed previously, add ALT IPL parameters if you want GDDR to perform HMC LPAR LOAD actions using an alternate SYSRES volume.

- Add DRT IPL parameters for systems for which you want GDDR to perform HMC LPAR LOAD actions during DR test scripts and if you want to have a SYSRES volume for DR-test circumstances which is different from the normal or alternate SYSRES volumes.
- Only RES STD IPL and ALT STD IPL parameters are allowed for HMC-only systems.
- The Define IPL Parameters panel (M,P,H,I) also allows you to specify ALT RES IPL parameters for C-systems. The only use case in GDDR for these is during system recovery for a C-system.

To complete the panel:

1. If the panel is not prepopulated with existing entries, type **Add** in the command line at the top of the panel to display a new line for field entry.
2. Insert additional sites by entering **R** (Repeat) in the CMD field to the left of an existing entry and overtype the existing data.
3. Confirm or change the values that are displayed using the information provided with each field described below, subject to the precedence rules that apply to data returned to the GDDR Parameter Wizard by the Auto-Discovery feature.

   - DASD site

     Indicates the site location of the IPL parameters being defined. Valid values are DC1, DC2, DC3, or DC4. This is the site location of the DASD devices reflected in the LOAD address portion of the IPL parameters. During GDDR scripts, GDDR will select STD or BCV IPL parameters based on the type of script and on which site is current primary DASD site.

   - System

     This field is populated from the System name supplied in the Define Managed Systems (M,P,H,S) panel, one row per system, for the site which is the home site for that system.

     ---

     **Note:** If the field has not been pre-populated, this is the z/OS system name of a managed system or a C-system which resides at the specified site. The name may be found on the SYSNAME=*system-name* statement in SYS1.PARMLIB(IEASYS00) or the equivalent parameter file.

     ---

     Add rows for each additional site housing a CPC where a populated system can be IPL'd.

   - TYP

     Valid values are:

     – ALT: Alternate STD IPL parameter
     – DRT: DR Test RES IPL parameter
     – RES: RES STD IPL parameter

     ---

     **Note:** GDDR IPL capabilities using BCVs include support for alternate SYSRES volumes on BCV devices.

     ---

- STD IPL parameters
- Gold BCV IPL parameters
- Test BCV IPL parameters

  In the format: *ssss*,*iiiixxmn*

  Where:

  – *ssss* is the  device address.

  – *iiii* is the IODF device address.

  – *xx* is the LOAD*xx* member suffix.

  – *m* is the IMSI Field.

  – *n* is the IEANUC0*n* suffix.
- HMC Only

  This read-only field indicates if the specified system is external to the population of systems managed by GDDR in scripts.

  An external system must be defined as an HMC-only system in the Define Managed Systems panel (M,P,H,S). HMC-only systems may be managed using the Perform HMC LPAR Actions panel (A,L). GDDR parameter validation is bypassed for the systems defined as HMC-only systems.

  In the Define IPL Parameters panel (M,P,H,I), the External systems may only specify the RES and ALT STD IPL parameters. All other data entry columns for an external system (BCV and DRT) are set to N/A.

You can locate IPL parameters using the D IPLINFO MVS console command, as shown in the following sample output:

```
RESPONSE=SYSB
IEE254I 21.14.59 IPLINFO DISPLAY 860
SYSTEM IPLED AT 15.59.20 ON 02/02/2009
RELEASE z/OS 01.10.00 LICENSE = z/OSe
USED LOAD02 IN SYS1.IPLPARM ON 0A26
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = B0
IEASYS LIST = B0 (OP)
IODF DEVICE 0A26
IPL DEVICE 0A1D VOLUME RESB14
```

4. If you made any changes, type **Save** on the command line and press **Enter**.
5. Press **F6** to go to the Define HMC Load Activation Profiles panel (M,P,H,A).

## Define HMC load activation profiles (M,P,H,A)

When you specify option **A** in the Define Host Objects panel (M,P,H), the Define HMC Load Activate Profiles panel is displayed:

```
------------------ GDDR - Define HMC Load Activation Profiles -- Row 1 to 13 of 13
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define IPL Parameters              <F6> Define Managed Couple DSNs (1/2)
 Type SAVE to save, LOAD to restart from last saved values.

 Enter the required information. Press <Enter> to Validate.

                                                                        Load
        Site  System   C/P  Load type    HMC Load Profile     Parameters
        ---- --------  ---  ---------    ----------------    -------------
        DC1  PRD1      P    STANDARD     PRD1STDPROF1        70FD,70FDQ3M1
        DC1  TST1      P    STANDARD     ________________    70FD,70FDQ3M1
        DC1  SYS1      C    STANDARD     SYS1STDPROF1        72FD,72FDQ3M1
        DC1  TST2      P    STANDARD     ________________    71FD,71FDQ3M1
        DC2  CNT1      P    STANDARD     CNT1STDPROF1        B1FD,B0FDQ3M1
        DC2  TST3      P    STANDARD     ________________    B1FD,B1FDQ3M1
        DC2  SYS2      C    STANDARD     SYS2STDPROF1        B2FD,B2FDQ3M1
        DC3  PRD3      P    STANDARD     PRD3STDPROF1        C1FD,C1FDQ3M1
        DC3  TST4      P    STANDARD     ________________    C1FD,C1FDQ3M1
        DC3  SYS3      C    STANDARD     SYS3STDPROF3        C1FD,C1FDQ3M1
        DC4  CNT2      P    STANDARD     CNT2STDPROF1        D1FD,D1FDQ3M1
        DC4  TSTA      P    STANDARD     ________________    D0FD,D0FDQ3M1
        DC4  SYS4      C    STANDARD     SYS4STDPROF1        D4FD,D4FDQ3M1
 ******************************** Bottom of data ********************************
```

**Figure 72** Define HMC Load Activation Profiles panel (M,P,H,A)

**Note:** The Define HMC Load Activation Profiles panel is not applicable in sites where cross-site host-DASD channels are not available.

HMC load activation profiles are optional. The GDDRMAIN started task is required to be present on systems for which this feature is desired.

The same load profile name cannot be used by more than one system per site.

View, define, or modify the panel fields as follows:

6. To “delete” a row, blank out the profile name and SAVE (will not write a line to the panel's member in the parameter work dataset for that row).
7. Optionally complete the fields as follows:
   - Site

     Populated from the Define C-Systems panel (M,P,C,C).
   - System

     Populated from the Define Managed Systems panel (M,P,H,S).
   - C/P

     Indicates a C-system (C) or a managed (P) system. Populated from the Define C-Systems panel (M,P,C,C) and Define Managed Systems panel (M,P,H,S).
   - Load type

     Populated from the Define IPL Parameters panel (M,P,H,I).

- HMC Load Profile

  A 1-16 character user-specified name of the load activation profile where the IPL residence volume and load parameter for the managed system will be stored.

- Load Parameters

  Populated from the Define IPL Parameters panel (M,P,H,I).

8. If you made any changes, type **Save** on the command line and press **Enter**.

**Validate and update HMC load activation profiles**

Run the GDDR Load Profile Management utility (GDDRLPRF) described in "GDDR Load Profile Management utility (GDDRLPRF)" on page 508 to validate and update the HMC load activation profiles.

## Define managed coupled datasets (M,P,H,D)

When you specify option **D** in the Define Host Objects panel (M,P,H), the Define Managed Couple Datasets (1/2) panel is displayed:

```
----------------- GDDR - Define Managed Couple Datasets (1/2)  Row 1 to 1 of 1
 Command ===>

Press <F3> to return to the Define Host Objects Menu
<F5> Define HMC Load Activation Profile <F6> Define Managed Couple DSNs (2/2)
Type SAVE to save, LOAD to restart from last saved values.

Indicate (Yes/No) for which types you want GDDR Couple Dataset management.

     Sysplex      ARM   BPX   CFR  LOGR   SFM   SYS   WLM
     --------     ---   ---   ---   ---   ---   ---   ---
     PRODPLX      YES   YES   YES   NO    YES   YES   NO
 ****************************** Bottom of data ******************************
```

**Figure 73** Define Managed Couple Datasets (1/2) panel (M,P,H,D)

The use of coupled datasets is dependent on the site's exploitation of sysplex architecture. If no managed systems are part of a sysplex, the following message is displayed upon opening this panel:

```
GDDI371W No sysplexes defined, please either assign systems to a
   sysplex or proceed to the next panel.
```

The fields in the Define Managed Couple Datasets (1/2) panel indicate the types of coupled datasets that are used on a specified sysplex.

1. View, define, or modify the panel fields as follows:

   - Sysplex

     Indicates the sysplex name, if applicable. Sysplex values are populated from the Define Managed Systems panel (M,P,H,S).

   - Coupled dataset types: ARM, BPX, CFR, LOGR, SFM, SYS, WLM

     Type **YES** or **NO** under each dataset type to indicate whether or not you want GDDR to ensure correct coupled dataset placement for that type during site swap scripts for a specified sysplex.

Coupling facility dataset and structure management is further controlled for planned and unplanned scripts by the Realign Couple Datasets, Rebuild CF Structures, and the CF Rebuild Timeout fields described in "Define script sysplex options (M,P,O,S)" on page 341.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to complete the coupled dataset definitions as shown in the Define Managed Couple Datasets (2/2) panel:

**Note:** The example illustrates an SRDF/Star configuration.

```
----------------- GDDR - Define Managed Couple Datasets (2/2) ------------------
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define Managed Couple DSNs (1/2)     <F6> Define Managed CF Structures
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat

 CMD  Sysplex    Type  Site  Role  Couple Dataset Name
 ---  --------   ----  ---   -     ------------------------------------------
  _   PRODPLX    ARM   DC1   P     PRODPLX.ARM.CDS01
  _   PRODPLX    ARM   DC2   P     PRODPLX.ARM.CDS02
  _   PRODPLX    ARM   DC1   A     PRODPLX.ARM.CDS04
  _   PRODPLX    ARM   DC2   A     PRODPLX.ARM.CDS03
  _   PRODPLX    BPX   DC1   P     PRODPLX.BPX.CDS01
  _   PRODPLX    BPX   DC2   P     PRODPLX.BPX.CDS02
  _   PRODPLX    BPX   DC1   A     PRODPLX.BPX.CDS04
  _   PRODPLX    BPX   DC2   A     PRODPLX.BPX.CDS03
  _   PRODPLX    CFR   DC1   P     PRODPLX.CFR.CDS01
  _   PRODPLX    CFR   DC2   P     PRODPLX.CFR.CDS02
  _   PRODPLX    CFR   DC1   A     PRODPLX.CFR.CDS04
  _   PRODPLX    CFR   DC2   A     PRODPLX.CFR.CDS03
  _   PRODPLX    SFM   DC1   P     PRODPLX.SFM.CDS01
  _   PRODPLX    SFM   DC2   P     PRODPLX.SFM.CDS02
  _   PRODPLX    SFM   DC1   A     PRODPLX.SFM.CDS04
  _   PRODPLX    SFM   DC2   A     PRODPLX.SFM.CDS03
  _   PRODPLX    SYS   DC1   P     PRODPLX.SYS.CDS01
  _   PRODPLX    SYS   DC2   P     PRODPLX.SYS.CDS02
  _   PRODPLX    SYS   DC1   A     PRODPLX.SYS.CDS04
  _   PRODPLX    SYS   DC2   A     PRODPLX.SYS.CDS03
 ****************************** Bottom of data ******************************
```

**Figure 74** Define Managed Couple Datasets (2/2) panel (M,P,H,D)

The fields in the Define Managed Couple Datasets (2/2) panel indicate the names of primary and alternate coupled datasets of the specified type to be used on the systems belonging to the specified sysplex.

**Note:** Each coupled dataset must be cataloged on all systems in the sysplex.

There must be a primary and an alternate coupled dataset each sysplex, for each of the possible primary DASD sites, and for each coupled dataset type being used.

GDDR ensures that the primary coupled datasets are located wherever the primary DASD currently resides. If the secondary DASD site is available, GDDR ensures that an alternate coupled dataset is used on the secondary DASD site. If the secondary DASD site is unavailable, and valid dataset names have been provided for DSN3 and DSN4 as described in Table 32 on page 334, GDDR ensures that an alternate coupled dataset is used at the primary DASD site.

Four lines display for each sysplex and managed type, one line for each dataset role (P or A) for both sites DC1 and DC2. For each type and role within a sysplex, specify the dataset names that are cataloged at DC1 and DC2 as shown in Table 32.

**Table 32** Defining managed coupled datasets

| Sysplex | TYPE | SITE | ROLE | Coupled dataset name |
|---|---|---|---|---|
| TESTPLX1 | ARM | DC1 | P | DSN1: used as primary when DC1 is primary and used as alternate when DC1 is secondary |
| | | | A | DSN3: used as alternate when DC1 is primary and DC2 is unavailable |
| | | DC2 | P | DSN2: used as primary when DC2 is primary and used as alternate when DC2 is secondary |
| | | | A | DSN4: used as alternate when DC2 is primary and DC1 is unavailable |

1. View, define, or modify the fields as follows:
   - Sysplex

     Indicates the sysplex name. Sysplex values are populated with the list of sysplexes which have at least one managed coupled dataset type, as specified in the Define Managed Couple Datasets (1/2) panel (M,P,H,D).
   - Type

     Indicates the coupled dataset type as specified in the Define Managed Couple Datasets (1/2) panel (M,P,H,D). Valid values are ARM, BPX, CFR, LOGR, SFM, SYS, or WLM.
   - Site

     Indicates the ID of the site location.
   - Role

     Indicates the role of the dataset. Valid values are P for primary or A for alternate.
   - Couple Dataset Name

     Specify the dataset names that are cataloged at DC1 and DC2.
2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Define Managed CF Structures panel (M,P,H,CF).

## Define managed CF structures (M,P,H,CF)

When you specify option **CF** in the Define Host Objects panel (M,P,H), the Define Managed CF Structures panel is displayed:

```
---------------------- GDDR - Define Managed CF Structures -- Row 1 to 16 of 28
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define Managed Couple DSNs (2/2)     <F6> Define External Workloads
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat

 CMD  Sysplex     Structure Name   CF-Site   Eligible Coupling Facility Names
 ---  --------   ----------------    ---    ----------------------------------
  _    PROD       DSNDB2I_GBP0        DC1    C15
  _    PROD       DSNDB2I_GBP0        DC2    O15
  _    PROD       DSNDB2I_GBP14       DC1    C15
  _    PROD       DSNDB2I_GBP14       DC2    O15
  _    PROD       DSNDB2I_GBP32K      DC1    C15
  _    PROD       DSNDB2I_GBP32K      DC2    O15
  _    PROD       DSNDB2I_LOCK1       DC1    C15
  _    PROD       DSNDB2I_LOCK1       DC2    O15
  _    PROD       DSNDB2I_SCA         DC1    C15
  _    PROD       DSNDB2I_SCA         DC2    O15
  _    PROD       EMCP_RRS_STR_1      DC1    C15,O15
  _    PROD       EMCP_RRS_STR_1      DC2    O15,C15
  _    PROD       EMCP_RRS_STR_2      DC1    C15,O15
  _    PROD       EMCP_RRS_STR_2      DC2    O15,C15
  _    PROD       IEFAUTOS            DC1    C15
  _    PROD       IEFAUTOS            DC2    O15
```

**Figure 75** Define Managed CF Structures panel (M,P,H,CF)

**Note:** If the managed systems are not members of a sysplex, the following message is displayed in the panel:

```
GDDI371W No sysplexes defined, please either assign systems to a
sysplex or proceed to the next panel.
```

During swap scripts, GDDR will issue Display XCF commands for the CF Structures identified in the Define Managed CF Structures panel and determine if they are currently located in the preferred coupling facility. If a structure is found in a different coupling facility than the one which is first in the list for the current primary DASD site, GDDR will issue REBUILD commands and verify the results. Verification is repeated until all structures are located in one of the acceptable coupling facilities.

Coupling facility structure management is controlled for planned and unplanned scripts by the Rebuild CF Structure and the CF Rebuild Timeout fields described in "Define script sysplex options (M,P,O,S)" on page 341.

1. View, define, or modify the panel fields as follows:

   - Sysplex

     Indicates the sysplex name, if applicable. Sysplex values are populated from the Define Managed Systems panel.

     If the panel is not pre-populated with existing Sysplex name entries, type **Add** in the command line at the top of the panel to display a new line for an entry. Insert additional Sysplex names by entering **R** in the CMD field to the left of an existing Sysplex entry.

- Structure Name

  Provide a coupling facility structure name or names for each specified sysplex using the following guidelines:

  - Length: 1-16 characters
  - Position 1: Uppercase alphabetic
  - Positions 2-16: Uppercase alphabetic, numeric, or _, @, $, #

- CF Site

  The ID of the site location being specified. Site values are populated from the Define Managed Systems panel. This is the site which is the primary DASD site when the listed sequence of eligible coupling facility names should be used to determine correct location of CF Structures.

- Eligible Coupling Facility Names

  Provide up to 4 coupling facility names, delimited by commas, using the following guidelines:

  - Length: 1-8 characters
  - Position 1: Uppercase alphabetic
  - Position 2-8: Uppercase alphabetic, numeric, or _, @, $, #

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Define External Workloads panel (M,P,H,W).

## Define external workloads (M,P,H,W)

External workloads run in mainframe systems which do not have their DASD in the managed storage systems. GDDR can coordinate stop and start of the workload on these "non-managed" mainframe systems with the workload stop and start for managed systems. When GDDR takes actions that impact the managed systems, it has the capability to trigger the stop or start of external workloads through a user exit.

---

**Note:** The definition of external workloads is optional. If no external workloads are to be stopped or started in sequence with managed systems, press **F6** to go to the Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E).

---

When you specify option **W** in the Define Host Objects panel (M,P,H), the Define External Workloads panel is displayed:

```
---------------------- GDDR - Define External Workloads ----- Row 1 to 2 of 2
 Command ===>

 Press <F3> to return to the Define Host Objects Menu
 <F5> Define Managed CF Structures       <F6> Define Dell EMC MF STCs
 Type SAVE to save, LOAD to restart from last saved values.

 Line commands: A dd, D elete, R epeat

 CMD  Site    System
 ---  ---    --------
  _   DC1    SYS8
  _   DC2    SYSA
 ****************************** Bottom of data ********************************
```

**Figure 76** Define External Workloads panel (M,P,H,W)

1. Type **Add** in the command line at the top of the panel to display a new line for an entry.
2. View, define, or modify the panel fields as follows:
   - Site

     Indicates the ID of the specified site location.
   - System

     Indicates the system names which host external workload requiring coordination with system actions performed by GDDR.

     If you are using the user exits provided with GDDR, then communication to the targeted systems requires message interception rules GDDX191I and GDDX291I to be enabled. Use the following GDDRMAIN console commands to enable these rules:

     ```
     F GDDRMAIN,MSGS,GDDX191I,ENABLE
     F GDDRMAIN,MSGS,GDDX291I,ENABLE
     ```
3. If you made any changes, type **Save** on the command line and press **Enter**.
4. Press **F6** to go to the Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E).

## Define Mainframe Enablers STCs (M,P,H,E)

GDDR uses information about the site installation parameters to communicate with Mainframe Enablers started tasks that reside on managed systems and C-systems.

When you specify option **E** in the Define Host Objects panel (M,P,H,E), the Define Dell EMC Mainframe Enabler STCs panel is displayed:

```
---------- GDDR - Define Dell EMC Mainframe Enablers STCs  Row 1 to 713 of 1013
Command ===>

Press <F3> to return to the Define Host Objects Menu
<F5> Define External Workloads          <F6> Specify GDDR Options
Type <SAVE> to save, <LOAD> to restart from last saved values.

Line commands: A dd, D elete, R epeat

                  STC         SUB
CMD   System  Type   Name     MSTR       Parameter Dataset and Member(s)
---  -------- ---- --------  ---- ------------------------------------------
 _   SYS4     CG   GDDRCGRP  YES  X
                                   CGPCFG09
 _   SYS4     RDF  N/A       NO   EMC.GDDR520.CUSTOM.PARMLIB
                                   SRDFSTAR
 _   CNT1     SCF  GDDRSCF   NO   N/A
                                   N/A
 _   PRD1     SCF  GDDRSCF   NO   N/A
                                   N/A
 _   SYS4     SCF  GDDRSCF   NO   N/A
                                   N/A
****************************** Bottom of data ******************************
```

**Figure 77** Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E)

1. Type **Add** in the command line at the top of the panel to display a new line for an entry.
2. Review or edit field values as appropriate. Ensure the values meet the requirements listed in "Mainframe Enablers STC requirements" on page 338.

- System

  Indicates the z/OS system name of a managed system or a C-system which resides at the specified site. System Name values are prepopulated from the Define Managed Systems panel.

- STC Type

  Indicates the Mainframe Enablers application. This field is prepopulated. Valid values are:

  – SCF: Symmetrix Control Facility (ResourcePak Base)

  – RDF: SRDF Host Component

  – CG: Consistency Groups

- STC Name

  Indicates the STC name.

- SUB MSTR

  Indicates whether the started task runs under the master subsystem (SUB=MSRT) or not:

  – YES: SUB=MSRT is specified

  – NO: SUB=MSRT is not specified

- Parameter Dataset and Member(s)

  Provides the name of the partitioned dataset and members containing the customization parameters for the specified started task (described in "Install GDDR started procedures" on page 141). This field is prepopulated.

3. If you made any changes, type **Save** on the command line and press **Enter**.
4. Press **F6** to go to the Specify GDDR Options panel (M,P,O).

### Mainframe Enablers STC requirements

Ensure that the values specified in the Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E) meet the requirements listed in Table 33 on page 338 for managed systems and Table 34 on page 339 for C-systems.

**Table 33** Mainframe Enablers STC requirements for managed systems

| STC type | STC name | SUB= MSTR | Parameter Dataset | Parameter Member |
|---|---|---|---|---|
| CG | The STC name.<br>If you do not provide a ConGroup STC name, GDDR uses a default value of GDDRCGRP and does not specify a MBR= keyword on ConGroup start commands. If this default handling is not acceptable, specify the ConGroup STC name and, if necessary, a member name for ConGroup parameters. | YES | X<br>(dummy name, required for syntax) | Optional.<br>Provide a member name if the ConGroup STC JCL does not have a hardcoded member name. |
| SCF | The STC name. | YES | Not applicable | Not applicable |

**Table 34** Mainframe Enablers STC requirements for C-systems

| STC type | STC name | SUB= MSTR | Parameter Dataset | Parameter Member |
|---|---|---|---|---|
| CG | The STC name. | NO | X<br>(dummy name, required for syntax) | Optional.<br>Provide a member name if the ConGroup STC JCL does not have a hardcoded member name. |
| RDF | The STC name. | NO[a] | The parameter dataset pointed to by the RDFPARM DD statement of the SRDF Host Component STC. This is a required value. | The parameter member pointed to by the RDFPARM DD statement of the SRDF Host Component STC. This is a required value. |
| SCF | The STC name. | NO | Not applicable | Not applicable |

a. SUB=MSTR is ignored for RDF entries.

## Specify GDDR options (M,P,O)

When you specify option **O** in the Parameter Management Options Menu panel (M,P), the Specify GDDR Options panel is displayed:

```
------------------------- GDDR - Specify GDDR Options -------------------------
Option ===>

     O   Default Script Call Overrides                    This System: SYS
     S   Script Sysplex Options                             This Site: DC2
     J   Script JCL Parameters                            This region: 1
     U   Utility Parameters                                  Master-C: SYS
     T   Tuning Values                                    Primary Site: DC1
     L   GDDR user labels                                 Primary DASD: DC1
                                                            Automation: ON
                                                       Planned script: None
                                                     Unplanned script: None




 Select an option and press <Enter>


<F5> Define Dell EMC MF Enabler STCs    <F6> Default Script Call Overrides
Press <F3> to return to the Parameter Management Options Menu
```

**Figure 78** Specify GDDR Options panel (M,P,O)

Complete the following tasks to define your site-specific script options:

- Specify default script call overrides (M,P,O,O)
- Define script sysplex options (M,P,O,S)
- Specify script JCL parameters (M,P,O,J)
- Specify utility parameters (M,P,O,U)
- Specify tuning values (M,P,O,T)
- Define GDDR user labels (M,P,O,L)

## Specify default script call overrides (M,P,O,O)

When you specify option **O** in the Specify GDDR Options panel (M,P,O), the Specify Default Script Call Overrides panel is displayed:

```
--------------- GDDR - Specify Default Script Call Overrides ------------------
Command ===>

Press <F3> to return to the Specify GDDR Options Menu
<F5> Specify GDDR Options              <F6> Script Sysplex Options
Type <SAVE> to save, <LOAD> to restart from last saved values.

Enter Y to call the function, N to NOT call the function.

 Call   Program   Function Description
 ----   --------  ------------------------------------------------------------
                                                                    More:     +
   Y    GDDRRDF0  Call DYNAPI Interface
   Y    GDDRRDF0  DYNAPI - SRDF/S Commands
   Y    GDDRRDF0  DYNAPI - DeletePair and Half-DeletePair Commands
   Y    GDDRRDF0  DYNAPI - CreatePair and ResumePair Commands
   Y    GDDRRDF0  DYNAPI - SRDF/A Commands
   Y    GDDRRDF0  DYNAPI - Swap and Half-Swap commands

   N    GDDRXDRV  Manage Distributed Workload
   N    GDDRKF0C  Trigger Production Workload Stop
   N    GDDRKF0I  Trigger Production Workload Startup

   N    GDDRGF08  Use ConGroup Shutdown/Startup instead of Refresh

   Y    GDDRKF0E  Transfer AutoSwap Ownership
   Y    GDDRGFFM  Transfer Master Function Ownership

   N    GDDRKF20  Simulate BCV Actions
   Y    GDDRKF20  Manage BCVs at DC1
   Y    GDDRKF20  Manage BCVs at DC2
   Y    GDDRKF20  Manage BCVs at DC3
   Y    GDDRKF20  Manage BCVs at DC4

   N    GDDRKF20  Manage External BCV devices

   N    GDDRKF20  Manage TEST BCVs at DC1
   N    GDDRKF20  Manage TEST BCVs at DC2
   N    GDDRKF20  Manage TEST BCVs at DC3
   N    GDDRKF20  Manage TEST BCVs at DC4

   N    GDDRKF20  Use BCVGROUP data set

   N    GDDRCL00  Perform SDDF Session Cleanup

   N    GDDRXACT  Perform SQAR Single Region Restart
   N    GDDRXACT  Perform SQAR Single Side Restart

   N    GDDRRDF0  Manage External SRDF devices

   Y    GDDRZDP0  Perform zDP PAUSE and RELDLOCK Actions
   Y    GDDRZDP0  Perform zDP RESUME and START Actions

   Y    GDDRZDP0  Perform zDP Unplanned RESUME at SITEA
   Y    GDDRZDP0  Perform zDP Unplanned RESUME at SITEB
   Y    GDDRZDP0  Perform zDP Unplanned RESUME at SITEC
   Y    GDDRZDP0  Perform zDP Unplanned RESUME at SITED

   Y    GDDRGFHS  Allow Primary Site HMC Actions in Planned Scripts
   Y    GDDRGFHS  Allow Secondary Site HMC Actions in Planned Scripts

   Y    GDDRGFHS  Allow Primary Site HMC Actions in Unplanned Scripts
   Y    GDDRGFHS  Allow Secondary Site HMC Actions in Unplanned Scripts

   Y    GDDRGFHS  Allow Primary Site HMC Actions in Test Scripts
   Y    GDDRGFHS  Allow Secondary Site HMC Actions in Test Scripts

   Y    GDDRGFHS  Allow Primary Site HMC Actions in RDR Scripts
   Y    GDDRGFHS  Allow Secondary Site HMC Actions in RDR Scripts

   Y    GDDRGFHS  Allow LOAD CLEAR when Activating LPARs in Scripts

   Y    GDDRGFHS  Confirm GDDR Script HMC actions by WTOR

   N    GDDRHMC2  Prompt for Alternate SYSRES usage
   N    GDDRHMC2  Use DR-Test IPL parameters
   N    GDDRKF20  BACKGROUNDCOPY for TF/Clone
****************************** Bottom of Data ******************************
```

**Figure 79** Specify Default Script Call Overrides panel (M,P,O,O)

1. Overtype the default **Y** or **N** call override values to change the default overrides used by all scripts.

   Table 5, “GDDR call overrides,” on page 85 lists call overrides and their meanings.

- You can set the 'Confirm GDDR Script HMC actions by WTOR' call override setting to **P** for Priority, meaning that WTORs for HMC actions during scripts will be reduced to 1 per priority level. The usual confirmation WTORs for each individual action will then be suppressed.
- GDDR BCV management is controlled using call overrides to program GDDRKF20. If you set call overrides to allow GDDR BCV management for a given site, it is assumed that BCVs are in the proper state prior to the script being started; therefore, you should not change BCV states in between scripts using non-GDDR actions.

  Ensure that call overrides for BCV management at a given site are set to N, if no BCVs are defined to GDDR for that site.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Script Sysplex Options panel (M,P,O,S).

## Define script sysplex options (M,P,O,S)

When you specify option **S** in the Specify GDDR Options panel (M,P,O), the Script Sysplex Options panel is displayed:

```
---------------------- GDDR - Script Sysplex Options --------------------------
Command ===>

Press <F3> to return to the Specify GDDR Options Menu
<F5> Default Script Call Overrides      <F6> Script JCL Parameters
Type <SAVE> to save, <LOAD> to restart from last saved values.

Please indicate (Yes/No) GDDR Script Sysplex management options by Site.

               Realign              Rebuild
            Couple Datasets       CF Structures       CF Rebuild
   Site     Planned Unplanned    Planned Unplanned       Timeout
   ----     ------- ---------    ------- ---------       -------
    DC1        NO       NO          NO       NO            600
    DC2       N/A      N/A          N/A      N/A           N/A
    DC3        NO       NO          NO       NO            600
    DC4        NO       NO          NO       NO            600
****************************** Bottom of Data ******************************
```

**Figure 80** Script Sysplex Options panel (M,P,O,S)

1. Specify **YES** or **NO** for the following script management options for each site:
   - Realign Couple Datasets Planned and Unplanned

     Specify whether planned or unplanned scripts should manage coupled datasets when the primary site is the one specified in the Site field.

     – When Planned is YES: Automate the management of coupled datasets as a part of planned script processing.

     – When Unplanned is YES: Automate the management of coupled datasets as a part of unplanned script processing.

     Further control is provided by the coupled datasets defined in the Define Managed Couple Datasets panel (M,P,H,D) and the External field in the Define IPL Parameters panel (M,P,H,I).

- Rebuild CF Structures Planned and Unplanned

  Specify whether planned or unplanned scripts should manage coupling facility (CF) structures at the site indicated in the Site field.

  - When Planned is YES: Automate the management of CF structures as a part of planned script processing.
  - When Unplanned is YES: Automate the management of CF structures as a part of unplanned script processing.

  Further control is provided by the PSTR.* parameters, and the System Exclude list.

- CF Rebuild Timeout

  Specify the number of seconds to allow for the processing required to rebuild CF structures. Valid values are 1-3600. The default value is 600.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Script JCL Parameters panel (M,P,O,J).

**System qualification for sysplex options**

The following qualifications apply both to coupled dataset management and CF structure management in scripts, as well as standalone actions in the Manage Couple Datasets panel (A,S) and the Manage CF Structures panel (A,CF).

- z/OS XCF commands are issued for a list of eligible systems constructed from primary and secondary sites.
- Exclude systems from the Exclude list.
- Exclude systems with Desired State = D.
- Exclude systems where GDDRMAIN is down.
- Limit resulting list of systems to 1 system per sysplex.
- The eligible system is preferably located at the site where the script runs.

Usage of GDDR coupled dataset and CF structure management require that systems are defined in GDDRPARM COMM and CPC parameters, and have GDDRMAIN running at the time of the action.

Systems not defined in the GDDRPARM file or where GDDRMAIN is found inactive are not considered eligible for these features, and sysplexes where no systems meet the criteria will be skipped. GDDR issues a warning message when no systems are retained.

The reduction of the list of eligible systems to 1 per sysplex requires correct sysplex names to be provided in the Define Managed Systems panel (M,P,H,S).

Possible values of the CF_STRUCTURE.REBUILD.*site* and COUPLE_DS.REALIGN.*site* global variables are ALL, Planned, Unplanned or None, using the script management options available in the Script Sysplex Options panel (M,P,O,S).

**Table 35** Script management options

| Global Variable values | Rebuild Couple Datasets | | Rebuild CF Structures | |
|---|---|---|---|---|
| | **Planned** | **Unplanned** | **Planned** | **Unplanned** |
| ALL | Yes | Yes | Yes | Yes |
| Planned | Yes | No | Yes | No |
| Unplanned | No | Yes | No | Yes |
| None | No | No | No | No |

If the value of COUPLE_DS.REALIGN.*site* is Unplanned:

- The primary coupled datasets will stay at their original site during planned scripts.
- Unplanned swap scripts will align the primary coupled datasets to the recovery site.

Script GDDRPA0A (used to resume replication) is generally considered a planned script. However, since this script runs after the unplanned scripts which have realigned the primary coupled datasets to the recovery site, GDDRPA0A must restore the original situation.

By default, GDDR assumes that the primary coupled datasets need to be at DC1 if the current primary region is region 1, and at DC3 if the current primary region is region 2. If this is unacceptable, use the Manage GDDR System Variables panel (M,S) to update the USEROPT.CPLPRIM.1 variable, giving it a value of DC2, and the USEROPT.CPLPRIM.2 variable, giving it a value of DC4. To make the changes permanent, start a new GDDR Parameter Wizard session using a backup of your current GDDR global variables, and do an activate.

Script GDDRPA0A then realigns the alternate coupled datasets to the opposite site in the same region.

## Specify script JCL parameters (M,P,O,J)

When you specify option **J** in the Specify GDDR Options panel (M,P,O), the Script JCL Parameters panel is displayed:

```
----------------------- GDDR - Script JCL Parameters ------------------------
Command ===>

Press <F3> to return to the Specify GDDR Options Menu
<F5> Script Sysplex Options             <F6> Utility Parameters
Type <SAVE> to save, <LOAD> to restart from last saved values.

Enter the required information. Use <Up> or <Down> to scroll through C-Systems.

Site: DC1   C-System: LB01M3A           HC-Prefix: M5
  Work Unit: SYSALLDA              Jobname Prefix: GDD        Enforce: NO
  Work HLQ:  EMC.GDDR520.TEMP    Surrogate User ID: _______   Enforce: NO
Enter jobcards:
  //GDDRPACT JOB (0),'P',MSGLEVEL=(1,1),CLASS=A,MSGCLASS=X,SYSAFF=*,
  // NOTIFY=&SYSUID
  //*
  ____________________________________________________________________________
Site: DC2   C-System: LB01M34           HC-Prefix: M5
  Work Unit: SYSALLDA              Jobname Prefix: GDD        Enforce: NO
  Work HLQ:  EMC.GDDR520.TEMP    Surrogate User ID: _______   Enforce: NO
Enter jobcards:
  //GDDRPACT JOB (0),'P',MSGLEVEL=(1,1),CLASS=A,MSGCLASS=X,SYSAFF=*,
  // NOTIFY=&SYSUID
  //*
  ____________________________________________________________________________
```

**Figure 81** Script JCL Parameters panel (M,P,O,J)

GDDR automation sequences run as z/OS batch jobs. The Script JCL Parameters panel (M,P,O,J) provides the customization required for those batch jobs to run successfully in your z/OS environment. For GDDR scripts started internally by GDDR, this panel is the only place where this information can be provided. The job card information can be overridden as part of the script submission started from the Select Script to Run panel (S) as described in "Run GDDR scripts (S)" on page 375.

1. View, define, or modify the panel fields as follows:

   - Site

     Indicates the site. This field is read-only.

   - C-System

     Indicates the C-system. This field is read-only.

     Script JCL values are maintained for each C-system. Specify the values for the first displayed C-system, and scroll to the next C-system using **F8**.

   - HC-Prefix

     Specifies the prefix that will be used for SRDF Host Component API commands. The default value set by GDDR is null, and must be replaced by a valid prefix during GDDR customization.

     If you run multiple GDDR instances on the same z/OS system, then each instance must have its own unique HC-Prefix.

- Work Unit

  Indicates the user-specified device esoteric value that represents the storage device type to be used when GDDR dynamically allocates temporary datasets. The default value is SYSALLDA.

  **Note:** If you wish to change the default from SYSALLDA, ensure that the replacement device esoteric is not defined to z/OS using the 'VIO' attribute.

- Jobname Prefix and Enforce

  Specifies a 3-character prefix that is used as the first three characters of GDDR jobnames. When you provide the jobname prefix and enter **Y** in the Enforce field, GDDR internally submitted jobs will use the jobname prefix specified in this panel. Any attempts to start a script from the Select Script to Run panel (S) using a different jobname prefix will fail.

  If no prefix is defined, or the prefix is not enforced, GDDR uses the 3 first characters of the user-provided jobname as a prefix, or "GDD" if no jobname can be identified in the job cards you provided in the GDDRJCL*n* variables.

  If you run multiple GDDR instances on the same z/OS system, it is strongly recommended that each instance is configured with its own unique Jobname Prefix.

- Work HLQ

  Specifies a dataset high-level qualifier with a maximum length of 17 characters that is used for work datasets. The default is GDDR. "Summary of RACF permissions" on page 131 provides recommended authorizations for this HLQ.

  If you run multiple GDDR instances on the same z/OS system, it is strongly recommended that each instance is configured with its own unique Work HLQ.

- Surrogate User ID and Enforce

  Specifies the user ID that has been authorized to all of the resources required by GDDR processes. It is recommended to specify the user ID provided in the ADDUSER statement of the GDDCRACF C-system RACF definitions, as described in "Set up GDDR security" on page 130.

  When you provide the user ID and enter **Y** in the Enforce field, GDDR internally-submitted jobs will use the surrogate user ID specified in this panel. Any attempts to start a script from the Select Script to Run panel (S) using a different user ID will fail.

- Enter Jobcards

  These fields specify the job cards that will be used when a job is submitted internally by GDDR. Always include the REGION=0M job card parameter.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Utility Parameters panel (M,P,O,U).

## Specify utility parameters (M,P,O,U)

When you specify option **U** in the Specify GDDR Options panel (M,P,O), the Utility Parameters panel is displayed:

```
------------------------ GDDR - Utility Parameters ----------------------------
Command ===>

Press <F3> to return to the Specify GDDR Options Menu
<F5> Script JCL Parameters              <F6> Tuning Values
Type <SAVE> to save, <LOAD> to restart from last saved values.

Enter the required information. Press <Enter> to Validate.
                                                                    More:     +
   EMCGROUP Allocation Parameters      EMCTF Parameters
   - SYSIN    : TRACKS,1,1             - SYSIN  : TRACKS,5,2
   - REPORT   : CYLINDERS,1,1          - SYSOUT : TRACKS,50,50
   - SYSPRINT : CYLINDERS,1,1          - MAXREQ : 20000

   RPTOUT Allocation Parameters
   - SCFRDFME : TRACKS,5,2
   - SCFRDFM6 : TRACKS,5,2
   - SCFRDFM9 : TRACKS,5,2

   Utility Names
   - IEBCOPY      : IEBCOPY
   - IEBGENER     : IEBGENER
   - SDDF Cleanup : GDDFCLN1
   - SDDF List    : GDDFLIST

   eSTEM exec dsname for GDDR exec: ( Default: Prefix.STEM.REXX(ESFRRUN) )
   EMC.MFE830.ISPELIB(ESFRRUN)
******************************** Bottom of Data ********************************
```

**Figure 82** Utility Parameters panel (M,P,O,U)

GDDR script processing sometimes requires invoking external utilities, which have their own space allocation requirements and control settings. This is done using values supplied by the Utility Parameters panel (M,P,O,U).

1. View, define, or modify the panel fields as follows:
   - EMCGROUP Allocation Parameters
     - SYSIN: Recommended allocation is TRACKS,1,1
     - REPORT: Recommended allocation is CYLINDERS,1,1
     - SYSPRINT: Recommended allocation is CYLINDERS,1,1
   - EMCTF Parameters
     - SYSIN: Recommended allocation is TRACKS,5,2
     - SYSOUT: Recommended allocation is TRACKS,50,50
     - MAXREQ: Recommended value is 20000
   - RPTOUT Allocation Parameters
     - SCFRDFME: Recommended allocation is TRACKS,5,2
     - SCFRDFM6: Recommended allocation is TRACKS,5,2
     - SCFRDFM9: Recommended allocation is TRACKS,5,2
   - ECGUTIL Parameters
     - COMMAND: Recommended allocation is CYLINDERS,10,10

  - Number of Tasks: Recommended value is 22
  - MSGLEVEL: Recommended value is 5
- Utility Names

  The Utility Names fields allow site-specific customization of these utilities. JCL generated for parameter load validation and activation jobs includes the specified names for IEBCOPY and IEBGENER.

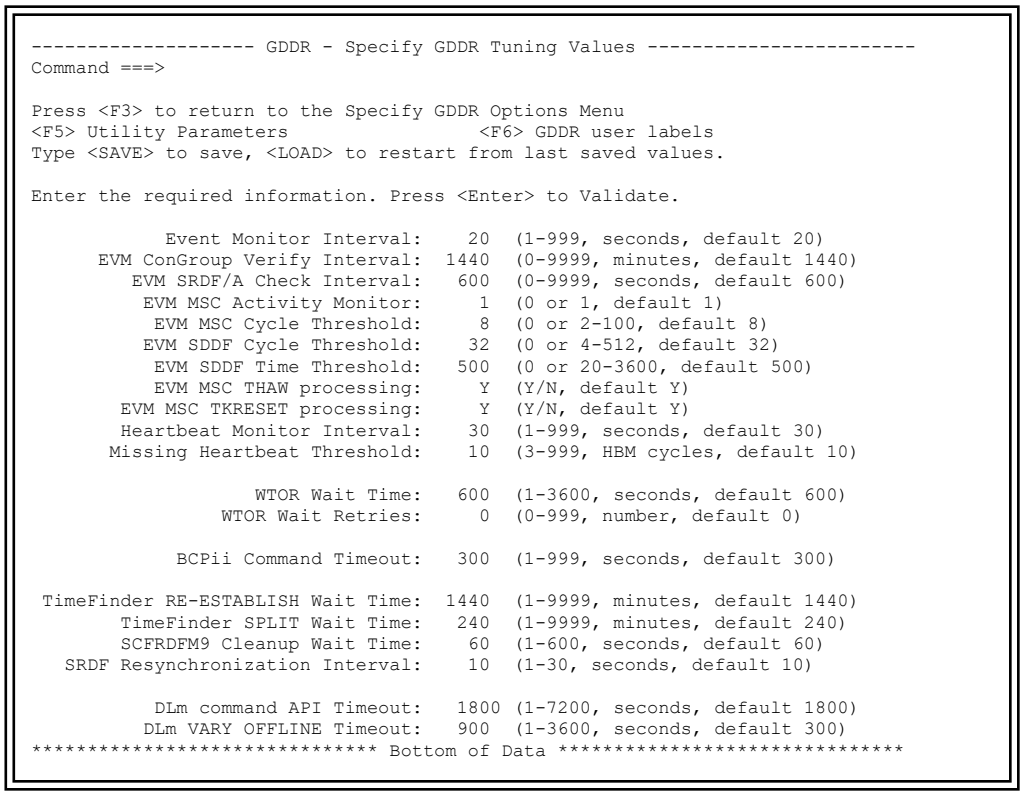
  - IEBCOPY: IEBCOPY
  - IEBGENER: IEBGENER
  - SDDF Cleanup: GDDFCLN1
  - SDDF List: GDDFLIST
- eSTEM exec dsname for GDDR exec

  This is the full name of the exec to be executed when option ST: eSTEM is selected in the Primary Options Menu panel.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Specify GDDR Tuning Values panel (M,P,O,T).

## Specify tuning values (M,P,O,T)

When you specify option **T** in the Specify GDDR Options panel (M,P,O), the Specify GDDR Tuning Values panel is displayed:



```
-------------------- GDDR - Specify GDDR Tuning Values ------------------------
Command ===>

Press <F3> to return to the Specify GDDR Options Menu
<F5> Utility Parameters                 <F6> GDDR user labels
Type <SAVE> to save, <LOAD> to restart from last saved values.

Enter the required information. Press <Enter> to Validate.

           Event Monitor Interval:    20  (1-999, seconds, default 20)
     EVM ConGroup Verify Interval:  1440  (0-9999, minutes, default 1440)
        EVM SRDF/A Check Interval:   600  (0-9999, seconds, default 600)
         EVM MSC Activity Monitor:     1  (0 or 1, default 1)
          EVM MSC Cycle Threshold:     8  (0 or 2-100, default 8)
         EVM SDDF Cycle Threshold:    32  (0 or 4-512, default 32)
          EVM SDDF Time Threshold:   500  (0 or 20-3600, default 500)
          EVM MSC THAW processing:     Y  (Y/N, default Y)
       EVM MSC TKRESET processing:     Y  (Y/N, default Y)
       Heartbeat Monitor Interval:    30  (1-999, seconds, default 30)
      Missing Heartbeat Threshold:    10  (3-999, HBM cycles, default 10)

                  WTOR Wait Time:   600  (1-3600, seconds, default 600)
                WTOR Wait Retries:     0  (0-999, number, default 0)

            BCPii Command Timeout:   300  (1-999, seconds, default 300)

 TimeFinder RE-ESTABLISH Wait Time:  1440  (1-9999, minutes, default 1440)
       TimeFinder SPLIT Wait Time:   240  (1-9999, minutes, default 240)
       SCFRDFM9 Cleanup Wait Time:    60  (1-600, seconds, default 60)
  SRDF Resynchronization Interval:    10  (1-30, seconds, default 10)

          DLm command API Timeout:   1800 (1-7200, seconds, default 1800)
         DLm VARY OFFLINE Timeout:   900  (1-3600, seconds, default 300)
****************************** Bottom of Data ******************************
```

**Figure 83** Specify GDDR Tuning Values panel (M,P,O,T)

When the Specify GDDR Tuning Values panel (M,P,O,T) is initially displayed, all values are blank. When you press **Enter**, all fields are populated with the defaults.

---

**Note:** Regardless of the settings in the Specify GDDR Tuning Values panel (M,P,O,T) fields, GDDR Event Monitor MSC activity monitoring is only performed after MSC is seen to be activated for the GDDR-managed group or groups, and stops when MSC is disabled for these groups.

---

1. View, define, or modify the panel fields as follows:
   - Event Monitor Interval

     Specify the time, in seconds, that the GDDR Event Monitor on each C-system waits between successive checks of the various GDDR event indicators. The value must be a number between 1 and 999. The default is 20 seconds.
   - EVM ConGroup Verify Interval

     Specify the time, in minutes, that the GDDR Event Monitor waits between successive ConGroup verify commands. The default value is 1440 minutes. A value of 0 disables the feature. Non-zero values lower than 60 are rounded up to 60.
   - EVM SRDF/A Check Interval

     Specify the time, in seconds, that the GDDR Event Monitor waits between successive polls of the configured SRDF/A groups. The default value is 600 seconds. A value of 0 disables the feature. Non-zero values lower than 600 are rounded up to 600.
   - EVM MSC Activity Monitor

     Controls whether or not the GDDR Event Monitor performs MSC activity monitoring. A setting of 1 causes the GDDR Event Monitor to issue message GDDS227E when the MSC Post Timestamp is 2 or more times older than MSC_CYCLE_TARGET seconds. Message GDDS228E signals the return to normal.

     A setting of 0 suppresses this monitor, and also causes the GDDR Event Monitor to ignore any non-zero value for the MSC Cycle Threshold field and the SDDF Cycle and Time Threshold fields.
   - EVM MSC Cycle Threshold

     Controls if and when the GDDR Event Monitor issues E-level messages for delays in MSC cycle switching, with allowed values 0 and range 2-100. For example, with a value of 5, the GDDR Event Monitor issues message GDDS229E when the most recent MSC cycle switch TOD value is 5 or more times older than MSC_CYCLE_TARGET seconds. Message GDDS230E signals the return to normal.

     A setting of 0 suppresses this monitor.
   - EVM SDDF Cycle Threshold

     This field expresses the threshold delay as a number of MSC cycle switches that have occurred between SDDF resets for any group of sessions (for example, B1 or B2 for Concurrent SRDF/Star configurations), with allowed

values 0 and range 4-512. Values of 4-8 should only be used for demonstration purposes. A value of 32 is recommended as a starting value for production environments. If SDDF resets are seen more MSC cycles apart than the number specified in this field, the GDDR Event Monitor issues message GDDS231E. Message GDDS232E signals the return to normal.

A setting of 0 suppresses this monitor.

- EVM SDDF Time Threshold

  If there is a serious delay in MSC cycle switching, the number of MSC cycle switches in between SDDF resets is not adequate to provide timely warning of an increased SRDF/SQAR differential. The EVM SDDF Time Threshold field expresses the threshold delay as a number of seconds in between SDDF resets, with allowed values 0 and range 20-3600.

  The value of the EVM SDDF Cycle Threshold can be used with the MSC_CYCLE_TARGET time for guidance on the EVM SDDF Time Threshold field. For example, if the MSC_CYCLE_TARGET time is 5 seconds, then values of 20-40 should only be used for demonstration purposes. A value of 160 (32 times 5) would then be the recommended starting value for production environments. If SDDF resets are seen more seconds apart than the number specified in this field, the GDDR Event Monitor issues message GDDS233E. Message GDDS234E signals the return to normal.

  A setting of 0 suppresses this monitor.

- EVM MSC THAW Processing

  Enables (default) or disables MSC THAW processing.

  **Note:** This setting applies to the SRDF/SQAR configuration with GDDR configured for High Availability (HA).

  "Support for MSC THAW processing" on page 74 describes MSC THAW processing.

- EVM MSC TKRESET Processing

  Enables (default) or disables MSC TKRESET processing.

  **Note:** This setting applies to the SRDF/SQAR configuration with GDDR configured for High Availability (HA).

  MSC TKRESET processing helps prevent unnecessary MSC takeovers in cases when a secondary MSC server signals possible loss of the primary MSC server but this is a false alarm.

  With MSC TKRESET processing enabled, the GDDR Event Monitor assesses the situation and responds as follows:

  – If the primary MSC server is inactive, MSC takeover is performed (script GDDRUP35).

  – If the primary MSC server is in an active state, GDDR runs the MSC,TKRESET command to reset the takeover counter on the secondary MSC server to zero (0).

When MSC TKRESET processing is disabled, the GDDR Event Monitor responds by submitting the GDDRUP35 script to perform the MSC takeover.

- Heartbeat Monitor Interval

  Specify the time, in seconds, that the GDDR Heartbeat Monitor on each C-system waits before setting and propagating its new heartbeat value. The value must be a number between 1 and 999. The default is 30 seconds.

- Missing Heartbeat Threshold

  Specify the number of times a GDDR Heartbeat Monitor on a C-system will need to detect no change in the heartbeat value of another C-system, upon awakening from its own wait interval, before it will declare the other C-system dead. The value must be a number from 3 to 999. The default value is 10.

- WTOR Wait Time

  Specify the number of seconds that a GDDR script will wait for an operator reply to a WTOR it has issued. When the specified interval has expired, the WTOR is deleted and the script proceeds as if the operator had replied 'N' or 'CANCEL' depending upon the particular message.

  This parameter is optional. If not specified, the default value of 600 is used, which is equivalent to 10 minutes.

- WTOR Wait Retries

  Specify the number of times to re-issue the same WTOR. This optional parameter allows you to prolong the time GDDR has to wait for a certain event.

  The GDDR WTOR module accept the RETRIES argument, which specifies the number of times to re-issue the same WTOR when intercepting a timeout condition. You can end the wait by replying negatively to the WTOR message. By default (WTOR Wait Retries=0) the WTOR will be issued just once, resulting in script termination with RC=32 if a timeout occurs.

  **Note:** This parameter is used during all MSC state changing operations.

- BCPii Command Timeout

  This setting applies to all BCPii commands. If a BCPii command times out during a GDDR script, the command result indicates 'ERROR(TIMEOUT)'. In the case of reoccurring timeouts, the BCPii Command Timeout setting may need to be increased.

  The value must be a number between 1 and 999. The default value is 300 seconds.

- TimeFinder RE-ESTABLISH Wait Time

  Specify the number of minutes that is used when TimeFinder ESTABLISH commands specify a WAIT parameter. The default is 1440 minutes.

- TimeFinder SPLIT Wait Time

  Specify the number of minutes that is used when TimeFinder SPLIT commands specify a WAIT parameter. The default is 240 minutes.

- SCFRDFM9 Cleanup Wait Time

  Specify the number of seconds that GDDR waits for SRDF/A cleanup to complete after the SCFRDFM9 utility completes. The value must be a number between 1 and 600. The default is 60 seconds.

- SRDF Resynchronization Interval

  Specify the number of minutes that GDDR waits for invalids to synchronize. The value must be a number between 1 and 30. The default is 10 minutes.

- DLm command API Timeout

  Specify the number of seconds that GDDR is to wait for a response to a command sent to a DLm Access Control Point (ACP) via the TCP/IP command API. The default is 1800 seconds.

- DLm VARY OFFLINE Timeout

  Specify the number of seconds that GDDR is to wait for all DLm virtual tape drives to be varied offline. The default is 300 seconds.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to go to the Define GDDR User Labels panel (M,P,O,L).

## Define GDDR user labels (M,P,O,L)

When you specify option **L** in the Specify GDDR Options panel (M,P,O), the Define GDDR User Labels panel is displayed:

```
------------------------ GDDR - Define GDDR User Labels -----------------------
Command ===>

 Press <F3> to return to the Specify GDDR Options Menu
 <F5> Tuning Values                      <F6> Parameter Management Options Menu
 Type SAVE to save, LOAD to restart from last saved values.

      GDDR Complex name ===> ________ (default: GDDRPLEX)

      Site  User label (uuuuuuu)              Region  User Label (uuuuuuu)
      ----  ----------                        ------  ----------
       DC1    ________  (7 chars max)           RG1    _________ (7 chars max)
       DC2    ________
       DC3    ________                         RG2    _________
       DC4    ________
 DISPLAY FORMAT:
      User site display format ===> 0        User region display format  ===> 0
   Format Template:  Example:                Format Template:  Example:
      0 - DC#        DC1      (default)         0 - RG#        RG1    (default)
      1 - uuuuuuu#   London1                   1 - uuuuuuu#   Europe1
      2 - #uuuuuuu   1London                   2 - #uuuuuuu   1Europe
      3 - DC#-uuuu   DC1-Lond                  3 - RG#-uuuu   RG1-Euro
      4 - uuuu-DC#   Lond-DC1                  4 - uuuu-RG#   Euro-RG1
```

**Figure 84** Define GDDR User Labels panel (M,P,O,L)

The Define GDDR User Labels panel accepts user-specified names for the GDDR-plex, regions, and sites.

1. To proceed, enter the following values:

- GDDR Complex name

  A name assigned by the user to the entire complex of C-systems and managed systems as follows:

  - Format: 8 characters, alphanumeric
  - If no name is specified, the default is 'GDDRPLEX'.

- Site and User label

  Provide site names as follows:

  - Length: 1-7 characters, alphanumeric
  - If no names are specified, Format '0' is assumed; sites will be displayed as DC1, DC2, DC3, and DC4.

- User site display format

  Specify the number of the template which will determine how site user labels will be shown in panels and messages.

- Region and User label

  Provide the region names as follows:

  - Length: 1-7 characters, alphanumeric
  - If no names are specified, Format '0' is assumed; regions will be displayed as RG1 and RG2.

- User region display format

  Specify the number of the template which will determine how region user labels will be shown in panels and messages.

2. If you made any changes, type **Save** on the command line and press **Enter**.
3. Press **F6** to return to the Parameter Management Options Menu panel (M,P).

## Validate GDDR parameter set (M,P,V)

GDDR parameter validation determines that your proposed changes to parameter values are consistent and relationally correct. The validation operations are similar to the activation operations described in "Activate GDDR parameter set (M,P,A)" on page 354, but do not update any GDDR variables.

When you specify option **V** in the Parameter Management Options Menu panel (M,P), the Validate GDDR Parameter Set panel is displayed:

```
---------------------- GDDR - Validate GDDR Parameter Set ----------------------
 Option ===>


 Options below in effect for VALIDATION

    Specify Options for this Parameter Set Validation :

  Validation in Foreground or Background : FORE        (FORE,BACK)
        Specify GDDR Parameter Load Type : _______     (PARTIAL,FULL)
     Specify GDDR State Variables Action : ________    (RESET, ASIS, NOUPDATE)
            Propagate to Other C-systems : ___         (YES/NO/TRY)


    Issue one of the following commands:

       SAVE : Save options above as the default for your userid
     CANCEL : <F3> return to the GDDR Parameter Management Options Menu
        VAL : Proceed with GDDR Parameter Validation using options above
```

**Figure 85** Validate GDDR Parameter Set panel (M,P,V)

Complete the options listed in this panel to validate your parameter set.

1. View, define, or modify the panel fields as follows:
   - Validation in Foreground or Background
     - FORE: Processing occurs in the TSO foreground.
     - BACK: Processing occurs in the background as a batch job. A panel is displayed containing the job card and the PROCLIB and SKELETON libraries previously specified in your personal GDDR ISPF profile.

     Modify these settings if needed and then press **Enter**.
   - Specify GDDR Parameter Load Type
     - PARTIAL: Validate only the changed GDDR Parameter Wizard work dataset members. There is one work dataset member per parameter definition panel. Validation does not include contents of other members.

       **Note:** Partial validation is not currently supported to implement a change in the names or the number of managed systems or LPARs.
     - FULL: Validate the entire set of GDDR parameters. This is the recommended setting.
   - Specify GDDR State Variables Action
     - RESET: Reset all state variables to a neutral (non-error) state. This is the recommended setting.
     - ASIS: Set all state variables to the values found in the input.
     - NOUPDATE: Do not update any state variables. This option is invalid with Parameter Load type “FULL”.
   - Propagate to Other C-systems
     - YES: Send the updates to other C-systems. If any other C-system cannot be contacted, the action will fail. This is the recommended setting.

- NO: Do not send the updates to other C-systems.
- TRY: Try to send the updates to other C-systems, but if they cannot be contacted, continue.

2. After specifying the requested options, type the **Save** command to save your validation options to your TSO profile.
3. Type **VAL** to validate the proposed parameter definitions, or **CANCEL** to return to the Parameter Management Options Menu panel (M,P).

   If Background mode is selected, the following confirmation message displays:

   ```
   GDDI116I Job for Parameter Validation submitted successfully
   ```

4. Following a successful validation, press **F3** to return to the Parameter Management Options Menu panel (M,P).

## Activate GDDR parameter set (M,P,A)

To complete the parameter definition and load process, you need to activate the GDDR parameter set.

IMPORTANT

Parameters required by the GDDR Event Monitor and GDDR Heartbeat Monitor may be changed during parameter activation. For this reason, GDDRMAIN will stop the GDDR Event Monitor and GDDR Heartbeat Monitor if they are not already stopped before you perform a parameter activation. If they were started before the parameter activation, GDDRMAIN will restart the GDDR Event Monitor and GDDR Heartbeat Monitor after the activation completes.

The parameter activation process performs the following actions:

- If a GDDR script is running, then only partial activation is allowed, and only if the script job is not currently running.
- Locks variables for integrity and consistency on the local system.
- Performs an implicit backup and validation before activation.
- Updates variables and propagates them to other C-systems, depending on the Activate GDDR Parameter Set panel (M,P,A) setting.
- Performs an implicit backup after activation.
- Clears the command queue if requested.
- Releases the variables.
- Copies the GDDR Parameter Wizard work dataset to the last activated parameter dataset.

IMPORTANT

If you are running the parameter activation batch job with CONSISTENCY(ENFORCE) and you cancel it after it has started the load, then you may need to disable the dynamic exit called GDDREXG1*nnnn*. The same is true for a parameter backup job if you are running with any CONSISTENCY option other than IGNORE. "Dynamic exits" on page 143 describes how to manually disable the GDDREXG1*nnnn* dynamic exit.

Specify option **A** in the Parameter Management Options Menu panel (M,P). The Activate GDDR Parameter Set panel is displayed:

```
---------------------- GDDR - Activate GDDR Parameter Set ----------------------
Command ===>


Options below in effect for ACTIVATION


   Specify Options for this Parameter Set Activation :

 Activation in Foreground or Background : BACK        (FORE,BACK)
       Specify GDDR Parameter Load Type : FULL        (PARTIAL,FULL)
    Specify GDDR State Variables Action : RESET       (RESET, ASIS, NOUPDATE)
           Propagate to Other C-systems : YES         (YES/NO/TRY)
                    Enforce consistency : RETRY=5     (YES/NO/RETRY=1-5)
                  Ignore Backup Failure : NO          (YES/NO)
           Clear the GDDR Command Queue ? YES         (YES/NO)

   Issue one of the following commands:

      SAVE : Save options above as the default for your userid
    CANCEL : <F3> return to the GDDR Parameter Management Options Menu
       ACT : Proceed with GDDR Parameter Activation using options above
      VIEW : View last activation's run log
```

**Figure 86** Activate GDDR Parameter Set panel (M,P,A)

Complete the options listed in this panel to activate your parameter set.

1. View, define, or modify the panel fields as follows:
   - Validation in Foreground or Background
     - FORE: Processing occurs in the TSO foreground. Activation processing messages are displayed in the Parameter Load Activation status pop-up panels shown in Figure 87 and Figure 88 on page 357.
     - BACK: Processing occurs in the background as a batch job. This is the recommended setting. A panel is displayed containing the job card and the PROCLIB and SKELETON libraries previously specified in your personal GDDR ISPF profile. Modify these settings if needed and then press **Enter**.
   - Specify GDDR Parameter Load Type
     - PARTIAL: Activates only the changed GDDR Parameter Wizard work dataset members. There is one work dataset member per parameter definition panel. Activation does not include contents of other members.

       **Note:** Partial validation is not currently supported to implement a change in the names or the number of managed systems or LPARs.
     - FULL: Deletes all global variables and activates the entire GDDR parameter set. This is the recommended setting.

- Specify GDDR State Variables Action
  - RESET: Reset all state variables to a neutral (non-error) state. This is the recommended setting.
  - ASIS: Set all state variables to the values found in the input.
  - NOUPDATE: Do not update any state variables. This option is invalid with Parameter Load type “FULL”.
- Propagate to Other C-systems
  - YES: Send the updates to other C-systems. If any other C-system cannot be contacted, the action will fail. This is the recommended setting.
  - NO: Do not send the updates to other C-systems.
  - TRY: Try to send the updates to other C-systems, but if they cannot be contacted, continue.
- Enforce consistency
  - YES: Any updates to global variables other than by parameter activation will be blocked while the backup or parameter activation is in progress. This is the recommended setting.
  - NO: Outside updates to globals will be ignored.
  - RETRY (1-5): If an outside update occurs while parameter backup is running, the backup will be retried the indicated number of times.
- Ignore Backup Failure
  - YES: If the parameter backup which is done before the parameter load fails, the parameter load will continue.
  - NO: If the parameter backup which is done before the parameter load fails, the parameter load will not be done. This is the recommended setting.
- Clear the GDDR Command Queue
  - YES: The command queue is cleared if the parameter load is successful. Use this choice for a full parameter load. This is the recommended setting.
  - NO: The command queue is not cleared. This choice may be appropriate for a partial parameter load.

  Clearing the command queue is attempted on all C-systems if Propagate to Other C-Systems option is 'Yes' or 'Try' and all C-systems are in communication with one another. If the choice is to clear the command queue, but communication between the C-systems is temporarily interrupted, use the Manage Internal Command Queue panel (M,Q) to clear it on the other C-systems, or use the GDDRCLRQ job provided in *hlq*.GDDR*vrm*.SAMPLIB.

  If no script is in progress, it is recommended to clear the command queue. If a script is in progress, consult GDDR Solution Support.

2. After specifying the requested options, type the **SAVE** command to save your activation options to your TSO profile.

3. Type **ACT** to activate the proposed parameter definitions, or **CANCEL** to return to the Parameter Management Options Menu panel (M,P).

   The following message is displayed:

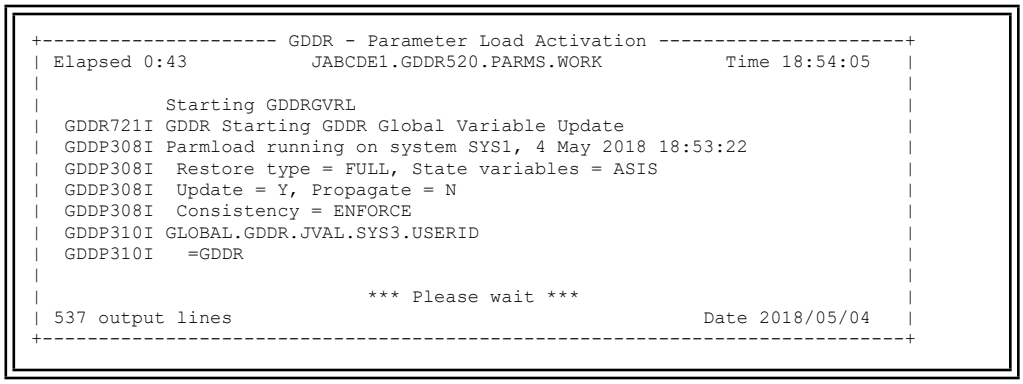
```
+-------------------------------------------------------------------------------+
| GDDR387E Parm activation job started (Job GDDRPACT J0003283) - Use ACT FORCE |
| to start another activation                                                   |
+-------------------------------------------------------------------------------+
```

   If you already have started an activation job that did not complete yet, or was canceled, aborted, or never was run, you will not be able to submit another job (see the message below). In this case you must either wait for the activation job to complete, or you can issue either the ACT FORCE command to force a resubmission of a new activation job or issue ACT RESET to reset the status (if the job was canceled or aborted or was never run). ACT RESET does not submit another activation job, but *will* allow you to edit parameters again once you exit the activation panel (parameter editing is not allowed while an activation job is in progress).

4. If Background activation mode is selected, the following confirmation message is displayed:

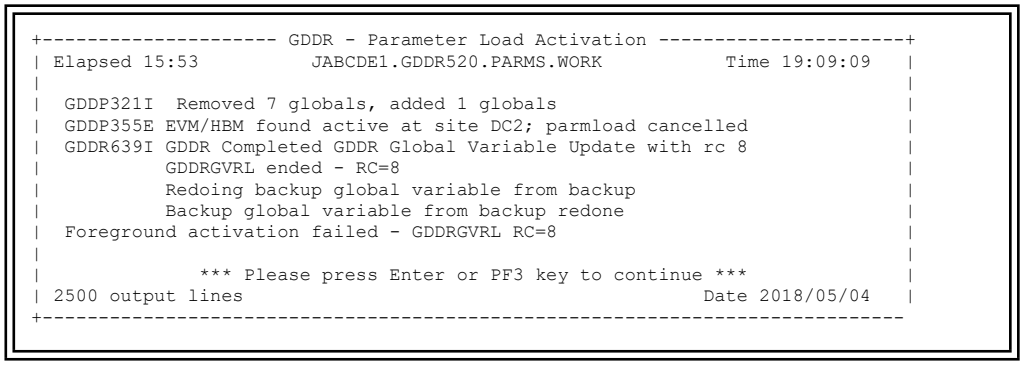
```
GDDI116I Job for Parameter Activation submitted successfully
```

5. If Foreground activation mode is selected, activation processing messages are displayed in the Parameter Load Activation status pop-up panels, for example:



```
+---------------------- GDDR - Parameter Load Activation -----------------------+
| Elapsed 0:43           JABCDE1.GDDR520.PARMS.WORK           Time 18:54:05    |
|                                                                               |
|           Starting GDDRGVRL                                                   |
|  GDDR721I GDDR Starting GDDR Global Variable Update                           |
|  GDDP308I Parmload running on system SYS1, 4 May 2018 18:53:22                |
|  GDDP308I  Restore type = FULL, State variables = ASIS                        |
|  GDDP308I  Update = Y, Propagate = N                                          |
|  GDDP308I  Consistency = ENFORCE                                              |
|  GDDP310I GLOBAL.GDDR.JVAL.SYS3.USERID                                        |
|  GDDP310I   =GDDR                                                             |
|                                                                               |
|                             *** Please wait ***                               |
| 537 output lines                                          Date 2018/05/04     |
+-------------------------------------------------------------------------------+
```

**Figure 87** Parameter Load Activation status panel 1 of 2



```
+---------------------- GDDR - Parameter Load Activation -----------------------+
| Elapsed 15:53          JABCDE1.GDDR520.PARMS.WORK           Time 19:09:09    |
|                                                                               |
|  GDDP321I  Removed 7 globals, added 1 globals                                 |
|  GDDP355E EVM/HBM found active at site DC2; parmload cancelled                |
|  GDDR639I GDDR Completed GDDR Global Variable Update with rc 8                |
|           GDDRGVRL ended - RC=8                                               |
|           Redoing backup global variable from backup                          |
|           Backup global variable from backup redone                           |
|  Foreground activation failed - GDDRGVRL RC=8                                 |
|                                                                               |
|              *** Please press Enter or PF3 key to continue ***                |
| 2500 output lines                                         Date 2018/05/04     |
+-------------------------------------------------------------------------------
```

**Figure 88** Parameter Load Activation status panel 2 of 2

6. Press **Enter** or **PF3** to return to the Activate GDDR Parameter Set panel (M,P,A).

The following message is displayed:

```
GDDI378E Parm activation failed, return code = 8 - Use VIEW command
to see log
```

Type the **VIEW** command at the command prompt to review the messages pertaining to activation processing.

## Set message, debug and trace options (M,D)

When you specify option **D** in the Setup and Maintenance Menu panel (M), the Set Output Message Levels by Program panel is displayed:

```
--------- GDDR - Set Output Message Levels By Program ------------------------
Option ===>                                                 Scroll ===> CSR
This panel shows the message, debug and trace output levels in effect for user
shown.  Levels on each line apply to program on that line only, while levels
in parentheses are defaults applying to any program not found in the list.

You may change the defaults or the levels for specific programs by overtyping.
Use ADD to add a new program to the list with initial output levels.
Press <F3> to save changes and return to previous panel
Press <F1> for a complete description of available actions on this panel

   Program   Msg ( 1 )  Debug ( 0 )  Trace ( 0 )  For userid:     JABCD1
-  --------  ---        -----        -----        HMC Simulation? N  ( Y or N )
                                                  BCPii Tracing?  N  ( Y or N )
****************************** Bottom of data ********************************
```

**Figure 89** Set Output Message Levels by Program panel (M,D)

This panel enables you to individually customize message, debug, and trace settings for a selected user ID and GDDR module.

- HMC Simulation

  Performs operations up to the point of the actual submit of BCPii interface functions from the Perform HMC LPAR Actions panel (A,L) or from within a script. The BCPii interface functions used by GDDR are LIST, QUERY, COMMAND, SET, CONNECT and DISCONNECT.

- BCPii Tracing

  When set to 'Y', detailed BCPii diagnostic information will be written to the primary sysout dataset for the GDDRWORK address space on the appropriate C-system. Keep in mind this might not be the C-system on which the original task was initiated (either via scripting, batch work, or the ISPF interface). This is the system upon which the task is executed. This can be very useful when trying to diagnose a problem with HMC management.

The defaults for these options are the recommended settings for your production environment. You may be asked to make changes to the defaults if diagnostic information is needed as a result of a question or problem.

Use the **ADD** command to make changes to the default settings. The Add Program to MsgLevel/Debug/Trace List panel is displayed:

```
--------------- GDDR - Add Program to MsgLevel/Debug/Trace List ------
 Command ===>

             Program ===> ________        MsgLevel    ===> 1
                                          Debug Level ===> 0
                                          Trace Level ===> 0

       Enter program name (required)
       You may overtype default message, debug and trace levels
       Press <Enter> when ready to add new program to the list and return
       Press <F3> to return without adding a program to the MDT list
```

**Figure 90** Add Program to MsgLevel/Debug/Trace List panel

This panel allows the program names provided by Dell EMC Customer Service to be specified with the requested message level, debug, or trace flags.

If the program name is less than eight characters, extend the name with "$" signs up to a length of eight characters.

## Manage GDDR internal command queue (M,Q)

When you specify option **Q** in the Setup and Maintenance Menu panel (M), the Manage Internal Command Queue panel is displayed. If no GDDR script is currently running, the panel shows totals as follows:

```
----------------- GDD5 - Manage  Internal Command Queue -----------------------
  Option ===>

      Total elements:  0     Active elements: 0      Maximum active: 8


  Press <F3> to return to the GDDR Setup and Maintenance Menu
```

**Figure 91** Manage Internal Command Queue panel (M,Q) when no script is running

When a GDDR script is running, the panel shows commands present in the queue and looks similar to the following:

```
---------------------- GDDR - Manage Internal Command Queue -- Row 1 to 2 of 2
Option ===>                                                    Scroll ===> CSR

  WARNING: Do not use unless instructed to do so by Dell EMC Support

    Total elements: 2     Active elements: 0     Maximum active: 35
    Entries listed in order by script name (A)

Press <F3> to return to the GDDR Setup and Maintenance Menu

Sel  No  RetCode  Script    Created            Updated
---  --  -------  --------  -----------------  --------------------
 _    1  152      GDDRPA07  0530201106023291   0530201110255782
       SC VOL,LCL(BA14,18),HSWAP(FORCE,STAR,GDDR),ALL
 _    2  152      GDDRPA07  0530201106023320   0530201110255163
       SC VOL,LCL(CC12,20),HSWAP(FORCE,STAR,GDDR),ALL
****************************** Bottom of data ********************************
```

**Figure 92** Manage Internal Command Queue panel (M,Q) when a script is running

**⚠WARNING**

**The Manage Internal Command Queue panel (M,Q) provides the capability to alter GDDR processing. Dell EMC advises against use of this panel unless specifically directed by GDDR Customer Support.**

The following command line commands are available:

- **Refresh**—Display from the actual command queue elements.
- **Clear Queue**—Clear the entire queue.
- **SORT NUM D[A]**—Sort by command number, Ascending [Descending].
- **SORT SCR A[D]**—Sort by script name, Ascending [Descending].
- **SORT CRE A[D]**—Sort by date/time created, Ascending [Descending].
- **SORT UPD A[D]**—Sort by last update date/time, Ascending [Descending].
- **SORT QUE A[D]**—Sort by queue order, Ascending [Descending].

The initial sort order is Command number, Descending.

Message GDDI041E is issued if the command is not valid.

The display is kept in sync with actions; there is no need to refresh the display after a delete.

## Perform HMC discovery (M,H)

To discover HMC objects accessible to GDDR, specify option **H** in the Setup and Maintenance Menu panel (M). A pop-up dialog is displayed and HMC object discovery is activated:

```
+------- Discovering HMC Objects ---------+
|                                          |
|   Discovering HMC objects at site DC1    |
|                                          |
|           *** PLEASE WAIT ***            |
|                                          |
+------------------------------------------+
```

**Figure 93** Discovering HMC Objects panel

When the discovery operation completes, the HMC Discovery Results panel is displayed:

```
---------------------- GDDR - HMC Discovery Results ------- Row 1 to 16 of 16
Command ===>
Enter <F3> to return to the previous menu
------------------------------------------------------------------------------
GDDR > BEGIN OUTPUT FROM SYSTEM LB01M34
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR CPC   :IBM390PS.M2964
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR LPAR  :M3A
GDDR >   **UNABLE TO ESTABLISH CONNECTION TO LPAR :K148
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR LPAR  :M34
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR LPAR  :M3B
GDDR > END OUTPUT FROM SYSTEM LB01M34, RC=0
GDDR > BEGIN OUTPUT FROM SYSTEM LB01M3A
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR CPC   :IBM390PS.M2964
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR LPAR  :M3A
GDDR >   **UNABLE TO ESTABLISH CONNECTION TO LPAR :K148
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR LPAR  :M34
GDDR >   GDDR BCPII DISCOVERY VALIDATED FOR LPAR  :M3B
GDDR > END OUTPUT FROM SYSTEM LB01M3A, RC=0
> Unable to get connectivity information from LB01K145
>  for CPC IBM390PS.K13906: no HMC workers
***************************** Bottom of data ******************************
```

**Figure 94** HMC Discovery Results panel (M,H)

When you are finished examining the results of the discovery operation, press **F3** to return to the Setup and Maintenance Menu panel (M).

## Refresh GDDR message table (M,R)

When you specify option **R** in the Setup and Maintenance Menu panel (M), the GDDRMSG table is refreshed and the message 'GDDRMSG Table refreshed' is displayed in the panel:

```
+---------------------------------+
| GDDR GDDRMSG Table refreshed    |
+---------------------------------+
```

**Note:** After applying maintenance to the GDDR software, in many cases you must refresh GDDRMTXT, which is managed via GDDRMAIN. GDDRMTXT is refreshed daily at midnight, or whenever GDDRMAIN is restarted.

## Manage GDDR system variables (M,S)

GDDR stores parameters that describe the environment as global GDDR system variables. With the exception of the Manage GDDR System Variables panel (M,S), update of GDDR system variables is under the exclusive control of GDDR.

**⚠WARNING**

**The Manage GDDR System Variables panel (M,S) provides the capability to alter GDDR processing. Dell EMC advises against use of this panel unless specifically directed by GDDR Solution Support. Changes done in this panel remain local and they are not propagated to other C-systems.**

To view and change the values that are used by GDDR, specify option **S** in the Setup and Maintenance Menu panel (M).

A Manage GDDR System Variables panel similar to the following is displayed:

```
---------------------- GDDR - Manage GDDR System Variables -- Row 1 to 44 of 44
  Command ===>                                                  Scroll ===> CSR

  Global variable level: GLOBAL.GDDR.
  Use the <ADD> command to create a GDDR System Variable
  When finished, press <F3> to return to previous level or menu

  Line commands: S elect, E dit, V iew, Z Delete level, D elete item

  Actn Node                  Nodes  Value
  ------------------------------------------------------------------------------
    _   ALLOC                    9
    _   AUTOMATION               1
    _   BKUPINFO               126
    _   CG                       1
    _   CONFIG                  18
    _   CURRENT                 15
    _   CURRMAST                 1
    _   DATA                     9
    _   DCN                     31
    _   DC1                     23
    _   DC2                     23
    _   DC3                     23
    _   DC4                     23
    _   DIAGNOSTICS              2
    _   DLM                      1
    _   ECGCLEAN                 2
    _   ELIGIBLE                 1
    _   EVENT                    7
    _   EVM                      2
    _   EZSM                     1
    _   GDDR                     1
    _   HEARTBEAT                1
    _   HMC                     12
    _   JVAL                    33
    _   LASTLOAD_DESC            0   (User: <TSOid> 01/26/18 14:52:53)
    _   LASTLOAD_SOURCE          0  SYSU.GDDR520.PARMS.LAPD.M38  on GA2LB148
    _   MDT_LIST                 2
    _   MFEOPTS                  9
    _   MISSING                  1
    _   PARMS                   126
    _   PROTECTION               1
    _   RUN                      1
    _   RUND                     1
    _   SCRIPT                 850
    _   SINGLE                   2
    _   STEM                     1
    _   TEMPVAR                  4
    _   TIMEFINDER               1
    _   USEBCV                   5
    _   USERLABEL                8
    _   USEROPT                  8
    _   USEZDP                   2
    _   UTILITY                  4
    _   WAIT_INTERVAL           13
    _   WTOR_WAIT_INTER+         0  3600
  ****************************** Bottom of data ********************************
```

**Figure 95** Manage GDDR System Variables panel (M,S)

View, define, or modify the panel fields as follows:

- Global variable level

  Specifies the name of the global variable, including the stem name of 'GLOBAL.GDDR'.

- Add

  Type **Add** in the command line at the top of the panel to display the Create GDDR System Variable panel. This panel allows you to create new variables under the direction of GDDR Solution Support.

- Node

  Global variables may have a variable number of subnodes. The Node field displays information about subnodes, such as the number of subnodes and for the lowest level nodes, their value.

- Actn

  For each node listed in the Node field, you can specify the following line commands in the Action field:

  - S—Select

    Shows the next level subnode, if there are subnodes to the specified node.

  - E—Edit

    Allows the variable at the existing level to be changed.

  - V—View

    Displays information about the creation, update, and last access for the global variable.

  - Z—Delete level

    Deletes all variables at or under the level shown.

  - D—Delete item

    Deletes just the variable at the existing level.

## Transfer master C-system (M,T)

When you specify option **T** in the Setup and Maintenance Menu panel (M), the Transfer Master C-System panel is displayed:

```
---------------------- GDDR - Transfer Master C-System ------ Row 1 to 4 of 4
 Command ===>

 Press <F3> to return to the Setup and Maintenance panel
 Select a row to transfer the Master C-System to that site/system.

 CMD   SIte     C-System  Link   MHB  Notes
 ---  --------  --------  ----   ---  ----------------------------------------
  _    DC1       SYS1       0     0
  _    DC2       SYS2       0     0    Current and Recommended Master, *Here*
  _    DC3       SYS3       0     0
  _    DC4       SYS4       0     0
 ****************************** Bottom of data ******************************
```

**Figure 96** Transfer Master C-System panel (M,T)

As shown in Figure 96, based on the current GDDR parameters, site DC2 is the current master C-system and is the recommended site for the master C-system based on the parameters. Note that *Here* indicates the current local system.

No action is recommended. Press **F3** to return to the Setup and Maintenance Menu panel (M).

# View GDDR configuration (G)

Specify option **G** in the Primary Options Menu panel (Figure 34 on page 281) to view your configuration settings.

**Note:** In the View GDDR Configuration panel (G), the RG1 and RG2 region names and the DC1, DC2, DC3, and DC4site names may reflect user-specified names for the GDDR-plex, regions, and sites as defined in the Define GDDR User Labels panel (M,P,O,L). See "Define GDDR user labels (M,P,O,L)" on page 351 for more details.

The View GDDR Configuration panel is displayed:

```
----------------- GDDR - View GDDR Configuration for GDDR_AB ------------------
  Command ===>

    GDDR complex name:  GDDR_AB

     Regions:  RG1 (PRIME1), RG2 (SECON2)
     Sites:    DC1 (DC1-SITE), DC2 (DC2-SITE), DC3 (DC3-SITE), DC4 (DC4-SITE)
     Features: SRDF/SQAR with AutoSwap, MSC-HA
               R22, No DLM support, FBA devices


    Region:          Site:            C-System:

    RG1 (PRIME1)     DC1 (DC1-SITE)   SQRL132

                     DC2 (DC2-SITE)   SQRL133

    RG2 (SECON2)     DC3 (DC3-SITE)   SQRL1146

                     DC4 (DC4-SITE)   SQRL1147



    Press <F3> to return to the previous menu
```

**Figure 97** View GDDR Configuration panel (G)

# Perform GDDR health check (C)

Specify option **C** in the Primary Options Menu panel (Figure 34 on page 281) to perform the GDDR on-demand health check. A Perform Health Check panel similar to the following is displayed, showing the items to be validated:

```
------------------ GDDR - Perform Health Check --------------- Row 1 to 6 of 6
Command ===>                                                    Scroll ===> CSR

Primary Cmds   : ConFiG, GateK, LICense, MAINTenance, MSC,  (PF1 for more info)
                 SubSYS, SUMmary, SYStems, TOPOlogy, DLM SET|RESET, REFresh,
                 SORT ASC|DESC SITE|SYSTEM|TYPE|GDDRMAIN|HMC|DATE|TIME|EVENTS

Active Events  : MSC STR
Expected Events: None
Degraded Mode  : No          Consistency :  Consistent        Star-HA :  0
                                                            DLm Abort :  0
Command Queue  : Status  Job      Count  Active Free   Max
                 ------- -------- ------ ------ ------ ------
                 Empty            0      0      0      8

------------------------ System Communication Status -------------------------
    Home  System       GDDRMAIN   HMC      Last Communicated   System
Sel Site   Name   Type  Status   Status     Date     Time      Events
--- ---- -------- ---- -------- --------- ----------------- -----------
 _  DC1  LB04M41  CSYS Active   n/a       02/02/18 08:55:48 None
 _  DC2  LB04M42  CSYS Active   Operating 02/02/18 08:55:51 None
 _  DC3  CLAYB034 PSYS Inactive Error     Never             None
 _  DC4  LB04K149 CSYS Active   Error     02/02/18 08:55:53 None
```

**Figure 98** Perform Health Check panel (C)

The Perform Health Check panel provides the following functions:

- Primary Commands

  You can enter the following primary commands on the command line:

  **Note:** The following commands can be executed from the Perform Health Check panel but are not listed explicitly on the panel. See the panel help for information about the purpose and parameters of these commands: CHECKUP, ENQ, ParmREFresh, REGION, SET, SVCDUMP. "GDDRMAIN console commands" on page 173 provides a detailed description of these commands.

  - ConFiG *parameters*

    Validates the GDDR-managed configuration, with the following optional parameters:

    - DeBuG — Enable debugging
    - SITE=*site* — Limit scope to SRDF groups including the specified site
    - CTRL=*serial#* — Limit scope to SRDF groups including the specified storage system

    **Note:** "CONFIG" on page 194 provides more information about the CONFIG command.

  - GateK *parameters*

    Validates gatekeepers, with the following optional parameters:

- Auto_UnBoX — Automatically unbox devices
  **Note:** GDDRMCMD must be APF-authorized in order to use the Auto_UnBoX option and should be added to the IKJTSO parameter file under AUTHPGM.
- PRompt — Prompt before unboxing a device
- SIMulate — Simulate unboxing of devices without actually unboxing
- FoRCe — Force unboxing of devices when normally not allowed
- DeBuG — Enable debugging
- ALL — Include paths to storage systems that are not GDDR-managed
- SITE=*site* — Limit scope to gatekeepers at the specified site
- CTRL=*serial#* — Limit scope to gatekeepers on the specified storage system

**Note:** "GATEK" on page 201 provides more information about the GATEK command.

- LICense *parameters*

  Validates licenses across all systems where GDDRMAIN is active, with the following optional parameters:

  - DeBuG — Enable debugging
  - SITE=*site* — Limit license check to systems at the specified site
  - SYStem=*system* — Limit license check to the specified system
  - TiMeOut=*seconds* — Number of seconds to wait for license check per system

    **Note:** Specifying an explicit timeout is not recommended.

**Note:** "LICENSE" on page 207 provides more information about the LICENSE command.

- MAINTenance *parameters*

  Validates maintenance level for the TSO user, with the following optional parameters:

  - SUMmary — Show cumulative maintenance level only
  - COMposite — Display composite load module report (default)
  - DETail — Display CSECT detail report
  - DeBuG — Enable debugging
  - LoadMOD=*module* — Display CSECT detail report for the specified load module

**Note:** "MAINTENANCE" on page 211 provides more information about the MAINTENANCE command.

- MSC *parameters*

  Validates MSC configuration, with the following optional parameters:

  - DeBuG — Enable debugging

- SubSYS *parameters*

  Displays GDDR subsystems, with the following optional parameters:

  - ACTive — Show only active GDDR subsystems
  - DeBuG — Enable debugging

  **Note:** "SUBSYS" on page 235 provides more information about the SUBSYS command.

- SUMmary *parameters*

  Displays summary information for the current system, with the following optional parameters:

  - DeBuG — Enable debugging

  **Note:** "SUMMARY" on page 237 provides more information about the SUMMARY command.

- SYStems *parameters*

  Validates systems, with the following optional parameters:

  - DeBuG — Enable debugging
  - SITE=*site* — Limit scope to systems at the specified site
  - SYStem=*sys* — Limit scope to the specified system

  **Note:** "SYSTEMS" on page 240 provides more information about the SYSTEMS command.

- TOPOlogy *parameters*

  Displays storage systems and SRDF groups, with the following optional parameters:

  - DeBuG — Enable debugging
  - ALL — Show SRDF groups to storage systems that are not GDDR-managed
  - SITE=*site* — Limit scope to storage systems at the specified site
  - CTRL=*serial#* — Limit scope to specified storage system and its SRDF groups

  **Note:** "TOPOLOGY" on page 243 provides more information about the TOPOLOGY command.

- DLM SET|RESET

  Sets or resets the DLM Abort flag.

- REFresh

  Refreshes the System Communication Status table.

- SORT ASC|DESC *fieldname*

  Performs an ascending or descending sort, where *fieldname* is either SITE, SYSTEM, TYPE, GDDRMAIN, HMC, DATE, TIME, or EVENTS.

---

**Note:** The following commands can be executed from the Perform Health Check panel but are not listed explicitly on the panel. See the panel help for information about the purpose and parameters of these commands: CHECKUP, ENQ, ParmREFresh, REGION, SET, SubSYS, SVCDUMP.

---

- Active Events

  Indicates any exceptions to conditions monitored by the GDDR Event Monitor.

  "Monitored events" on page 63 lists the events monitored by the GDDR Event Monitor.

- Expected Events

  Lists expected events.

  "Expected events" on page 69 discusses expected events.

- Degraded Mode

  Indicates whether Degraded mode is set (Yes) or not (No).

  "Degraded mode" on page 70 explains Degraded mode.

- Consistency

  Shows if there are any discrepancies between GDDR systems in regard to the GDDRPARM file or GDDR subsystem name.

  - Consistent: All GDDR systems are using the same GDDRPARM file and the same GDDR subsystem name.
  - Inconsistent: The GDDRPARM file in use or the GDDR subsystem name is not consistent for all GDDR systems. Inconsistency will cause Degraded mode to be set.

- Star-HA

  Indicates the status of the SRDF/Star High Availability (HA) feature. Possible values are as follows:

  - 0

    The configuration does not support SRDF/Star HA operation or there is no secondary MSC server available.

  - 1

    GDDR is configured for SRDF/Star HA support and currently there is a secondary MSC server available to take over as primary, should the current primary MSC server fail. This is the desired state for users of the SRDF/Star HA feature.

    ---

    **Note:** You configure GDDR for SRDF/Star HA support using the Define GDDR Configuration features panel (M,P,C,F).

    ---

- 2

  The primary MSC server has failed and there was a secondary MSC server available at the time of failure. If the GDDR SRDF/Star HA level was specified as 2 in the Define GDDR Configuration features panel (M,P,C,F), this failure triggers the Unplanned Star-HA Takeover script (GDDRUP35).

- SQAR-HA

  Indicates the status of the SRDF/SQAR High Availability (HA) feature. Possible values are as follows:

  - 2

    The primary MSC server has failed and there was a secondary MSC server available at the time of failure. If the GDDR SRDF/SQAR HA level was specified as 2 on the Define GDDR Configuration features panel (M,P,C,F), this failure triggers the Unplanned Star-HA Takeover script (GDDRUP35).

- DLm Abort

  Indicates the value of the DLM Abort global variable. Possible values are as follows:

  - 0

    All script DLm integration steps will be processed.

  - 1

    DLm support has been aborted in a prior script, subsequent script processing will bypass DLm integration steps. Resolve the managed virtual tape device issue that caused the abort, and use the DLM RESET command to change the value to '0' as described below.

  The DLM SET and DLM RESET commands, entered at the panel's command line, change the DLm Abort global variable to 1, DLM SET, or to 0, DLM RESET.

- Command Queue

  Indicates the current GDDR command queue status and size. GDDR uses an internal command queue to perform operations.

- System Communication Status

  - Line Commands

    - **S** (Select) queries the system and replaces the initial status value of 'Unknown' with one of three values: Active, Inactive, or Degraded.
    - **R** (Refresh) reruns the status query. You can get updated status for all systems by entering the command REFRESH at the command prompt.
    - **D** (Detail) displays the GDDRMAIN System Details panel (Figure 99 on page 373) containing information about the state of GDDRMAIN subtasks, dependent address spaces, and worker tasks.

  - Home Site

    The Site ID of the site where the system normally runs.

- System Name

  The System names are the C-systems and managed systems defined in the GDDRPARM file.

- Type
  - CSYS indicates a C-system.
  - PSYS indicates a GDDR-managed production or test system.
  - CF indicates this is a system assigned to a coupling facility LPAR. The S and R line commands, as well as the REFRESH primary command will not attempt to verify GDDRMAIN status for such a system, but will check for HMC console communication connectivity errors.

- GDDRMAIN Status

  GDDRMAIN communicates between C-systems and managed systems. This field shows the communication status. Initially, all systems except the C-system where you are, will show a status of "Unknown". Status values are as follows:

  - Active

    Indicates that GDDRMAIN is up and fully functional.

  - Inactive

    Indicates that GDDRMAIN on the system where this is running is unable to communicate with GDDRMAIN on the target system.

  - Degraded

    Indicates that GDDRMAIN is up on the target system, but not all subtasks or workers are active. The conditions considered when Degraded Status is set are displayed in the GDDRMAIN System Details panel as shown in Figure 99 on page 373. You can display this panel by typing **D** (Detail) in the Sel field associated with a system.

    Degraded status reflected by one or more inactive subtasks may be a transitory state, which is not a cause for concern. For example, the parameter Activate process requires that the GDDR Event Monitor and GDDR Heartbeat Monitor be stopped. If the Activate option of 'FULL' is selected, then GDDRMAIN will stop the GDDR Event Monitor and GDDR Heartbeat Monitor on all C-systems, causing Degraded status to be reported for each C-system.

  - N/A

    Indicates GDDRMAIN status is not applicable; shown for CF systems. There is no GDDRMAIN task installed on CF LPAR systems.

- HMC Status

  GDDR uses the z/OS Hardware Management Console (HMC) to maintain and view its managed systems, issue operator commands, and to discover HMC objects. The HMC Status field indicates the HMC communication status as follows:

  - Operating

    Indicates HMC functions are supported.

  - Error

    Indicates the HMC communication query failed.

  - Bypass

    Indicates this system was assigned HMC BYPASS status on the Define Managed LPARs panel (M,P,H,L) during GDDR configuration. No communication is allowed to the HMC console for this system. The S and R line commands, as well as the REFRESH primary command, will not attempt communication with the HMC for HMC Bypass systems. However, GDDRMAIN status will be checked by these commands on HMC Bypass systems.

  - N/A

    Displays for the C-system where you are.

- Last Communicated Date and Time

  The meaning of the date and time shown vary depending on the value of GDDRMAIN status.

  - If status is Active or Degraded, the date and time reflect the last time that health check queried that system and received a response.
  - If the status is Unknown or Inactive, the date and time specify the last time any communication was received.

- System Events

  Lists active system events, both planned and unplanned.

## Viewing system details

Type **D** for an entry in the System Communication Status table available in the Perform Health Check panel (C) and press **Enter** to open the GDDRMAIN System Details panel shown in Figure 99.

The GDDRMAIN System Details panel provides GDDRMAIN status information which may identify conditions causing a Degraded Mode of 'Yes' on the Perform Health Check panel (C).

```
+-------------------------------------------------------------------------------+
| -------------------- GDDR - GDDRMAIN System Details -------------------      |
| Command ===>                                                                  |
|                                                                               |
|  Site: DC1                 System:      LB04M41                              |
|  Type: CSYS                Status Date: 08/09/18 16:41:57                    |
|  IP:   nnn.nnn.nnn.nnn     HMC Status:  n/a                                  |
|                                                                               |
|  GDDRMAIN Status: Active                                                      |
|   CONSOLE: Y COMM: Y MISC: Y MCSOPER: Y WORKMGR: Y GVT: Y EVM: Y HBM: Y       |
|   Workers: Worker  Description         Min Max Act                            |
|            GDDWCM  Console message       1   2   1                           |
|            GDDWCO  Command               1   5   1                           |
|            GDDWCX  User msg automation   0   0   0                           |
|            GDDWDV  Device                1   25  1                           |
|            GDDWGV  Global variable        1   5   2                          |
|            GDDWST  Status                2   7   5                           |
|            GDDWXH  HMC                   1   5   1                           |
|            GDDWXQ  Command queue         1   1   1                           |
|            GDDWXR  REXX                  1   5   1                           |
|                                                                               |
+-------------------------------------------------------------------------------+
```

**Figure 99** GDDRMAIN System Details panel

The following fields are populated from the Perform Health Check panel (C):

- Site
- System
- Type
- Status Date
- GDDRMAIN Status
- HMC Status

The IP address field is populated from the COMM parameters defined in the GDDRPARM file.

The next lines provide the status of GDDRMAIN subtasks and worker tasks. GDDRMAIN subtasks are either shown as active with 'Y' or inactive with 'N'. For each worker type, the number of active tasks, as well as the minimum and maximum limits, are shown.

**Note:** "Worker task management" on page 165 discusses GDDR workers.

Managed systems will show subtasks GVT, EVM, and HBM with 'N' and workers GDDWGV and GDDWXR with '0' since these subtasks and workers are not applicable to managed systems.

The GDDRMAIN System Details panel shows all subtasks for C-systems as 'Y', except when a parameter activation is performed. GDDRMAIN stops EVM and HBM when a parameter activation is performed. Further, when the Activate operands 'FULL' and Propagate 'Y' are selected, HBM and EVM will be stopped on all C-systems. During this time, the Perform Health Check panel (C) and the GDDRMAIN System Details

panel show GDDRMAIN Status as Degraded. A GDDRMAIN Status of Degraded on C-systems also triggers the GDDR Event monitoring mode of Degraded, as shown on the Perform Health Check panel (C) under the Active Events field.

# Run GDDR scripts (S)

When you specify option **S** in the Primary Options Menu panel (Figure 34 on page 281), the Select Script to Run panels are displayed, similar to the following:

```
---------- GDDR - Select Script to Run on GDDRPLEX ---------------------------
 Command ===>                                                Scroll ===> CSR
 Enter S next to script to run and press <Enter>     Current Master:   SYS2
 Press <F3> to return to previous menu               Primary Site:     DC1
 Press <F1> to view all available commands           Primary DASD:     DC1
 Configuration: 4 SITES, SRDF/S SRDF/A AUTOSWAP CASC Automation:       ON

                                                    >
 Sel ---------------------- Scripts ---------------------- ----- Status ------
     =========== Scripts for  PLANNED  Actions ============
  _  Automated Configuration Check - DASD......... GDDRPCCD

  _  Swap production from DC1 to DC2.............. GDDRPA42 *SRA is ON

  _  Swap DASD from DC1 to DC2.................... GDDRPA21 *SRA is ON

  _  Abandon Region RG1 (site swap)............... GDD2P17A *SRA is ON

  _  Abandon Tertiary site (DC3).................. GDDRPA60 *SRA is ON

  _  Abandon Quaternary site (DC4)................ GDDRPA60 *SRA is ON
     =========== Scripts for   TEST    Actions ============
  _  Perform test IPL from BCVs at DC3............ GDD2P01A

  _  Perform test IPL from BCVs at DC4............ GDD2P01A

  _  Perform test IPL from R2s at DC2............. GDD2P03A

  _  Perform test IPL from R2s at DC3............. GDDRPA27

  _  Perform test IPL from R2s at DC4............. GDDRPA27

  _  Resume SRDF/A after test IPL at DC3.......... GDDRPA28

  _  Resume SRDF/A after test IPL at DC4.......... GDDRPA28
     =========== Scripts for UNPLANNED Actions ===========
  _  Recover after unplanned swap................. GDDRUP31 *CAX is OFF

  _  Recover after loss of DC1 (LDR).............. GDD2U13A *Requires LDR event
     =========== Scripts for  RESUME   Actions ============
  _  Resume SRDF/S replication after ConGroup trip GDDRPA23

  _  Resume SRDF/A in MSC mode to DC3............. GDDRPM29 *SRA is OFF

  _  Resume SRDF/A in MSC mode to DC4............. GDDRPM29 *Requires LDR event

  _  Resume SRDF/A in SQAR mode................... GDDRPF29 *CGT is ON

  _  Reclaim Secondary site (DC2)................. GDDRPA65

  _  Reclaim Tertiary site (DC3).................. GDDRPA65

  _  Reclaim Quaternary site (DC4)................ GDDRPA65
     =========== Scripts for  SPECIAL  Actions ============
  _  Planned STAR-HA Takeover..................... GDDRPA36

  _  Unplanned STAR-HA Takeover................... GDDRUP35 *STAR-HA state inv.

  _  Restart Secondary MSC Server at DC2   ....... GDDRPA36

  _  Restart primary MSC server at DC1............ GDDRMMSC *STAR-HA state inv.

  _  Restart primary MSC server at DC2............ GDDRMMSC *SRA is ON

  _  Transfer AutoSwap Owner to DC1............... GDDRPXAS

  _  Transfer Master C System to DC1.............. GDDRPXMC

  _  Transfer Master C System to DC3.............. GDDRPXMC

  _  Transfer Master C System to DC4.............. GDDRPXMC

     Global Variable Backup...................... GDDRPGVB
 ****************************** Bottom of data ********************************
```

**Figure 100** Select Script to Run panel (S)

The Select Script to Run panel (S) is a dynamic display. It lists GDDR scripts that are eligible to run based on the current state of the configuration, while providing script generation status messages about events which affect script eligibility. The panel also specifies configuration information and scripts that are in progress, if any. Depending on the location of the primary site and primary DASD locations, the panel is customized to display valid choices. Scripts that you cannot use at your current site do not appear.

Eligible scripts are grouped in the panel by category, such as “Scrips for PLANNED Actions” or “Scripts for SPECIAL Actions”.

See “Running GDDR Scripts” on page 517 for information about each category and the scripts it contains.

In the Select Script to Run panel (S), the DC1, DC2, DC3, and DC4 site names may reflect user-specified names for the sites as defined in the Define GDDR User Labels panel (M,P,O,L).

When the master C-system has been transferred to site DC3 or DC4, following a primary region loss, or a planned region swap by use of the Abandon Region *RG1* script, only resumption, recovery and special operations scripts are eligible for use.

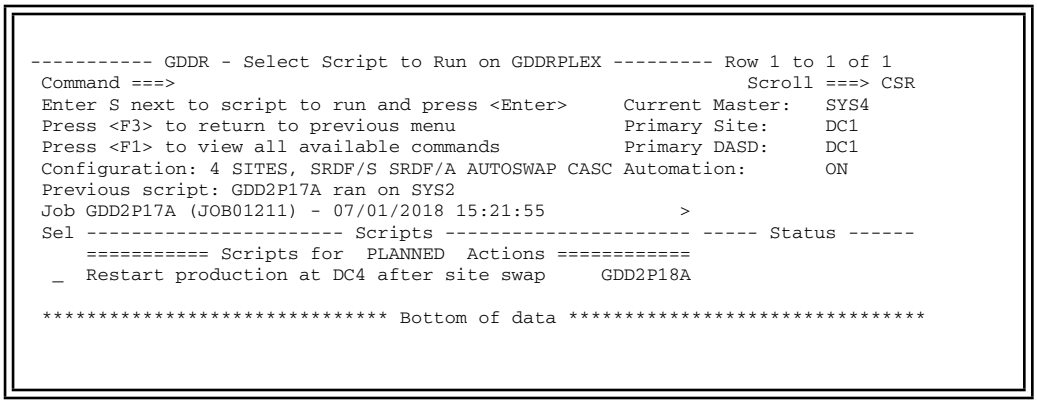



**Figure 101** Select Script to Run panel (S), master C-system at Secondary Region, 1 of 3

The Restart production at DC4 after site swap GDD2P18A script is the first script that is run at the secondary region following Abandon Region RG1 (site swap) GDD2P17A script.

Following completion of GDD2P18A, the Select Script to Run panel (S) is populated with the eligible Resumption, Regional Disaster Recovery and Special scripts, as shown in Figure 102 and Figure 103 below.

```
------------ GDDR - Select Script to Run on GDDRPLEX ------- Row 1 to 11 of 18
Command ===>                                                  Scroll ===> CSR
Enter S next to script to run and press <Enter>     Current Master:   SYS4
Press <F3> to return to previous menu               Primary Site:     DC3
Press <F1> to view all available commands           Primary DASD:     DC3
Configuration: 4 SITES, SRDF/S SRDF/A AUTOSWAP CASC Automation:       ON
Previous script: GDD2P18A ran on SYS4
Job GDD2P18A (JOB01215) - 07/01/2018 16:21:55            >
Sel ----------------------- Scripts ---------------------- ----- Status ------
    ============ Scripts for  RESUME   Actions ===========
 _  Resume SRDF/S replication after ConGroup trip  GDDRPA23

 _  Resume SRDF/A in MSC mode to DC3..    .......  GDDRPM29 *Requires LDR event

 _  Resume SRDF/A in MSC mode to DC4   ..........  GDDRPM29 *LDR is OFF @ csds
    ============ Scripts for   RDR     Actions ===========
 _  Recover at DC4 after RDR in primary region     GDDRPA05

 _  Recover at DC3 after RDR in primary region     GDDRPA05

 _  Recover at DC4 with SRDF/S in secondary region GDDRPA05 *RDF is ON

 _  Recover at DC3 with SRDF/S in secondary region GDDRPA05 *RDF is ON

 _  Restart production at DC4 with concurrent SRDF GDDRPA06

 _  Restart production at DC3 with concurrent SRDF GDDRPA06

 _  Restart production at DC4 with cascaded SRDF   GDDRPA06

 _  Restart production at DC3 with cascaded SRDF   GDDRPA06
```

**Figure 102** Select Script to Run panel (S), master C-system at Secondary Region, 2 of 3

```
------------ GDDR - Select Script to Run on GDDRPLEX ------ Row 12 to 18 of 18
Command ===>                                                  Scroll ===> CSR
Enter S next to script to run and press <Enter>     Current Master:   SYS4
Press <F3> to return to previous menu               Primary Site:     DC3
Press <F1> to view all available commands           Primary DASD:     DC3
Configuration: 4 SITES, SRDF/S SRDF/A AUTOSWAP CASC Automation:       ON
Previous script: GDD2P18A ran on SYS4
Job GDD2P18A (JOB01215) - 07/01/2018 16:21:55            >
Sel ----------------------- Scripts ---------------------- ----- Status ------
 _  Recover at DC4 after LDR at DC2, cascaded      GDDRPA07 *Data precedence

 _  Recover at DC3 after LDR at DC2, concurrent    GDDRPA07 *Requires LDR @ DC1
    ============ Scripts for  SPECIAL  Actions ===========
 _  Transfer Master C System to DC2     ...... ... GDDRPXMC

 _  Transfer Master C System to DC1....    . ..... GDDRPXMC

 _  Transfer Master C System to DC3...    .. ..... GDDRPXMC

 _  Global Variable Backup.....................  GDDRPGVB
****************************** Bottom of data ********************************
```

**Figure 103** Select Script to Run panel (S), master C-system at Secondary Region, 3 of 3

The following fields are provided in the Select Script to Run panel (S):

- Configuration

  This field indicates the features of your configuration that GDDR takes into account when determining which scripts are eligible to run and hence which are present in the list of scripts displayed on your panel.

- Scripts

  For a header row, the Scripts field contains a description of the category of scripts which follow. For a script row, this field contains a description of the script.

- Status

  This field displays script generation status events and dependencies. Each time you select a script to be submitted from the Select Script to Run panel (S), the script prerequisites are compared to the state of the environment. In cases when conflicts are identified or event prerequisites have not been satisfied, the messages listed in Table 36 on page 378 are returned to the Status field.

  If a Hide command has been issued, unavailable scripts with associated status descriptions do not appear.

  **Note:** See Table 1, "Monitored events," on page 63 for all event descriptions and related actions.

**Table 36** Script generation status messages (1 of 2)

| Message | Description |
|---|---|
| CAX > 0 | Not valid when a Swap has begun |
| CAX and CGT are ON | Not allowed when there is a CAX and a CGT event |
| CAX is OFF | Requires a CAX event |
| CFG event is on | Not allowed when there is a CFG event |
| CG Verify running | The GDDR Event Monitor has started a ConGroup Verify process |
| CGD is ON | Not allowed when there is a CGD event |
| CGT is OFF | Requires a CGT event |
| CGT is ON | Not allowed when there is a CGT event |
| Data precedence | Not allowed because the site with the most recent data is not appropriate |
| DC3 is down | Not allowed when DC3 is down |
| Degraded mode ON | GDDRMAIN Degraded mode is set. See the System Details panel in Option C: Checkup Perform Pre-script checkup for assistance. |
| DLM is OFF | Requires a DLM event |
| ECA is ON | Not allowed when there is an ECA event |
| GDDRPF29 can run | GDDRPM29 is not allowed to run |
| Invalid for state | Not valid for current state |
| Last unpl.scr. NOK | Can only run after a specific previous script |

**Table 36** Script generation status messages (2 of 2)

| | |
|---|---|
| LDR is OFF | Requires the LDR event |
| LDR is OFF | Requires an LDR event |
| LDR is OFF @ csds | Requires an LDR event for the secondary site |
| LDR is on for DC1 | Not allowed if there is an LDR event for DC1 |
| LDR is on for DC2 | Not allowed if there is an LDR event for DC2 |
| MHB ON @ Sec. site | Missing Heartbeat Event is on at the secondary site |
| MSC is OFF | Requires an MSC event |
| MSC is ON | Not allowed when there is an MSC event |
| MSF is ON | Not valid if there is an MSF event |
| Needs prot. pref. | Requires user-specified protection preference |
| Needs RDF+CGT ON | Requires both an RDF and a CGT event |
| Not at $DC3 | Not valid at DC3 |
| Only after P17A | Valid only after prerequisite GDD2P17A is complete |
| Only at DC3 | Valid only at DC3 |
| Only on Pri. MSC | Valid only at the primary MSC server site |
| Only on Sec. MSC | Valid only at the secondary MSC server site |
| Only Pri.DASD site | Valid only at the primary site |
| Only Sec.DASD site | Valid only at the secondary site |
| Passed site prim. | Not allowed for the primary site |
| Plan. scr. running | A planned script is running |
| RDF is ON | Not allowed when there is an RDF event |
| RDR event is OFF | Requires an RDR event |
| Requires LDR @ DC1 | Requires an LDR event at DC1 |
| Requires LDR @ DC2 | Script requires an LDR event at DC2 |
| Requires LDR event | Requires the LDR event |
| Sec.DASD site only | Valid only at the secondary site |
| SRA is OFF | Only allowed if there is an SRA event |
| SRA is ON | Not allowed when there is an SRA event |
| Star-HA state inv. | The SRDF/Star with HA configuration is in an invalid state |
| STR event is on | Not allowed when there is an STR event |
| Unpl. scr. running | An unplanned script is running |
| Wrong SQARA/B leg | In the secondary region, GDDRPM29 is only allowed on the SRDF/A leg that was activated earlier by a GDDRPA06 or GDDRPA07 script |

## Controlling which scripts are displayed

Scripts that are not applicable for the assigned configuration are not displayed in the Scripts list. If a script is applicable for the assigned configuration, but not available due to a specific state not being met, a comment is posted next to the script in the Status section stating why the script is not available. For example, the following script cannot be run because of the Degraded mode:

```
Swap production from DC1 to DC2..... GDDRPA42 *Degraded mode ON    .
```

"Table 36 Script generation status messages" on page 378 lists status messages that may appear. When entering the Select Script to Run panel (S), GDDR always shows all scripts applicable for the assigned configuration. You can control the scripts that appear in the panel by issuing commands at the prompt. To hide scripts that cannot be run because a required state has not been met, type **Hide** at the command prompt. Only scripts that can currently be run are displayed.

To revert to the default display and show all scripts in the panel, type **Unhide** at the command prompt or leave and re-enter the panel.

## Running scripts

Before running a script, perform additional pre-script checks listed in "Pre-script environment checks" on page 520.

To request that a script be run or rerun, type **S** next to the row describing the script you want to run and press **Enter**.

## Viewing script details

Type **D** next to the row with the script name and press **Enter** to open the Script Details panel similar to the following:

```
+-------------------- Script Details ---------------------+
| Command ===>                                            |
| Script: Global Variable Backup.....                     |
|    Program : GDDRPGVB                                   |
|    Skeleton: GDDRXACT                                   |
|                                                         |
|                                                         |
|                                                         |
+---------------------------------------------------------+
```

**Figure 104** Script Details panel

The Script Details panel shows the program name for the script and the name of the script skeleton. Press **F3** to return to the Select Script to Run panel (S).

# View GDDR script statistics (T)

When you specify option **T** in the Primary Options Menu shown in Figure 34 on page 281, the View GDDR Script Statistics panel (shown in Figure 105 on page 381) is displayed if no script is currently running.

If a script is running, option **T** of the Primary Options Menu displays the View Script Step Statistics panel (shown in Figure 106 on page 381) for the currently running script

```
------------------------- View GDDR Script Statistics ----------------------------
Option ===>

 Enter S next to a script to select it for viewing
 Press <F3> to return to the GDDR Primary Option Menu
 Last planned:   (none)
 Last unplanned: (none)

    Script
 -  ------------------------------------------------------------
 _  GDDRPA42 - Swap production from DC1 to DC2
 _  GDDRPA21 - Swap DASD from DC1 to DC2
 _  GDDRPA29 - Restart SRDF/A in MSC mode to DC3
 _  GDDRPA29 - Resume SRDF/A (SRDF/Star AutoSwap) to DC3
 _  GDDRPA23 - Resume ConGroup
 ****************************** Bottom of data ********************************
```

**Figure 105** GDDR Operation - Script Selection for Status panel (T)

In the View GDDR Script Statistics panel, type **S** next to a script and press **Enter** to view its step execution statistics. The View Script Step Statistics panel is displayed similar to the following:

```
-------------------- GDD5 - View Script Step Statistics ------------------------
 Option ===>

 Script:  GDDRPA29: Resume SRDF/A (SRDF/Star Autoswap) to DC3
 Initial: 01/20/2018 at 09:06:29.96   Iterations: 3   Maximum return code: 32
 Last run started 01/20/2018 at 10:09:27.44 and still running

   Function  Subfunction                RC  Step start/end           Cnt
   --------  -------------------------  --  ----------------------  ---
   GDDRDY01  RUN_ADCOPY_DC1_DYN__1       0           at

   GDDRDY01  RUN_RDFSUS_DC1_DYN__1       0           at

   GDDRDY01  RUN_READY_DC1_DYN__1        0           at

   GDDRDY01  RUN_RNGREF_DC1_DYN__1       0           at

   GDDRDY01  RUN_RNGRES_DC1_DYN__1       0           at

   GDDRGF3J  ADCOPY_DC1_CMD_1           GL           at

   GDDRGF3J  RDFSUS_DC1_CMD_1           GL           at
```

**Figure 106** View Script Step Statistics panel

The Step start/end field indicate the step completion date and time. If the step has not been executed, this field is empty.

# Perform GDDR actions (A)

Specify option **A** in the Primary Options Menu panel (Figure 34 on page 281) to access the GDDR Actions Menu:

```
--------------------------- GDDR - Actions Menu -------------------------------
 Option ===>

     H    Perform HMC Discovery                            This System: LB01M34
     L    Perform HMC LPAR actions                           This Site: DC2
     CBU  Perform HMC CBU actions                             Master-C: LB01M3A
     S    Manage Couple Datasets                           Primary Site: DC1
     CF   Manage CF Structures                             Primary DASD: DC1

                                                            Automation: ON
                                                       Planned script: None
                                                     Unplanned script: None




 Dell EMC Geographically Dispersed Disaster Restart  V5.2.0 BASE (08/13/18)
 © 2007-2018 Dell Inc. or its subsidiaries. All rights reserved.

 Select an option and press <Enter>
 Press <F3> to return to the GDDR Primary Options Menu
```

**Figure 107** Actions Menu panel (A)

## Perform HMC discovery (A,H)

To discover HMC objects accessible to GDDR, specify option **H** in the Actions Menu panel (A). A pop-up dialog such as the following is displayed and HMC object discovery is activated:

```
+-------- Discovering HMC Objects --------+
|                                          |
|   Discovering HMC objects at site DC1    |
|                                          |
|          ***  PLEASE WAIT  ***           |
|                                          |
|                                          |
+------------------------------------------+
```

**Figure 108** Discovering HMC Objects panel

When the discovery operation completes, the HMC Discovery Results panel containing a scrollable display such as the following is displayed:

```
------------------------ GDDR - HMC Discovery Results ---- Row 1 to 20 of 104
 Command ===>
 Enter <F3> to return to the previous menu
 ------------------------------------------------------------------------------
 GDDR STARTING DISCOVER HMC OBJECTS
 GDDR HMC COMMUNICATION WITH DC1 USING BCPII
 GDDR MANAGING THE FOLLOWING LPAR-SYSTEM PAIRS FOR SITE DC1
 >    IBM390PS.Q3,ZOSEQ311:Q311
 >    IBM390PS.C,ZOSEC11:T101
 >
 >   GDDR DISCOVERY INFORMATION FOR CPC       : IBM390PS.Q3
 >
 >   GDDR CURRENT ACTIVATION PROFILE          : DEFAULT
 >   GDDR LAST USED ACTIVATION PROFILE        : DEFAULT
 >   GDDR CPU MODEL                           : E10
 >   GDDR CPU TYPE                            : 2098
 >   GDDR CPU SERIAL NUMBER                   : 0000200F3FA4
 >   GDDR # GENERAL PURPOSE PROCESSORS        : 00000005
 >   GDDR # SERVICE ASSIST PROCESSORS         : 00000002
 >   GDDR # INTEGRATED FACILITY FOR APPLS     : 00000000
 >   GDDR # INTEGRATED FACILITY FOR LINUX     : 00000000
 >   GDDR # INTEGRATED COUPLING FACILITY      : 00000000
 >   GDDR # INTEGRATED INFORMATION PROCESSORS : 00000000
 >
```

**Figure 109** HMC Discovery Results panel (A,H)

When you are finished examining the results of the discovery operation, press **F3** to return to the Actions Menu panel (A).

## Perform HMC LPAR actions (A,L)

HMC actions are console operations affecting a processor, such as Load (IPL) or System Reset, or operations affecting an LPAR, such as Activate LPAR. When you select Option L - Perform HMC LPAR Actions in the Actions Menu panel (A), the following panel is displayed:

```
----------------------- GDDR - Perform HMC LPAR Actions ------ Row 1 to 6 of 6
  Command ===>                                                Scroll ===> CSR

  Actions: A Activate LPAR     D Deactivate LPAR       Current Master:   GA2LB148
           L Load Clear        R Reset Clear           Primary Site:     DC1
           X Load Recovery     W CPC Swap              Primary DASD:     DC1
           S Start LPAR        P Stop LPAR             Automation:       ON
           Q Query LPAR        E Show events           Planned script:   None
           H Query HMC Method  T Toggle Desired State  Unplanned script: None

  Type action to perform next to the desired system and site and press <Enter>
  When finished, press <F3> to return to the GDDR Actions Menu

  S
  E          LPAR                                D Load  Load
  L System   Site  CPC                 LPAR     S Addr  Parm      T  Message
  - -------- ----- ----------------- -------- - ----- -------- --- --------------
  _ GA2LB148 DC1   IBM390PS.M2964    M38      U ****  ******** M
  _ GA2LB34  DC1   IBM390PS.K12964   K134     D 02997 0F13KGM1 RES
  _ GA2LB34  DC1   IBM390PS.K12964   K134     D 02997 0F13KGM1 RBG
  _ GA2LB34  DC1   IBM390PS.K12964   K134     D 02997 0F13KGM1 RBT
  _ GA2LB29  DC3   IBM390PS.K12964   K139     U 9777  0F13KGM1 C
  _ GA2LB34  DC3   IBM390PS.K12964   K134     D ****  ******** RES
  ****************************** Bottom of data ********************************
```

**Figure 110** Perform HMC LPAR Actions panel (A,L)

The Perform HMC LPAR Actions panel (A,L) is only accessible on the master C-system. It should be viewed as a list of LPARs defined to GDDR, with one or more rows per LPAR, one for each type of IPL parameter defined for the system currently associated with the LPAR.

Most of the actions on this panel are meaningless if the listed system cannot be IPLed by GDDR based on the available IPL parameters for the listed system and the current state of the configuration. In that case, the relevant row will be shown with IPL parameters masked with "*". For example, if a system only has an STD IPL parameter defined for DC2, while the current Primary DASD site is DC1, this row will be shown with IPL parameters masked with "*". The R2 devices at DC2 are considered to be 'NR' in this case, and this not usable to IPL a system from.The system listed on that row can be queried (Option Q - Query LPAR) or events may be listed (Option E - Show Events).

You may enter an action code on any row for a system-site combination. The action you enter will be taken against the system or the system's LPAR using the site named on the selected row.

---

**Note:** All actions on this panel can also be performed using the GDDRHMCA utility described in "GDDR HMC Actions utility (GDDRHMCA)" on page 503.

---

The values displayed in the Perform HMC LPAR Actions panel (A,L) are defined in the Define Managed Systems panel (M,P,H,S) and the Define IPL Parameters panel (M,P,H,I). The following informational fields are conditionally displayed in the Perform HMC LPAR Actions panel (A,L) for GDDR users who are authorized to perform HMC actions:

- Sel

  Specify the action code for the action you want to perform for the system-site combination appearing on the row.

- System

  Indicates the name of the system associated with the LPAR shown on the table row.

- Site

  Identifies the site location of the LPAR shown on the row. This value displays as DC*n*-H for LPAR_RECOVERY protected systems if they are at their 'Home' location. This value displays as DC*n*-A for LPAR_RECOVERY protected systems if they are at their 'Away' location.

- CPC

  This is the name of the Central Processor Complex where the LPAR is defined.

- LPAR

  The name of the Logical Partition that this system runs in at the specified site.

- DS

  The desired state of a system, type **U** (up) or **D** (down).

Toggling the Desired State for a system on this panel Up or Down, has no effect on the system. This field only informs GDDR that the indicated system is intended to be down and that no action should be taken to bring it back up when GDDR observes the system is down.

If you use GDDR to bring a system down, either during scripts or by using this panel, GDDR sets the Desired State for the system to Down. Similarly, if you use GDDR to IPL a system, GDDR sets the Desired State for the system to Up.

- Load address

  Contains the MVS device address of the load device that will be used if you specify an action which performs a Load as part of its operation (for example, Load-Clear). The load address values are masked with asterisks for the master C-system, to prevent accidental IPL and for managed systems STP IPL parameters that are not currently usable to perform a LOAD action against the LPAR listed on the row. A STD IPL parameter is considered usable if it specifies a LOAD address on the primary DASD site, and the LPAR shown on the row is supposed to have channel connectivity to the primary DASD site.

- T

  Specifies the LPAR type, in some cases qualified with an IPL-parameter type.

  **C**: a C-system

  **M**: the master C-system

  **OS**: an HMC only LPAR, with RES IPL parameter

  **OA**: an HMC only LPAR, with ALT IPL parameter

  **RES**: a managed system, with RES IPL parameter

  **ALT**: a managed system, with ALT IPL parameter

  **DRT**: a managed system, with DR-TEST IPL parameter

  **RBG**: a managed system, with RES BCV Gold IPL parameter

  **ABG**: a managed system, with ALT BCV Gold IPL parameter

  **RBT**: a managed system, with RES BCV Test IPL parameter

  **ABT**: a managed system, with ALT BCV Test IPL parameter

  **CF**: a coupling facility LPAR

  **Note:** HMC actions LOAD and RESET CLEAR do not apply to CF LPARs.

- Load parms

  Contains the parameter used for the Load if you specify an action which performs a Load as part of its operation (for example, Load-Clear). The load parameter values are masked with asterisks if the Load address is masked with asterisks.

- Message

  Messages are active events for the listed system. Table 1, “Monitored events,” on page 63 provides a description of all events.

## Actions you can request

To request an HMC action, enter the action code on the command line and press **Enter**.

- **A**—Activate LPAR

  To activate the LPAR of the selected system, type **A** next to the selected system and press **Enter**. A confirmation pop-up displays:

```
+---------- Confirm Activate of LPAR 0013 ----------+
| Command ===>                                      |
|                                                   |
| Please confirm Activate request                   |
|                                                   |
| System: 0013                  Site DC1            |
| CPC:    IBM390PS.Q3           LPAR 0013           |
|                                                   |
| Proceed? ===> N  Reply Y / N / F / B / S          |
+---------------------------------------------------+
```

  - Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing.

    If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.

```
+----------------------- Confirm Activate of LPAR 0013 ------------------------+
| ------------------------- Specify Job card values ------------------------- |
| Command ===>                                                                |
|                                                                             |
|   Enter or change the job statement below:                                  |
|                                                                             |
|      //GDDRPA42 JOB (06610),GDDR-PARMUPDT,                                  |
|      //         CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID,REGION=0M                 |
|      //*                                                                    |
|      //*                                                                    |
|                                                                             |
|   Press <Enter> when ready to submit job                                    |
|   Press <F3> to exit without submitting job                                 |
+-----------------------------------------------------------------------------+
```

  - Cancel the request by entering **N**.
  - Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel similar to the following indicates where the JCL was saved:

```
+---------------------------------------------+
| JCL to run Deactivate of LPAR 0013 saved to |
| GDDR.GDDR520.WORK.PARMS(H0512294).          |
+---------------------------------------------+
```

  **Note:** The Activate action is not allowed for the master C-system or for an LPAR which is already activated.

- **D**—Deactivate LPAR

  To deactivate the LPAR the selected system runs in, type **D** next to the selected system and press **Enter**. A confirmation pop-up displays.

  - Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing. If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.
  - Cancel the request by entering **N**.
  - Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel indicates where the JCL was saved.

  **Note:** The Deactivate action is not allowed for the master C-system.

- **L**—Load Clear

  **Note:** Before taking this action, verify that the value of the Desired State field (DS) is consistent with the Load action. Taking this action will cause the Desired State for the system to become Up.

  To clear and load a selected system using the displayed load address and load parameters, type **L** next to the selected system and press **Enter**. A confirmation pop-up displays.

  The Load Address and Load Parm fields are populated from current IPL parameters, you may overtype these fields in the confirmation panel. If overtyped, the panel saves the new values by creating a new or overwriting an existing alternate IPL parameter, causing the Perform HMC LPAR Actions panel (A,L) to dynamically update, showing the new alternate IPL parameter entry.

  **Note:** The new or updated global variables representing these IPL parameters only exist on the master C-system. A Parameter Wizard Activate is required to broadcast these changes to the other C-systems. See "GDDR IPL Parameter Swap utility (GDDRISWP)" on page 492 for an alternative approach which includes keeping the remote C-systems up-to-date.

  - Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing. If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.
  - Cancel the request by entering **N**.
  - Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel indicates where the JCL was saved.

  The following rules apply to IPLs using STD GDDR-managed volumes:

  - By default, use the site selected for the IPL action.
  - For test-IPL scripts, always use the selected site.

  For other scripts and IPL actions:

  - For managed systems at the primary or secondary DASD site, use the current primary DASD site, unless an unplanned swap has completed and the recovery script has not run.
  - For C-systems, use the home site of the C-system, unless the system has LPAR Recovery protection and is in its 'Away' location.

  The Load Clear action is not allowed:

  - For the master C-system
  - For CF LPARs
  - If the Load address on the selected row is masked with ****
  - If the system is currently operating
  - If the LPAR is not activated

- **R**—Reset Clear

  **Note:** Before taking this action, verify that the value of the Desired State field (DS) is consistent with the Reset action. Taking this action will change the Desired State for this system to Down.

  To reset and clear a selected system, type **R** next to the selected system and press **Enter**. A confirmation pop-up displays. A confirmation pop-up displays.

  - Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing. If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.
  - Cancel the request by entering **N**.
  - Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel indicates where the JCL was saved.

  **Note:** The Activate action is not allowed for the master C-system or if the LPAR is not activated.

- **X**—Load Recovery

  **Note:** Before taking this action, verify that the value of the Desired State field (DS) is consistent with the Load Recovery action. Taking this action will change the Desired State for this system to Up.

  To IPL at the system's alternate location, type **X** next to the selected system and press **Enter**. If this option is available, the Site value will display a fourth character after DC*n*; either H or A**,** meaning the system is currently running at DC*n* in its home (H) location or its alternate (A) location. A confirmation pop-up displays.

  The Load Address and Load Parm fields are populated from current IPL parameters, you may overtype these fields in the confirmation panel. If overtyped, and if alternate IPL parameters do not exist for the system/site, the panel saves the new values as alternate IPL parameters, causing the Perform HMC LPAR Actions panel (A,L) to dynamically update, showing the new alternate IPL parameters entry.

  - Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing. If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.
  - Cancel the request by entering **N**.
  - Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel indicates where the JCL was saved.

  The following rules apply to IPLs using STD GDDR-managed volumes:

  - By default, use the site selected for the IPL action.
  - For test IPL scripts, always use the selected site.

  For other scripts and IPL actions:

- For managed systems at DC1 or DC2, use the current primary DASD site, unless an unplanned swap has completed and the recovery script has not run.
- For C-systems, use the home site of the C-system, unless the system has LPAR Recovery protection and is in its 'Away' location.

---

**Note:** The Load Recovery action is not allowed for the master C-system or for a system that does not have LPAR Recovery protection.

---

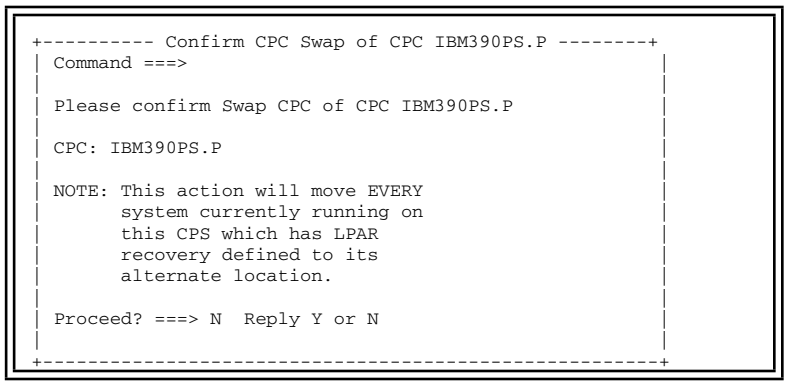
- **W**—CPC Swap

---

**Note:** The CPC Swap action applies to the entire CPC that houses the LPAR shown on the selected row. All Recovery Protected LPARs on that CPC will be moved to their alternate locations (from “Home” to “Away” or from “Away” to “Home”). This action is only allowed when the system on the selected row has LPAR Recovery Protection.

---

To perform a planned CPC swap, type **W** next to the selected system and press **Enter**. A confirmation pop-up displays:




- Reply **Y** or **N** to either continue or discontinue the requested action.

- **S**—Start LPAR

  To start a selected system, type **S** next to the selected system and press **Enter**. A confirmation pop-up displays.

  - Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing. If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.
  - Cancel the request by entering **N**.
  - Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel indicates where the JCL was saved.

  ---

  **Note:** The Start LPAR action is not allowed for C-systems. This action is only allowed if the selected LPAR is in an Activated, Not Operating state.

  ---

- **P**—Stop LPAR

  To stop a selected system, type **P** next to the selected system and press **Enter**. A confirmation pop-up displays.

- Confirm the request by entering **Y** or **F** for foreground processing or **B** for background batch processing. If you choose **B**, a second confirmation pop-up displays. You can modify the job statement as needed.
- Cancel the request by entering **N**.
- Save the batch execution JCL without processing by entering **S**. After pressing **Enter**, a confirmation panel indicates where the JCL was saved.

---

**Note:** The Stop LPAR action is not allowed for C-systems. This action is only allowed if the selected LPAR is in an Operating state.

---

- **Q**—Query LPAR

  To query the state of the LPAR of the selected system, type **Q** next to the selected system and press **Enter**. A confirmation pop-up will be displayed. Confirm the request by typing **Y** and pressing **Enter**, or cancel the request by pressing **F3**. When the query is performed, the Message field is updated with the current state of the selected LPAR: Not Activated, Not Operating, Operating.

- **E**—Show Events

  To show any relevant active events for the system in the Message field, type **E** next to the selected system and press **Enter**. System related events are IPL, (planned IPL), MXS, (unplanned MXS), and MHB, (unplanned MHB). Table 1, "Monitored events," on page 63 provides additional information about these events.

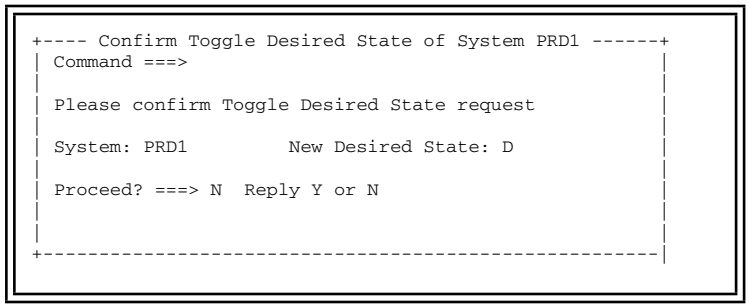
- **T**—Toggle desired state (U to D or D to U)

  The desired state of a system (U for up or D for down) is displayed in the DS (Desired State) column of the Perform HMC LPAR Action panel (A,L). The default value is U.

  Toggling the desired state up or down for a system on this panel has no effect on the system. This field only serves the purpose of informing GDDR that the indicated system is intended to be down and that no action should be taken to bring it back up when GDDR observes the system to go down.

  However, if you use GDDR to bring a system down, either during scripts or using this panel, GDDR will set the desired state for the system to Down. Similarly, if you use GDDR to IPL a system, GDDR will set the desired state for the system to Up.

  To toggle the desired state of a system, type **T** next to the selected system and press **Enter**. A confirmation pop-up with the new state displays:




  - Reply **Y** or **N** to either continue or discontinue the requested action.

# Perform HMC CBU actions (A,CBU)

When you specify option **CBU** in the Actions Menu panel (A), the following panel is displayed:

```
-------------------------- GDDR - Perform CBU Actions ------------- Row 1 of 6
 Command ===>                                                  Scroll ===> CSR

 Type one of the following action codes next to desired site/CPC pair
     A  activate CBU in REAL mode
     B  activate CBU in TEST mode
     C  Undo CBU in REAL mode
     D  Undo CBU in TEST mode
 When ready, press <Enter> to perform the action
 You may press <F3> to return to the Primary Options Menu

 Sel  Site  CPC       Message
 ---  ----  --------  ------------------------------------------------------
  _   DC1   X0
  _   DC1   X0
  _   DC1   X1
  _   DC3   X0
  _   DC3   X0
  _   DC3   X1
 ****************************** Bottom of data ********************************
```

**Figure 111** Perform CBU Actions panel (A,CBU)

Each row represents a site with a processor enabled for CBU (Capacity Backup Upgrade) activation.

Type one of the following action codes next to the desired site/CPC pair:

- **A** to activate backup capacity on the specified processor (REAL mode)
- **B** to simulate the activation of backup capacity on the specified processor (TEST mode)
- **C** to undo or terminate actual backup capacity activation
- **D** to undo or terminate a simulation of backup capacity activation

# Manage coupled datasets (A,S)

When you specify option **S** in the Actions Menu panel (A), the following panel is displayed:

```
------------------------ GDDR - Manage Couple Datasets ------------------------
Command ===>

  Current Primary DASD Site: DC1


  Specify site for   Primary Couple Datasets: DC1

  Specify site for Alternate Couple Datasets: DC1

  Simulation Mode ? NO    (YES/NO)

  Enter or change the job statement:
     //jobname  JOB account,name,CLASS=class,MSGCLASS=class,SYSAFF=*,
     //        USER=GDDR,REGION=0M
     /*JOBPARM LINES=999999
     //*

  Press <Enter> when ready.  Press <F3> to return without submitting job.
```

**Figure 112** Manage Couple Datasets panel (A,S)

You can use this panel to drive GDDR management of coupled datasets, aligning coupled dataset usage with the sites specified on this panel. GDDR coupled dataset management is driven regardless of script management options for each site specified in the Script Sysplex Options panel (M,P,O,S).

**Note:** See the criteria for selection of eligible systems for sysplex operations in "System qualification for sysplex options" on page 342.

In environments with geographically dispersed sysplexes supported by cross-site host-DASD channels, the sites can be primary (site A) or secondary (site B). In other environments, only the primary site (R1 site) is allowed.

Currently used coupled datasets are compared against those defined to GDDR for each site. If discrepancies are found, GDDRMAIN COMM services are used to issue the required SETXCF COUPLE commands and bring coupled dataset usage in line with GDDR parameter specifications. This requires GDDRMAIN to be up and running on the target managed systems.

The current primary DASD site is shown for information, in case you want to use primary coupled datasets from the primary DASD site, and alternate coupled datasets from the opposite site.

When operating in the secondary region after scripts GDDRPA05, GDDRPA06, or GDDRPA07 have run, the site shown here is the R1 location. This will be different from the current primary DASD site shown on other GDDR panels.

You can make the following changes using this panel:

- Specify site for Primary Couple Datasets

  Informs GDDR where you want to have primary coupled datasets used.

- Specify site for Alternate Couple Datasets

Informs GDDR where you want to have alternate coupled datasets used.

- Simulation mode? YES/NO

  Indicates whether to use simulation mode. The default setting is **NO**. Simulation mode issues real D XCF commands to determine the initial situation. All other commands will be shown in the job log but not issued on the target system.

  **Note:** Activating simulation mode on this panel affects this and all subsequent GDDR coupled dataset management actions (stand-alone or in scripts) on this C-system until the next GDDR Parameter Wizard activation. You can also activate coupled dataset management simulation using the Manage GDDR System Variables panel (M,S), by adding a new variable USEROPT.CPLSIMUL=1. This variable will be removed upon the next GDDR Parameter Wizard activation.

- Enter or change the job statement

  Provide the jobcard and any other JCL required.

Press **Enter** when ready. Press **F3** to return to the previous panel without submitting the job.

## Manage CF structures (A,CF)

When you specify option **CF** in the Actions Menu panel (A), the following panel is displayed:

```
------------------------ GDDR - Manage CF Structures   ------------------------
 Command ===>

   Current Primary DASD Site: DC1


   Specify site for CF Structure location: DC1

   Simulation Mode ? NO    (YES/NO)


   Enter or change the job statement:
      //GDDRSTR0 JOB (06610),GDDR-PARMUPDT,
      //         CLASS=A,MSGCLASS=X,SYSAFF=*,NOTIFY=&SYSUID,REGION=0M
      //*
      //*

   Press <Enter> when ready.  Press <F3> to return without submitting job.
```

**Figure 113** Manage CF Structures panel (A,CF)

This panel is used to control GDDR management of coupling facility structures, causing CF structures to be rebuilt in the next CF in the Preference List for each structure, as specified in the active CFRM policy. GDDR CF structure management is driven regardless of the script management options for each site specified in Script Sysplex Options panel (M,P,O,S).

**Note:** See the criteria for selection of eligible systems for sysplex operations in "System qualification for sysplex options" on page 342.

In environments with geographically dispersed sysplexes supported by cross-site host-DASD channels, you can request rebuild to primary site (site A) or the secondary site (site B). In other configurations, only the primary site (site A) is allowed.

Current CF structure locations are compared against those defined to GDDR for the site specified on this panel. If discrepancies are found, GDDRMAIN COMM services are used to issue the required SETXCF START,REBUILD commands on production systems.

The current primary DASD site is shown for information, in case you want to move CF structures to the primary DASD site.

When operating in the secondary region after RDR Recovery (scripts GDDRPA05, GDDRPA06 or GDDRPA07), the current primary DASD site shown here is the R1 location. This will be different from the current primary DASD site shown on other GDDR panels.

You can make the following changes using the Manage CF Structures panel (A,CF):

- Specify site for CF structures location

  Specify where you want to move CF structures.

- Simulation mode? YES / NO

  Specify whether to use simulation mode. The default is NO. Simulation mode issues real D XCF commands to determine the initial situation. All other commands will be shown in the job log but not issued on the target system.

  ---

  **Note:** Activating simulation mode on this panel affects this and all subsequent GDDR CF structure management actions, both stand-alone or in scripts, on this C-system until the next GDDR Parameter Wizard activation. You can also activate CF structure management simulation using the Manage GDDR System Variables panel (M,S), by adding a new variable USEROPT.CPLSIMUL=1. This variable will then be removed upon the next GDDR Parameter Wizard activation.

  ---

- Enter or change the job statement

  Specify the jobcard and any other JCL required.

Press **Enter** when ready. Press **F3** to return to the previous panel without submitting the job.

# Run Dell EMC Started Task Execution Manager (ST)

The Dell EMC Started Task Execution Manager allows you to start and stop GDDR and Mainframe Enablers related tasks from a single console.

Specify option **ST** in the GDDR Primary Options Menu panel (Figure 34 on page 281) to access the Dell EMC Started Task Execution Manager. The following panel is displayed:

```
---------------------- Dell EMC Started Task Execution Manager -------------------
Command ===>                                                  Scroll ===> PAGE
                                                          MCS Console ID: LHAIRS1
  _ ADD - "S" to add a new Procedure    _ LIST - "S" to List JOBPARM Members

Line Cmds: C = Clone D = Del E = Edit S = Edit Description R = RMT X = Execute
ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
============================ Procedure Member List ============================
   Sel Procedure Description
   --- --------- ------------------------------------------------------------
****************************** Bottom of data ********************************
```

**Figure 114** Dell EMC Started Task Execution Manager panel (ST)—Procedure Member List

A blank list is displayed if this is the first use of the panel; otherwise, any previously defined procedures display here.

## Adding a new procedure

A procedure is a named collection of tasks to be issued in a specific sequence.

1. Type **S** next to the word ADD and press **Enter**.

   A blank Procedure Add/Update panel is displayed:

```
---------------------- Dell EMC Started Task Execution Manager ----------------
Command ===>                                                  Scroll ===> PAGE


ENTER: Process Add / Update            PF3: Cancel Add / Update
============================ Procedure Add/Update ============================



  Procedure Name:
  Description...: Start up all my EC0 prefixed tasks_____________________________
```

**Figure 115** Dell EMC Started Task Execution Manager—Procedure Add/Update panel

2. Type in a name and description for the procedure and press **Enter**.

The procedure is added to the Procedure Member List panel. The following example shows a procedure named STRTECO:

```
---------------------- Dell EMC Started Task Execution      Member STRTECO Added
 Command ===>                                                 Scroll ===> PAGE
                                                          MCS Console ID: JPREST2
     _ ADD - "S" to add a new Procedure

Line Cmds: C = Clone D = Del E = Edit S = Edit Description R = RMT X = Execute
ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right


 ============================ Procedure Member List ============================
    Sel Procedure Description
    --- --------- ------------------------------------------------------------
     _  STRTECO   Start up all my ECO prefixed tasks
```

**Figure 116** EMC Started Task Execution Manager—Procedure Member List panel

## Adding tasks to the procedure

A task is a z/OS or 3rd party vendor command to be issued.

1. To add or update tasks associated with a procedure, type **E** next to the procedure you want to edit and press **Enter**.

   If there are already tasks for the procedure, they are listed on the Task List panel, otherwise the panel displays with a blank table:

```
---------------------- Dell EMC Started Task Execution Manager -----------------
 Command ===>                                                 Scroll ===> PAGE

     _ ADD  - "S" add new Task(s), or "C" to clone existing Procedure

 Line Commands...: E or S or / to Edit   C to Copy   D to Delete the entry(s)
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ================= STRTECO - Start up all my ECO prefixed tasks ================
   Sel SeqNbr Issue Command              String to Find             Skip Wait
   --- ------ ------------------------- -------------------------- ---- -----
 ******************************* Bottom of data ********************************




+-------------------------------------------------------------------------+
| There were zero (0) task records found for STRTECO, use the ADD primary |
| command to add tasks.                                                   |
+-------------------------------------------------------------------------+
```

**Figure 117** Dell EMC Started Task Execution Manager—blank task list

2. To add a new task, type **S** next to the word ADD and press **Enter**. A blank Task Add/Update panel displays:

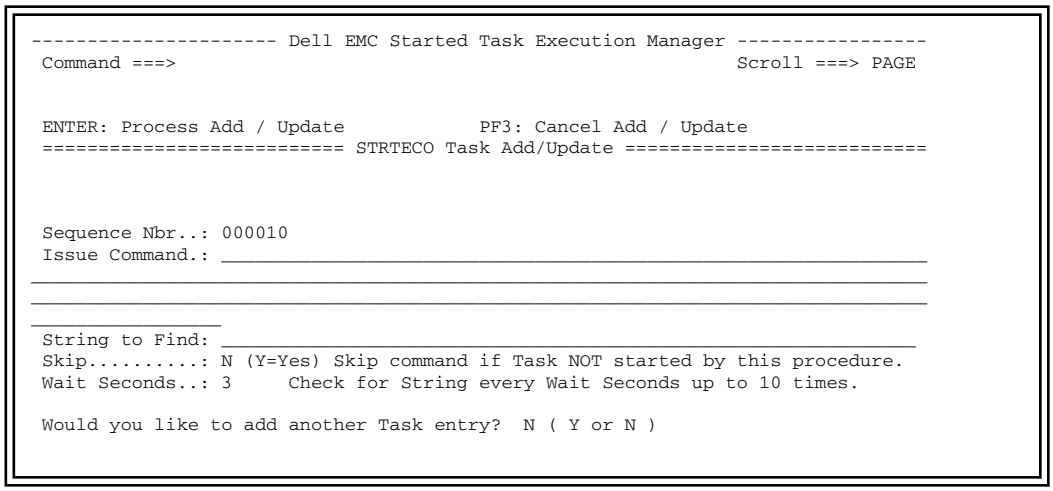



**Figure 118** Dell EMC Started Task Execution Manager—Task Add/Update panel, blank

3. Complete the panel fields as follows:

   - Sequence Nbr

     This number defines the order in which the tasks will be executed. You can accept the generated value or you can overtype this number with a number between two others to insert this command at its desired position. These sequence numbers are automatically regenerated as multiples of 10 each time the procedure is edited and saved to allow for future inserts.

   - Issue Command

     The command to be issued when the procedure is executed. Enter a valid z/OS, JES, operator, or third party vendor command.

     You can also use the ROUTE command to direct a command to one or more systems in a sysplex for processing. For example, if you are on X00B and X001 is an LPAR in the same sysplex, then enter the following in this field to execute a D IPLINFO command on X001: **RO X001,D IPLINFO**.

     **Note:** If the command string is longer than 26 characters, ISPF column scrolling is enabled to allow you to see the whole command on the panel.

   - String to Find

     A text string or a message number that must be found by the Dell EMC Started Task Execution Manager before the next task will be executed. Enter a text string or a message number. After waiting the specified period (see “Wait Seconds” below), up to 10 attempts will be made to find this string. If it is not found, an informational message displays and the Dell EMC Started Task Execution Manager exits. If the string is found, the complete string is echoed back, and execution continues.

**Note:** If the text string is longer than 26 characters, ISPF column scrolling is enabled to allow you to see the whole string on the panel.

- Skip

  Enter **Y** to direct the command to be skipped if the started task associated with this command is not started by this invocation of Dell EMC Started Task Execution Manager. If the started task associated to this command is already started, then this task is bypassed. Enter **N** to always execute the command.

- Wait Seconds

  Checks for the "String to Find" text string every *n* Wait Seconds up to 10 times. For example, if you want to wait no longer than a minute for the string to be found, enter a 6 into this field.

- Would you like to add another Task entry?

  Press **Y** to add the entry to the procedure and display a new blank Task Add/Update panel.

The following example shows a completed Task Add/Update panel:

```
- ---------------------- Dell EMC Started Task Execution Manager ---------------
 Command ===>                                                  Scroll ===> PAGE


 ENTER: Process Add / Update            PF3: Cancel Add / Update
 ============================ STRTECO Task Add/Update ============================



 Sequence Nbr..: 000010
 Issue Command.: S ECOSCF_____________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_________________
 String to Find: SCF0890I_____________________________________________________
 Skip..........: N (Y=Yes) Skip command if Task NOT started by this procedure.
 Wait Seconds..: 10    Check for String every Wait Seconds up to 10 times.

 Would you like to add another Task entry?  N ( Y or N )
```

**Figure 119** Dell EMC Started Task Execution Manager—Task Add/Update panel, completed

4. Press **Enter** when you are done adding or updating tasks.

   The Task List panel displays with the modified tasks. The following example shows the task has been added to the list for a procedure named STRCECO:

```
---------------------- Dell EMC Started Task Execution    Task 000010 Modified
 Command ===>                                               Scroll ===> PAGE

     _ ADD  - "S" add new Task(s), or "C" to clone existing Procedure

 Line Commands...: E or S or / to Edit   C to Copy   D to Delete the entry(s)
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ================= STRTECO - Start up all my ECO prefixed tasks ================
   Sel SeqNbr Issue Command              String to Find             Skip Wait
   --- ------ -------------------------- -------------------------- ---- -----
    _  000010 S ECOSCF                   SCF0890I                    N      10
 ****************************** Bottom of data ********************************
```

**Figure 120** Dell EMC Started Task Execution Manager—task list, completed

You can edit, copy, or delete tasks in the list using the specified line commands.

## Executing a procedure locally

Once you have entered all the tasks needed for a procedure, you can execute the procedure on the LPAR where you are running the Dell EMC Started Task Execution Manager.

1. Select the procedure by typing an **X** or a **W** line command.
   - **X** executes all tasks without stopping between each task.
   - **W** executes a single task. You can then press **Enter** to continue with the next task in the procedure or type **QUIT** to terminate the procedure. This method allows you to view the output of each individual command before proceeding.

   For example, in the following display, procedure JFPSCFD is selected for execution.

```
---------------- Dell EMC Started Task Execution Manager - Member STRTEC0 Added Command
===>                                                 Scroll ===> PAGE
             _ ADD (Select with an S to add new Procedures)

Line Cmds: C = Clone D = Del E = Edit S = Edit Description R = RMT X = Execute
ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right

================================================================================
   Sel Procedure Summary
   --- --------- ------------------------------------------------------------
    X  JFPSCFD   Startup for development SCF & SRDF
    _  JFPSTOP   Stop SCF & SRDF
****************************** Bottom of data ********************************
```

**Figure 121** Dell EMC Started Task Execution Manager—procedure JFPSCFD selected

2. Press **Enter**.

   Command output displays for the tasks in the procedure.

## Executing a procedure remotely

You can also execute Started Task Execution Manager (STEM) procedures on a remote host via TCP/IP.

### Prerequisites

Before running a procedure on a remote host, ensure the items listed below are complete.

- The STEM listener procedure JCL must be installed in the appropriate started task PROCLIB at the remote host. The JCL for the STEM listener procedure is as follows:

```
// SETPROG APF,ADD,DSN=EMC.STEM.LINKLIB,SMS
//*
//*  PRODECURE ESFSTEM
//*  STEM REMOTE EXECUTION LISTENER
//*
//STEMLSTN EXEC PGM=ESFSTEM,PARM='nnnn'
//STEPLIB  DD  DISP=SHR,DSN=EMC.STEM.LINKLIB
//SYSPROC  DD  DISP=SHR,DSN=EMC.STEM.REXX
//SYSEXEC  DD  DISP=SHR,DSN=EMC.STEM.REXX
//STEMTASK DD  DISP=SHR,DSN=EMC.STEM.JOBPARMS
//SYSUDUMP DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
```

  The PARM= *value* on the EXEC statement specifies the TCP/IP port that will be used on this host to accept STEM procedure execution requests.

  The user ID associated with the ESFSTEM started task must have an OMVS segment defined that allows read/write access to TCP/IP.

  You can start the ESFSTEM started task by issuing the S ESFSTEM console command. You can stop the task by issuing either the F ESFTEM,STOP command or the P ESFSTEM command.

  **Note:** The necessary LINKLIB, REXX and JOBPARMS PDS files will have been created at the time SCF was installed at the remote host.

- The STEM work task procedure JCL must be installed in the appropriate started task PROCLIB at the remote host. The JCL for the STEM work task procedure is as follows:

```
//ESFSTEMT PROC DATAHLQ=EMC.STEM.TEST,SCRIPT=A1234567,UID=USERID
//*
//*   PROCEDURE ESFSTEMT
//*   STEM TASK PROCEDURE WORK ADDRESS SPACE
//*
//ESFSTEMT EXEC PGM=ESFSTEMT,PARM=&UID,
//            REGION=256M,TIME=1440
//*
//STEPLIB  DD  DISP=SHR,DSN=EMC.STEM.LINKLIB
//SYSUEXEC DD  DISP=SHR,DSN=EMC.STEM.REXX
//SYSPROC  DD  DISP=SHR,DSN=EMC.STEM.REXX
//SYSEXEC  DD  DISP=SHR,DSN=EMC.STEM.REXX
//SYSTSPRT DD  DSN=&DATAHLQ..&SCRIPT..SYSTSPRT,
//             DISP=(,CATLG,CATLG),
//             DSORG=PS,RECFM=VBA,LRECL=137,
//             UNIT=SYSDA,SPACE=(CYL,(5,1))
```

```
//SYSTSIN  DD  DUMMY
//SYSOUT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SNAP     DD  DUMMY
//ABNLIGNR DD  DUMMY
//SYSUDUMP DD  SYSOUT=*
```

When a request to execute a STEM procedure is received over TCP/IP, ESFSTEM starts an ESFSTEMT address space to process the commands within the procedure. When the commands have been issued and the responses collected, the ESFSTEMT address space shuts down. The execution of STEM procedures is serialized, so there should never be more than one ESFSTEMT address space active at any point in time.

During initialization, the ESFSTEMT address space's ACEE is set to the user ID that sent the STEM procedure work request. This ensures you cannot issue any command through STEM that you would not have authority to issue via logging on to the host LPAR and entering the same command through SDSF.

- To execute a Dell EMC Started Task Execution Manager command on a remote host you must:
  - Have the same TSO user ID defined for the remote host and for the Dell EMC Started Task Execution Manager ISPF function on the local host.
  - Have COMMAND authority on the remote host.

## Executing the remote procedure

1. To execute a Dell EMC Started Task Execution Manager procedure on a remote host via TCP/IP, type **R** next to the procedure. For example:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 5 of 5
 Command ===>                                                  Scroll ===> PAGE
              _ ADD (Select with an S to add a new Procedure)

Line Cmds: C = Clone D = Del E = Edit S = Edit Description R = RMT X = Execute
ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right

 ============================ Procedure Member List ============================
    Sel Procedure Description
    --- --------- ------------------------------------------------------------
     _  ADDOWN    Shut down SRDF and SCF
     R  ADUP      Start up SCF and SRDF
     _  ADUP6     Start SCF SRDF HSM6X
     _  WTORTEST  Test WTOR reply feature
 ****************************** Bottom of data ******************************
```

**Figure 122** Dell EMC Started Task Execution Manager—procedure ADUP selected for remote execution

2. Press **Enter**.

A panel similar to the following is displayed:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 15 of 15  .
                                                                                     .
        _ ADD - "S" to add new Host                                                  .
                                                                                     .
Line Cmds..:  S = Select Host     E =Edit Host     D = Delete Host                   .
ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down    PF10: Left    PF8: Right     .
====================== Host List for procedure: DIPLINFO ======================== .
                                                    TCP/IP    Replace               .
       Sel  HostName  IP address or DNS Name        STC Name  Procedure             .
       ---  --------  ----------------------------  --------  ---------             .
        _   -ALL-                                             N                    .
        _   DC1CO012  nnn.nnn.nnn.nnn               TCPIP     Y                    .
        _   DC1PQ317  nnn.nnn.nnn.nnn               TCPIP     N                    .
        _   DC2CO017  nnn.nnn.nnn.nnn               TCPIP     N                    .
        _   DC2PQ31A  nnn.nnn.nnn.nnn               TCPIP     N                    .
        _   DC3CO01A  nnn.nnn.nnn.nnn               TCPIP     N                    .
```

**Figure 123** Dell EMC Started Task Execution Manager—Host List for procedure panel

3. Type **E** next to the host to edit and press **Enter**.

   A panel similar to the following is displayed:

```
---------------------- Dell EMC Started Task Execution Manager -----------------
 Command ===>                                                 Scroll ===> PAGE


 ENTER: Submit Remote Execution Request    PF3: Cancel Remote Execution
 ========================== ADUP Remote Execution ============================



 Remote Host...: nnn.nnn.nnn.nnn             Host Name or IP Address
 TCP/IP Port ..: 7836
 Replace Member: N ( Y or N ) Replace like named member at remote host.
```

**Figure 124** Dell EMC Started Task Execution Manager—Remote Execution panel

4. Complete the panel fields as follows:

   - Remote Host

     Enter the host name (if DNS is available) or the IP address of the LPAR where the Started Task Execution Manager procedure commands will be executed.

   - TCP/IP Port

     Enter the TCP/IP port number that the ESFSTEM started task at the remote host uses to listen for incoming requests.

   - Replace Member

     The name and contents of the Dell EMC Started Task Execution Manager procedure member will be sent via TCP/IP to the remote host. Reply **Y** to save this member at the remote host, replacing the content of any like-named member. Reply **N** to save the member using a temporary name until execution completes, at which time the temporary member is deleted.

5. Type **S** next to the selected host to run and press **Enter**.

A pop-up similar to the following is displayed:

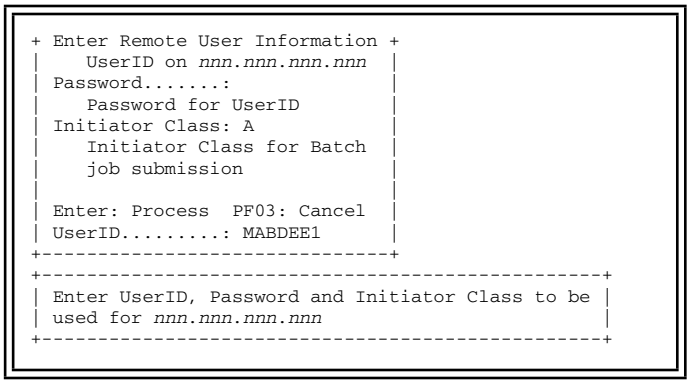



**Figure 125** Enter Remote User Information pop-up

6. Enter the user ID, password, and initiator class for the selected host.
7. Press **Enter**.

   The message "Processing, Please Wait..." displays until the Dell EMC Started Task Execution Manager procedure commands complete on the remote host and a response is received via TCP/IP. When remote host processing is complete, the output of the commands in the procedure displays.

## Executing a procedure within a procedure

A Dell EMC Started Task Execution Manager procedure may also execute another procedure or series of procedures. The command syntax for executing a procedure within a procedure is:

```
EX STEM_procedure_name [HOST(lparname)] [USER(userid)] [BATCH]
```

Where:

- *STEM_procedure_name* is the Dell EMC Started Task Execution Manager JOBPARMS member that contains the z/OS commands to be executed.
- *lparname* is the name of the LPAR where the *STEM_procedure_name* will be executed.
- *userid* is the TSO user ID on *lparname* and is used for the FTP job and setting the security environment for started task ESFSTEMT.
- BATCH indicates the Dell EMC Started Task Execution Manager procedure will be executed as a batch job.

The HOST, USER, and BATCH keywords are optional. The *STEM_procedure_name*, HOST, USER, and BATCH parameters can be separated by either blanks or commas.

If the HOST keyword is not specified, the default is the LPAR where the Dell EMC Started Task Execution Manager ISPF application is executing. This value can be overridden at the time the procedure is executed.

If the USER keyword is not specified, the default is the TSO user ID for the current Dell EMC Started Task Execution Manager ISPF application session. This value can be overridden at the time the procedure is executed.

If the BATCH keyword is not specified, the procedure is run in the defined sequence and subsequent procedure statements will not be executed until this procedure has completed. If the BATCH keyword is specified, the procedure is executed as a batch job and upon job submission the next sequential procedure statement is executed immediately.

Dell EMC Started Task Execution Manager procedures containing EX *STEM_procedure_name* statements can only be executed using the **W** (Watch) or **X** (Execute) line commands.

---

**Note:** Due to the need to provide a password on *lparname*, Dell EMC Started Task Execution Manager procedures that contain EX *STEM_procedure_name* statements cannot be run using the **R** (Remote) line command.

---

When a procedure that contains EX *STEM_procedure_name* statements is executed, Dell EMC Started Task Execution Manager first checks whether the *lparname* is in Dell EMC Started Task Execution Manager host name ISPF table. If the *lparname* is already defined, the IP address and TCP/IP procedure name is extracted from the table.

After you select **W** or **X** in the Dell EMC Started Task Execution Manager Procedure Member List and press **Enter**, a pop-up dialog displays to obtain the TSO (FTP) password for the *userid* on *lparname*:

```
+-- Enter Remote User Information ---+
|                                     |
|    Enter values for X00B            |
|                                     |
| IP Address.....: nnn.nnn.nnn.nnn    |
| TCP STC Name...: TCPIP              |
| UserID.........: NNNNNN             |
| Password.......:                    |
|    Password for UserID              |
| Initiator Class: A                  |
|    Initiator Class for Batch        |
|    job submission                   |
|                                     |
| Enter: Process  PF03: Cancel        |
|                                     |
+-------------------------------------+
```

You can use this panel to override the values of the IP Address, TCP STC Name, UserID and Initiator Class. Override values are only stored to the Dell EMC Started Task Execution Manager host name ISPF table when a table entry does not already exist.

Passwords are stored in REXX and ISPF variables that are deleted when the you exit the Dell EMC Started Task Execution Manager ISPF application.

## Example 1: Cycling a set of address spaces

This example illustrates a scenario where a series of commands is required to shut down an address space (ADDOWN) and another series of commands is needed to start it up (ADUP). It demonstrates how a new procedure (ADCYCL) could be created to execute the existing ADDOWN procedure and then execute the existing ADUP procedure.

The ADDOWN procedure stops the SRDF and SCF address spaces:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 4 of 4
 Command ===>                                                 Scroll ===> PAGE

     _ ADD  - "S" to add new Task(s)

 Line Commands...: E or S or / to edit entry or D to delete the entry
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ====================== ADDOWN - Stop SRDF and SCF ============================
   Sel SeqNbr Issue Command              String to Find             Skip Wait
   --- ------ -------------------------- -------------------------- ---- -----
    _  000010 P EC0MFTSK                                            N      10
    _  000020 P EC0SRDF                  EMCMN10I                   N      10
    _  000030 F EC0SCF,MSC DISABLE       SCF1321I                   N      10
    _  000040 F EC0SCF,INI SHUTDOWN                                 N      10
 ****************************** Bottom of data ********************************
```

The ADUP procedure starts SCF and SRDF:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 5 of 5
 Command ===>                                                 Scroll ===> PAGE

     _ ADD  - "S" to add new Task(s)

 Line Commands...: E or S or / to edit entry or D to delete the entry
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ========================= ADUP - Start SCF and SRDF ==========================
   Sel SeqNbr Issue Command              String to Find             Skip Wait
   --- ------ -------------------------- -------------------------- ---- -----
    _  000010 S EC0SCF                                              Y      15
    _  000020 D IPLINFO                                             N      10
    _  000030 d m=cpu                                               N
    _  000040 S EC0SRDF                  EMCMN03I                   Y      10
    _  000050 S EC0MFTSK                 EC0@006I                   Y      10
 ****************************** Bottom of data ********************************
```

You can combine these two procedures by creating a Dell EMC Started Task Execution Manager procedure named ADCYCL to execute ADDOWN and then ADUP:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 2 of 2
 Command ===>                                                 Scroll ===> PAGE

    _ ADD  - "S" to add new Task(s)
 Line Commands...: E or S or / to edit entry or D to delete the entry
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ====================== ADCYCL - Stop and restart AD ==========================
   Sel SeqNbr Issue Command              String to Find             Skip Wait
   --- ------ -------------------------- -------------------------- ---- -----
    _  000010 EX ADDOWN                                             N      10
    _  000020 EX ADUP                                               N      10
 ****************************** Bottom of data ********************************
```

Both ADDOWN and ADUP will be executed on this LPAR and under the user ID for the Dell EMC Started Task Execution Manager ISPF application. Note the following:

- You will be prompted for the IP address, user ID and password.
- Any overrides (for example, IP address) will apply to both EX commands since neither contains a HOST keyword and both default to the local LPAR.
- Any overrides are only saved if there is not already a host entry for the local LPAR in the Dell EMC Started Task Execution Manager host name ISPF table.

- To keep processing consistent and the code path short, ADDOWN and ADUP are submitted via TCP/IP and execute in the ESFSTEMT address space.
- Since the BATCH keyword is not specified, procedure ADDOWN will execute to completion before procedure ADUP is executed.
- Since the BATCH keyword is not specified, the output of the ADDOWN and ADUP procedures will be written to the ISPF screen.

## Example 2: Executing concurrent batch procedures

This example illustrates a scenario where multiple procedures are executed concurrently using the BATCH keyword.

A procedure named RUNIT contains display commands to obtain the last IPL date and time and the LPAR configuration information:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 2 of 2
 Command ===>                                                 Scroll ===> PAGE

     _ ADD  - "S" to add new Task(s)

 Line Commands...: E or S or / to edit entry or D to delete the entry
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ======================== RUNIT - RUN DISPLAY COMMANDS=========================
   Sel SeqNbr Issue Command              String to Find              Skip Wait
   --- ------ -------------------------- -------------------------- ---- -----
    _  000010 D IPLINFO                                             N      15
    _  000020 D M=CPU                                               N       5
 ***************************** Bottom of data ********************************
```

A procedure named GETIPLI executes procedure RUNIT on three LPARs simultaneously by specifying the BATCH parameter:

```
---------------------- Dell EMC Started Task Execution Manager - Row 1 to 3 of 3
 Command ===>                                                 Scroll ===> PAGE

     _ ADD  - "S" to add new Task(s)

 Line Commands...: E or S or / to edit entry or D to delete the entry
 ENTER: Process   PF3: Cancel   PF7: Up   PF8: Down   PF10: Left   PF11: Right
 ===================== GETIPLI - Collect LPAR Info  ========================
   Sel SeqNbr Issue Command  >           String to Find              Skip Wait
   --- ------ -------------------------- -------------------------- ---- -----
    _  000010 EX RUNIT,HOST(X00B),USER(K                            N       3
    _  000020 EX RUNIT,HOST(X118),USER(K                            N       3
    _  000030 EX RUNIT,HOST(Q311),USER(K                            N       3
 ***************************** Bottom of data ********************************
```

The complete Issue Command statements are not visible in the panel, the full syntax is the following:

```
EX RUNIT,HOST(X00B),USER(KANSAS1),BATCH
EX RUNIT,HOST(X118),USER(KANSAS1),BATCH
EX RUNIT,HOST(Q311),USER(KANSAS1),BATCH
```

When procedure GETIPLI is executed using the **W** (Watch) or **X** (Execute) option:

- You are prompted for the password on each LPAR.
- Two batch jobs are submitted per EX RUNIT statement; one to start the Dell EMC Started Task Execution Manager remote listener via FTP and one to execute the

RUNIT procedure statements on the remote LPAR.

IMPORTANT

It is recommended that you use a secure FTP.

- The output from the commands contained in procedure RUNIT appear in the SYSTSPRT DD in the batch jobs.

To maximize throughput by minimizing job name collisions, the batch jobs are named &*useridx* where *x* is as follows:

**FTP jobs—**replace *x* with 9-0 then Z-A. If a job name that matches the generated jobname is already executing or waiting for execution, the next sequential character is substituted for *x* until an available job name is obtained.

The FTP jobs usually execute very quickly, resulting in most (if not all) FTP jobs being named &*userid*9.

**STEM procedure remote execution jobs—**replace *x* with A-Z then 0-9. If a job name that matches the generated job name is already executing or waiting for execution the next sequential character is substituted for *x* until an available job name is obtained.

The STEM procedure remote execution jobs track the job name assigned to each EX *STEM_procedure_name* statement and will usually result in a series of job names. In the example above, the RUNIT procedures would typically be executed as job names &*userid*A through &*userid*C.

# CHAPTER 7
# Using GDDR Utilities

This chapter includes the following topics:


- GDDR Command Processor (GDDRMCMD) ..... 410
- GDDR Automated Configuration Discovery for DASD (GDDRACDD) ..... 412
- GDDR Environment Check utility (GDDRECHK) ..... 442
- GDDR SRDF Director Overview utility (GDDRDIRS) ..... 447
- GDDR MSC Configuration Validation and Cleanup utility (GDDMSCFX) ..... 450
- GDDR SDDF Session Verification utility (GDDRSDDF) ..... 469
- GDDR SRDF Device Status Check Utility (GDDRSC06) ..... 476
- GDDR TimeFinder Management utility (GDDRTF20) ..... 478
- GDDR BCV Reporting utility (GDDRTF00) ..... 483
- BCVGROUP Validation utility (GDDRBCVG) ..... 485
- GDDR Invalid Track Monitor utility (GDDRMIN0) ..... 486
- GDDR DLm Serial Number utility (GDDRDLSN) ..... 491
- GDDR IPL Parameter Swap utility (GDDRISWP) ..... 492
- GDDR IPL Assist Monitor utility (GDDRGIAM) ..... 495
- GDDR Workload Assist Monitor utility (GDDRHCMD) ..... 501
- GDDR HMC Actions utility (GDDRHMCA) ..... 503
- GDDR BCPii Connectivity Test utility (GDDRBCPI) ..... 505
- GDDR Load Profile Management utility (GDDRLPRF) ..... 508
- GDDR ECGUTIL Driver utility (GDDRECG0) ..... 510
- GDDR Expected Events utility (GDDREE00) ..... 511
- GDDR DIV Management utility (GDDRGVX) ..... 513
- GDDR Command Queue utility (GDDRXCMD) ..... 514
- GDDRMAIN Trace Print utility (GDDRTRCP) ..... 515

# GDDR Command Processor (GDDRMCMD)

GDDRMCMD is a GDDR command processor. GDDRMCMD supports the following commands:

- CHECKUP described on page 182
- CONFIG described on page 194
- ENQ described on page 199
- GATEK described on page 201
- LICENSE described on page 207
- MAINTENANCE described on page 211
- PARM_REFRESH described on page 218
- REGION described on page 220
- SET described on page 232
- SUMMARY described on page 237
- SVCDUMP described on page 239
- SUBSYS described on page 235
- SYSTEMS described on page 240
- TOPOLOGY described on page 243
- TRACE RESET described on page 245

All commands supported by GDDRMCMD can be issued using any of the following methods:

- As a GDDRMAIN modify command (F GDDRMAIN,*command*) per Table 26 on page 178
- As a Perform Health Check panel primary command as described in "Perform GDDR health check (C)" on page 366
- Using the GDDRMCMD batch interface described in "GDDRMCMD batch interface" on page 410

## GDDRMCMD batch interface

### Sample JCL

The following sample JCL (GDDRMCMD in SAMPLIB) can be used to execute GDDRMCMD in batch, with some modifications:

```
//GDDRMCMD JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//GDDRMCMD EXEC PGM=GDDRMCMD,PARM='<gddrmcmd-command>'
//STEPLIB  DD  DISP=SHR,DSN=DS-PREFIX.LINKLIB  <--- Your GDDR LINKLIB
//SYSPRINT DD  SYSOUT=*
//GVAROUT  DD  SYSOUT=*
//SYSABEND DD  SYSOUT=*
//SCF$EMC  DD  DUMMY  <--- Your SCF subsystem name ('EMC' is default)
//GDD$GDDR DD  DUMMY  <--- Your GDDR subsystem name ('GDDR' is default)
//*
```

Where:

- The STEPLIB must specify your GDDR LINKLIB.
- The SCF$*nnnn* and GDD$*nnnn* DDs link the GDDRMCMD utility to the correct SCF and GDDRMAIN address spaces:
  - The *nnnn* in your SCF$*nnnn* DD DUMMY must match your SCF subsystem name.

    The default SCF subsystem name is 'EMC', in which case the SCF$*nnnn* DD DUMMY is optional.
  - The *nnnn* in your GDD$*nnnn* DD DUMMY must match your GDDR subsystem name.

    The default GDDR subsystem name is 'GDDR', in which case the GDD$*nnnn* DD DUMMY is optional.
- The EXEC PARM specifies the *gddrmcmd-command* to execute followed by any command parameters.

## Return codes

GDDRMCMD return codes are listed in Table 37.

**Table 37** GDDRMCMD return codes

| Return code | Description |
|---|---|
| 0 | Successful, all entries in a valid state |
| 4 | Warning, one or more entries in an undesirable state |
| 8 | Error, one or more entries in an invalid state |
| 12 | Severe error, processing failed unexpectedly |
| 16 | Abend occurred, SCVDUMP is taken |
| 20 | Fatal error, one of the following errors occurred:<br>◆ LICENSE command: License mismatch detected<br>◆ MAINTENANCE command: GDDR version mismatch detected<br>◆ PARM_REFRESH command: No active GDDR licenses found, or this GDDR instance is prohibited from running for one of the following reasons:<br>■ One or more other C-systems for which the GDDR C-System Multi-Tenancy feature is enabled, are active on this z/OS system. This GDDR instance is not part of GDDR C-System Multi-Tenancy and cannot run concurrently with any other GDDR instance on the same z/OS system.<br>■ Another C-system for which the GDDR C-System Multi-Tenancy feature is not enabled, is active on this z/OS system and cannot run concurrently with any other GDDR instance on the same z/OS system.<br>■ A P-system is active on this z/OS system. P-systems cannot run concurrently with any other GDDR instance on the same z/OS system.<br>■ One or more C-systems are active on this z/OS system. This GDDR instance is a P-system and cannot run concurrently with any other GDDR instance on the same z/OS system. |

# GDDR Automated Configuration Discovery for DASD (GDDRACDD)

The GDDR Automated Configuration Discovery for DASD utility (GDDRACDD) serves the following purposes:

- Discovers SRDF and TimeFinder as well as DLm backend devices in a set of defined storage systems and SRDF groups.
- Generates parameters usable to define the discovered devices to GDDR.
- Validates the discovered configuration.
- Validates existing RDF.DEVICES, DLM.DEVICES, and BCV-related parameters and other configuration global variables against the discovered DASD configuration and against GDDRPARM information.
- Produces a set of reports to describe the configuration:
  - SYMMDEVS: lists contiguous device ranges
  - RDFGROUP: lists SRDF groups connecting GDDR-managed storage systems
  - CREPAIR: lists #SC VOL CREATEPAIR commands usable to create the GDDR-managed device pairs
  - FBAMETA: lists FBA meta devices and SRDF relationships
  - TFLIST: reports GDDR-managed BCV devices

The following operating modes are provided:

- Discovery mode

  Run GDDRACDD in Discovery mode to discover a PowerMax/VMAX DASD configuration and define it to GDDR. Discovery mode is typically run on the same system that has the GDDR Parameter Wizard session active, and only there. There is no GDDR requirement to run it on any other system.

- Validation mode

  Run GDDRACDD in Validation mode to verify that a previously defined configuration is still correct. Validation mode is typically part of the check-list used before any GDDR script is run. Validation mode is also used at the end of the GDDRPEDD script to validate that all the changes during dynamic device add/delete result in a valid configuration.

The utility can be run at any site in any GDDR configuration.

GDDRACDD can be invoked from option **G** on the GDDR Maintenance and Setup Menu (see "Manage automated configuration discovery for DASD (M,P,G)" on page 293) or in batch mode. The GDDRACDD utility is supported with REXX module GDDRACDD and sample JCL in *hlq*.GDDR*vrm*.SAMPLIB(GDDRACDD). Member GDDRPACD in *hlq*.GDDR*vrm*.PARMLIB provides parameters.

"Performing automated DASD discovery for active GDDR Parameter Wizard session" on page 294 provides a procedure used to perform automated configuration discovery for DASD using the GDDRACDD utility.

## Requirements

The requirements to run GDDRACDD are:

- ACDDPARM DD must point to a dataset containing GDDRACDD parameters. GDDRACDD parameters can also be in-stream as shown in the sample JCL.
- SCF must be running on the local system. The SCF$*nnnn* statement must provide connection to the running SCF STC.
- GDDRMAIN must be up and running

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRACDD). Customize it to meet the requirements of your configuration.

```
//GDDRACDD JOB CSE,TF51,CLASS=A,MSGCLASS=X,NOTIFY=USER007,

//GDDRACDD JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//GDDRACDD EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 ISPSTART CMD( GDDRACDD 256 ALIGN                                    +
               NOSIZE NORAID   META                                  +
               BCV(N) DLM(N)                                         +
               GLOBALS(NONE)                                         +
               SORT                                                  +
               CONFIG(DC1,CONCURRENT)                                +
               DEBUG(0)                                              +
               TRACE(ALL,0)                                          +
                     )
/*
//GDDR.SYMMDEVS DD SYSOUT=*
//GDDR.RDFGROUP DD SYSOUT=*
//GDDR.CREPAIR  DD SYSOUT=*
//GDDR.FBAMETA  DD SYSOUT=*
//GDDR.TFLIST   DD SYSOUT=*
//*DDR.E04SRDFD DD DISP=SHR,DSN=.....
//*DDR.E05TFDEV DD DISP=SHR,DSN=.....
//*DDR.ACDDPARM DD DISP=SHR,DSN=.....
//GDDR.ACDDPARM DD *
* ----+----1----+----2----+----3----+----4----+----5----+----6----+----
CSYSSITE SYS1     GDDR DC1
CSYSSITE SYS2     GDDR DC2
CSYSSITE SYS3     GDDR DC3
CSYSSITE SYS4     GDDR DC4
*
SYMM     DC1(H)   GDDR 123456789011,A000-A00F,AB10,AF08-AF0F
SYMM     DC2(H)   GDDR 123456789012,B000-B00F,...
SYMM     DC2      GDDR 123456789012,B000-B00F,...
SYMM     DC3(H)   GDDR 123456789013,C000-C00F,...
SYMM     DC4(H)   GDDR 123456789014,D000-D00F,...
*
PATH     DC1      GDDR 123456789011,A000
PATH     DC1      GDDR 123456789012,A000.10
PATH     DC1      GDDR 123456789013,A000.20
PATH     DC1      GDDR 123456789014,A000.10.30
PATH     DC2      GDDR 123456789011,B000.10
PATH     DC2      GDDR 123456789012,B000
PATH     DC2      GDDR 123456789013,B000.10.20
PATH     DC2      GDDR 123456789014,B000.30
PATH     DC3      GDDR 123456789011,C000.20
PATH     DC3      GDDR 123456789012,C000.40.30
PATH     DC3      GDDR 123456789013,C000
PATH     DC3      GDDR 123456789014,C000.40
PATH     DC4      GDDR 123456789011,D000.40.20
PATH     DC4      GDDR 123456789012,D000.30
PATH     DC4      GDDR 123456789013,D000.40
PATH     DC4      GDDR 123456789014,D000
*
VGROUP   50       GDDR 123456789012,123456789013
*
INCLUDE  DC1      GDDR I,2000,10,20
INCLUDE  DC2      GDDR I,6000,30
```

```
INCLUDE  DC1      GDDR E,2000,00,F9
INCLUDE  DC2      GDDR E,6000,31
INCLUDE  DC1      GDDR I,DLM,2000,12,FB
INCLUDE  DC2      GDDR I,DLM,6000,13
*
GENERATE DC3      GDDR I,A000,(20,30),(60,70),...
GENERATE DC3      GDDR E,A000,(F9,31),(61,71),...
*
TFINDER  ALL      GDDR MIRR,RA(INT),GOLD(WARN)
*
```

## Arguments

All GDDRACDD arguments are optional. If no arguments are specified, GDDRACDD runs with the default values.

The arguments can be specified in any order, except for the *range_size* which is the first argument (if specified).

*range_size*

Defines the maximum number of contiguous devices to be listed as a single range in the device ranges report and in RDF.DEVICES parameters.

Default: 256

Supported *range_size* values are 0 to 4096. The recommended range size is the default value of 256. You can use a very small value; for example, in a test environment to force GDDRACDD to generate a lot of RDF.DEVICES parameters. Larger range sizes (> 512) are intended for code testing purposes.

If you want to specify *range_size*, it *must* be the first argument. If you specify any other arguments, you must also specify a *range_size*. The other arguments can be specified in any order.

---

**Note:** For FBA meta devices, a hard-coded maximum range size of 128 is enforced, but you can use a smaller value.

---

ALIGN | NOALIGN

Determines whether to align device ranges.

Default: ALIGN

If ALIGN is specified (or defaulted), then device ranges will be aligned on boundaries defined by *range_size*. Alignment is supported on 256-, 512-, 1024-, 2048-, and 4096-device boundaries. If a range size smaller than a supported alignment size is specified, the *range_size* value is rounded up to the next alignment size. GDDRACDD informs you about a range size override if this happens.

If ALIGN is specified, and GDDRACDD discovers an asymmetrical configuration, it will issue a warning message (GDDP415W) and change to NOALIGN automatically when the asymmetry is detected.

If NOALIGN is specified, then device ranges will not require alignment to boundaries defined by *range_size*, allowing device ranges to span alignment boundaries.

Examples:

- `ACDD 256 ALIGN`

  Shows 003F-00FF then starts a new range going from 0100-013F.

- `ACDD 16 ALIGN`

  Forces the range_size to 256, and aligns as described above.

- `ACDD 256 NOALIGN`

  Could show a range going from 003F to 013F as a single contiguous range.

`BCV(Y|N)`

Performs discovery of TimeFinder devices.

**Note:** "TimeFinder device management" on page 75 describes GDDR support for TimeFinder devices.

Default: BCV(N)

When BCV(Y) is specified, GDDRACDD discovers TimeFinder devices in all or selected storage systems and generate GDDR BCV parameters for them. Site, storage systems, SRDF groups, and/or BCV sets can be selected using TFINDER parameters in the ACDDPARM deck.

- BCV(Y) causes GDDRACDD to write PARMS.STDBCV.*t*.DC*n*.*n*=*gk*,*bcvl*-*bcvh*,*stdl*-*stdh*,*h* parameters to the E05TFDEV dataset.

  **Note:** See "TFINDER DCn|ALL GDDR [{MIRR|CLON},] {RA({INT|EXT|ALL})|RA(srdfgrp,srdfgrp,...)|NONE}, GOLD({WARN|ERROR}),TEST({WARN|ERROR}) [,GK(gk,gk,...)]" on page 426 for more information about the PARMS.STDBCV GDDR parameter format.

`CONFIG(`*`primary-dasd-site`*`)`

Specifies the intended primary DASD site used by GDDRACDD to generate CREATEPAIR commands.

**Note:** The CONFIG argument affects the syntax of the CREATEPAIR commands generated by GDDRACDD. It has no effect on configuration validation or generation of RDF.DEVICES parameters.

By default, GDDRACDD uses the site where it is running.

`DEBUG(`*`n`*`)`

Sets the DEBUG level for GDDRACDD.

Default: 0.

Valid values for *n* are 0, 1, or 2.

`DLM(Y|N)`

Performs discovery or validation of DLm backend devices.

Default: N

When DLM(Y) is specified, GDDRACDD performs the same discovery and validation functions for DLM backend devices as it does for other GDDR-managed SRDF devices.

The DLM backend devices are identified to GDDRACDD by INCLUDE arguments with the DLM keyword. See "INCLUDE" on page 421 for more information.

`FORCE|NOFORCE`

Determines if GDDRACDD should force execution (FORCE) or stop immediately (NOFORCE) if certain configuration errors are found.

Default: NOFORCE

See message GDDP420E in the *GDDR Message Guide* for a description of the types of errors detected. Cases where RC=8 is set can be rerun with FORCE, but there is no guarantee that the generated RDF.DEVICES parameters (if any) will be usable by GDDR.

`GDDRINFO(TRACE)`

Generates trace records for all GDDRINFO calls, one line per call. Output is written to the GDDRINFO DD, which is dynamically allocated to SYSOUT.

`GDDRPFX(GDD`*n*`)`

Indicates the target GDDR subsystem.

Default: GDDRPFX(GDDR)

`GDDRQDEV([`*func*`[,`*func*`,...]`
`[,EXTRA]][,SNAPAPI][,SYMMAPI][,TRACE])`

Controls GDDRQDEV debugging and tracing.

*func*`[,`*func*`,...]`

Generates debugging information for the listed GDDRQDEV functions. This can also be specified as ALL to generate debugging information for all GDDRQDEV functions (not recommended except in very small environments).

Output is written to the DBUGQDEV DD, which is dynamically allocated to SYSOUT.

`EXTRA`

Generates additional debugging information for the requested GDDRQDEV functions (or ALL).

`SNAPAPI`

Generates Snap API debugging information for the CLON and SNVX functions. This can be restricted to just one of these by specifying the function name in addition to the SNAPAPI option, otherwise both CLON and SNVX are implied.

Output is written to the EMCQCAPI and/or ESNAZDPL DDs, which are dynamically allocated to SYSOUT.

SYMMAPI

Generates trace records for all SymmAPI calls, one line per call. This cannot be restricted to only certain GDDRQDEV functions. Any listed functions have no effect on this option.

Output is written to the EMCSAIL DD, which is dynamically allocated to SYSOUT.

TRACE

Generates trace records for all GDDRQDEV calls, one line per call. This cannot be restricted to only certain GDDRQDEV functions. Any listed functions have no effect on this option.

Output is written to the GDDRQDEV DD, which is dynamically allocated to SYSOUT.

GLOBALS(CHECK|NONE)

Selects GDDRACDD operating mode: validation (CHECK) or discovery (NONE).

Default: NONE

Validation of RDF/DLM.DEVICES parameters includes the following items, both for internal and external devices, except where noted otherwise:

- Syntactical validity of existing global variable names and values
- Existence of required RDF.DEVICES parameters for all site pairs in the configuration (internal devices only)
- Internal consistency of existing global variables against each other
- Consistency of global variables against GDDRPARM information
- Consistency of global variables against the discovered storage system configuration:
  - Discovered devices are defined and are the correct R1/R2 type
  - Discovered devices are Link-Blocked where expected (internal groups only)
  - Discovered devices are not Link-Blocked where not expected (internal groups only)
  - Defined devices are discovered and are the correct R1/R2 type

Errors during syntax and consistency validation of RDF/DLM.DEVICES parameters result in GDDP420E and RC=16.

- For discrepancies between RDF.DEVICES parameters and devices discovered in the storage system for internal devices, GDDRACDD issues message GDDP420E and terminates with RC=8 or RC=16.
- For discrepancies between RDF.DEVICES parameters and devices discovered in the storage system for external devices, GDDRACDD issues message GDDP421W, and terminates with RC=4. If this occurs during the GDDRPEDD script, the RC is 0.

Validation of BCV-related parameters includes the following:

- Syntactical validity of parameter name and value

- Consistency of BCV-related parameters with GDDRPARM SYMM parameters
- Consistency of BCV-related parameters with the discovered configuration:
  - Defined BCV devices are BCV devices (TimeFinder target devices)
  - Defined BCV devices are associated with GDDR-managed internal/external devices
  - Defined BCV devices are paired with expected STD devices
  - Sufficient BCV devices are defined and paired
  - Discovered BCV devices are defined to GDDR
  - No BCV devices are defined twice in different parameters of the same set
  - The defined and discovered BCV devices form a complete image of all GDDR-managed SRDF devices at sites selected for GDDR BCV management
  - The call override bytes for BCV management at any given site are matched by existing BCV devices at the affected site
  - SnapVX target devices are linked to GDDR-managed SRDF/DLM devices using the GDDR-defined snapshot names for the proper set of TimeFinder target devices

---

**Note:** Errors found during BCV validation result in GDDP421W and RC=0, except for GOLD/TEST set completeness validation, which results either in GDDP420E (RC=8) or GDDP421W (RC=4), depending on specifications on TFINDER parameters.

---

If GLOBALS(CHECK) is specified, GDDRMAIN must be running with GDDR 5.0 or later. GLOBALS(CHECK) is mutually exclusive with COMP(Y).

If GLOBALS(CHECK) is specified, ACDDPARM content is ignored with the exception of PATH parameters. GDDRACDD will use CSYSSITE and SYMM parameters from the GDDRPARM file and fabricate INCLUDE, GENERATE, TFINDER, SOFTLINK, and SNAPNAME parameters as required, using information from the existing RDF.DEVICES parameters. These fabricated parameters are echoed to the joblog prefixed with 'SIM:'.

`GROUPS(BREAK|NOBREAK)`

Discovered PowerMax/VMAX device ranges can be paired in multiple SRDF groups, currently limited to two by the operating environment. By default, GDDRACDD generates RDF.DEVICES parameters for each site pair independently of device pairing for other site pairs.

If GROUPS(BREAK) is specified, then if a range needs to be split due to an interruption in the device pairing in one SRDF group, it will also cause a split in RDF.DEVICES parameters for the other SRDF group. As a result there will be a higher total number of RDF.DEVICES parameters. This number will be higher than necessary for GDDR scripts to succeed. However, the generated RDF.DEVICES parameters will be symmetrical in nature when looking at parameters for different site pairs.

Default: NOBREAK

NOSIZE
NORAID
NOMETA

These optional parameters affect the generation of RDF.DEVICES parameters. . Each of them can be absent (default) or specified as NOSIZE / NORAID / NOMETA in any combination.

By default (absence of these 3 arguments), GDDRACDD generates a new device range, and consequently a new RDF.DEVICES parameter, every time any of the following attributes of the discovered SRDF devices changes:

- CKD versus FBA, meta-device versus regular device, STD device versus BCV.
- Non-contiguous other-side or hopped other-side devices. (All discovered SRDF groups between storage systems defined to GDDRACDD are considered.)
- Different number of cylinders.
- Different local RAID-scheme (RAID-0, RAID-1, RAID-5, RAID-6, none).

If NOSIZE, NORAID and/or NOMETA is specified, then a difference in cylinder count or local protection scheme, or meta-type will still generate a new device range in the PowerMax/VMAX device ranges report, but will no longer cause an additional RDF.DEVICES parameter to be generated.

SCAN

Indicates SCAN mode. Can be absent, or specified as SCAN.

If the SCAN argument is specified, GDDRACDD performs a validation of the arguments and parameters, as well as a limited infrastructure validation (SRDF group status). It will not proceed to PowerMax/VMAX device discovery.

SORT | NOSORT

Determines whether to sort the generated parameters.

Default: SORT

By default, GDDRACDD sorts the generated RDF.DEVICES parameters in turn over the available storage systems. Internal SRDF groups are sorted separately from external SRDF groups.

Sorting enhances GDDR script performance by increasing parallel processing over available resources. It provides the most benefit for users with multiple storage system.

NOSORT suppresses sorting of generated parameters.

Note the following:

- If one site of a site pair has a lower number of storage systems than the other site, the sort algorithm will do the distribution using the site with the lowest number. This avoids cases where a site with low parallel processing capabilities would get overrun by parallel processing initiated from the other site.
- This sorting of generated parameters does not force a strict sequence of execution of script commands in the desired round-robin fashion, but it increases the likelihood of an optimal sequence.

- If the managed configuration has an uneven distribution of device ranges across the storage systems there will be a “tail-end” effect where the remaining commands are all targeting the same storage system.

`TRACE([subroutine-name,trace-level)`

Sets the TRACE level for GDDRACDD.

**⚠WARNING**

**Use TRACE only under instruction from the GDDR Solution Support team. Tracing of GDDRACDD can produce enormous amounts of output in the SYSTSPRT dataset, and could consume all available spool space. The default value is TRACE(0).**

Default: TRACE(ALL,0)

Subroutine names are provided by the GDDR Solution Support team. The default subroutine name is "ALL".

TRACE level can be 0, 1, 2, or 3 and translates to REXX tracing values of "O", "R", "I", and "A" respectively.

## Optional DD cards

There are optional DD cards in the GDDRACDD JCL which slightly affect GDDRACDD behavior. If these DD cards are not present, the affected report is not written. It is recommended to allocate all corresponding datasets.

- E04SRDFD

  The E04SRDFD dataset contains the generated RDF.DEVICES parameters.

  This DD card must point to the GDDR Parameter Wizard work dataset member E04SRDFD. It is necessary for proper integration of GDDRACDD with the GDDR Parameter Wizard.

- E05TFDEV

  The E05TFDEV dataset contains the generated STDBCV.*t*.DC*n* or BCV.DC*n*.ONLY parameters.

  This DD card must point to the GDDR Parameter Wizard work dataset member E05TFDEV. It is necessary for proper integration of GDDRACDD with the GDDR Parameter Wizard.

- SYMMDEVS

  Lists contiguous PowerMax/VMAX device ranges in all storage systems discovered by the GDDRACDD utility. It provides better visibility to various warning and error messages written to the joblog. It is recommended to have this DD card in the JCL.

- RDFGROUP

  Contains the following sections:

  - A listing of all GDDR groups found in storage systems defined on SYMM parameters with indication of whether a group is included as regular DASD, DLM backend, internal, external or not at all, whether the group is offline, and whether the group is virtual.

- A listing of included groups by type and site pair, with associated gatekeepers.

- CREPAIR

  Contains the SRDF Host Component #SC VOL CREATEPAIR commands which can be used to create the configuration as it will be defined to GDDR, from scratch, in cases where it got deleted. The CREATEPAIR commands are generated according to the intended configuration specification based on the CPDS and TOPOLOGY information in the CONFIG argument.

- FBAMETA

  Lists FBA meta devices in the GDDR-managed configuration. It shows head-member relationships and SRDF device pairing information.

- TFLIST

  Lists STD-BCV device pairs for BCVs associated with GDDR-managed SRDF devices.

## ACDDPARM parameters

All ACDDPARM parameters are optional, except for the INCLUDE parameter. If CSYSSITE or SYMM parameters are specified in ACDDPARM, then GDDRACDD will use those parameters, even if GDDRMAIN is running. All ACDDPARM parameters, except PATH, are ignored when running GDDRACDD in Validation mode (with the GLOBALS(CHECK) argument specified).

ACDDPARM has specific syntax rules. All parameters are column-sensitive. You can specify GDDRACDD parameters in any sequence.

The parameter structure (columns 1-70 usable) is as follows:

```
0000000000111111111122222222223333333333444444444455555555556666666667
1234556789012345678901234567890123456789012345678901234567890123456789
PARMNAME  REF-id   GDDx value
```

- Parameter name goes in columns 1-8.
- System name or a site name or a group number, goes in columns 10-17.
- GDD*x* goes in column 19-22. This is the *targeted GDDR subsystem*.
- Parameter values go in columns 24-71.

### Required parameters

INCLUDE

The mandatory INCLUDE parameter instructs GDDRACDD whether to include discovered SRDF devices in the generation of GDDR RDF.DEVICES or DLM.DEVICES parameters. Inclusion is done at the SRDF group level and can include consistency protected (internal) SRDF groups as well as SRDF groups outside of the consistency protection (external groups) both for “regular” DASD devices and for DLm backend devices.

**Note:** Note: Although GDDRACDD allows definition of external SRDF groups for DLm backend devices, GDDR currently has no use for them.

The INCLUDE parameter limits the creation of RDF.DEVICES and DLM.DEVICES parameters to included groups. It has no effect on the discovery process. GDDRACDD will always perform a complete discovery of all devices in all storage systems defined to it on SYMM parameters. The discovery is also not limited by SCF.CNTRL.EXCLUDE.LIST parameters in SCFINI. SCF.DEV.EXCLUDE.LIST parameters in SCFINI should be considered for the selection of discovery paths specified on PATH parameters.

INCLUDE parameters are required (unless GLOBAL(CHECKS) is specified) and GDDRACDD will not run if no INCLUDE parameters are found in ACDDPARM. If GDDRACDD runs in validation mode, it ignores any INCLUDE parameters in ACDDPARM and generates syntactically valid INCLUDE parameters, based on the existing RDF.DEVICES and DLM.DEVICES parameters. Every group for which RDF.DEVICES or DLM.DEVICES parameters are needed must have an INCLUDE parameter. There is only one exception to this, which is discussed in "GENERATE" on page 423.

---

**Note:** Each SRDF group connects two storage systems, but only one INCLUDE is necessary per group, as the other end is automatically included.

---

Each INCLUDE parameter must specify:

- An inclusion type, which is either 'I' (Internal) or 'E' (External)
- A gatekeeper address. This is any MVS address that locates the intended storage system and was specified in the relevant gatekeeper pool-set on a SYMM parameter either in ACDDPARM or GDDRPARM.
- A comma-separated list of SRDF groups contained in the storage system located by the specified gatekeeper. This can be any mixture of SRDF/S and SRDF/A groups.

INCLUDE parameters can optionally have a DLM keyword, which identifies them for generation of DLM.DEVICES parameters as opposed to RDF.DEVICES parameters. If specified, the DLM keyword must follow the inclusion type ('I' or 'E') immediately.

---

**Note:** The RDF.DEVICES GDDR variables include both regular DASD devices and DLm backend devices. The DLM.DEVICES variables include only the DLm backend devices. This allows GDDR to manage DLm backend devices in the same script step as the regular DASD devices, as well as managing them separately for those actions that do not apply to regular DASD devices.

---

---

**Note:** All SRDF groups listed on INCLUDE parameters must have SRDF device pairs in the storage system. See "GENERATE" on page 423 for a discussion of the creation of RDF/DLM.DEVICES parameters for groups that do not have any SRDF device pairs.

---

Rules for GDDR-managed SRDF groups:

- All SRDF groups known to GDDR must be "pure": either all consistency protected devices or all external devices; no mixture is allowed within the same SRDF group. Either regular DASD or DLm backend, no mixture is allowed within the same SRDF group.
- All SQAR devices (these are in internal groups by definition) must be protected in SRDF groups so that they form closed "squares" at the device level.
- External SRDF/S leg devices can be concurrent or cascaded with an SRDF/A leg SRDF group, which must also be an external SRDF group.
- External SRDF/A leg devices can be concurrent or cascaded with an SRDF/S leg SRDF group, which can be an external SRDF group or a consistency protected SRDF group.

Examples:

```
INCLUDE  DC1      GDDR I,2000,10,20
INCLUDE  DC2      GDDR I,6000,30
INCLUDE  DC1      GDDR E,2000,00,F9
INCLUDE  DC2      GDDR E,6000,31
INCLUDE  DC1      GDDR I,DLM,2000,16,26
INCLUDE  DC2      GDDR I,DLM,6000,36
```

## Optional parameters

CSYSSITE

The CSYSSITE parameter links a system name to a site.

The CSYSSITE parameter tells GDDRACDD the site on which it is running. GDDRACDD uses this information to properly construct CREATEPAIR commands, if requested, and to select appropriate PATH parameters if any are specified.

GDDRACDD does not run on a system that is not named in a CSYSSITE parameter.

It is recommended to specify one CSYSSITE parameter for every C-system in the configuration. This makes the GDDRACDD parameter deck portable. However, GDDRACDD works with only one CSYSSITE coded; the one for the system where you intend to run GDDRACDD next.

CSYSSITE parameters in ACDDPARM are optional if you are satisfied with the CSYSSITE parameters in the GDDRPARM file.

If you specify any CSYSSITE parameter in ACDDPARM, then any CSYSSITE parameter in the GDDRPARM file is ignored.

Example:

```
CSYSSITE O016     GDDR DC1
```

GENERATE

The optional GENERATE parameter enables generation of RDF.DEVICES parameters and CREATEPAIR commands for the "cross-hair" SRDF groups connecting DC1 to DC4 and DC2 to DC3 based on the comma-separated list of pair of SRDF groups specified in the GENERATE parameter.

The site specified on GENERATE parameters can be DC3 or DC4.

Each GENERATE parameter must further specify:

- A type, which is either 'I' (Internal) or 'E' (External). This type must match the INCLUDE type of one the paired groups
- A “gatekeeper” address: this is any MVS address that locates the intended storage system and was specified in the relevant gatekeeper pool-set on a SYMM parameter either in ACDDPARM or GDDRPARM.
- A list of pairs of SRDF groups. Within each pair, at least one group must have been previously specified on an INCLUDE parameter with the same type as specified on the GENERATE parameter. The second group is automatically included, if it was not included earlier. One of the groups must have existing device pairs. Only one of the paired groups can be a virtual group (see “VGROUP” on page 430).

Generation for non-existing device pairs DC2-DC3 requires existing device pairs DC1-DC3 and DC1-DC2. Generation for non-existing device pairs DC1-DC4 requires existing device pairs DC2-DC4 and DC2-DC1. Generation for non-existing device pairs DC1-DC2, DC1-DC3, DC2-DC4 and DC3-DC4 is not supported.

Examples:

```
GENERATE DC3      GDDR I,A000,(20,30),(70,60)
GENERATE DC3      GDDR E,A000,(F9,31),(61,71)
```

PATH

By default GDDRACDD determines the discovery path for every defined storage system automatically. This is true whether the SYMM parameters are taken from ACDDPARM or from the GDDRPARM file. The algorithm selects the path with the fewest possible hops. For remote storage systems, the selected path is the one with the highest number of online local and remote SRDF directors.

If the automatically selected path is unacceptable, a PATH parameter can be specified in ACDDPARM to force usage of the specified path for discovery.

The site specified on PATH parameters, or “from” site, determines whether or not the PATH parameter will be used. GDDRACDD determines the site where it is running based on CSYSSITE parameters. Any PATH parameter referring to a different site is ignored. This again allows for a portable ACDDPARM parameter deck.

The PATH parameters further specify a target storage system by serial number, and finally the discovery path. This path is made up of a gatekeeper address, which must be usable at the “from” site, followed by as many “hops” as required to reach the target storage system, in a dot-separated list. The number of hops that can be specified is limited to eight.

Example:

```
PATH     DC1      GDDR 123456789014,A000.10.30
```

SYMM

The SYMM parameter defines a storage system to GDDRACDD, as well as the pool of gatekeeper addresses to be distributed by GDDRACDD over the generated RDF.DEVICES or DLM.DEVICES parameters.

**Note:** The SYMM statements in ACDDPARM are ignored if GLOBALS(CHECK) is specified.

If you specify SYMM parameters in ACDDPARM, then GDDRACDD requires one SYMM parameter with the (H) Home indicator in the site-id column, for every storage system in the GDDR-managed configuration. You can code as many SYMM parameters as needed to define the gatekeeper pools. These can be lists of single devices or device ranges. Gatekeeper addresses must be unique at a site.

SYMM parameters in ACDDPARM are optional if you are satisfied with the SYMM parameters in the GDDRPARM file.

If you specify SYMM parameters in ACDDPARM, you can copy and paste all SYMM parameters from the GDDRPARM file. SYMM parameters without the (H) Home indicator in the site-id column are only used by GDDRACDD if they refer to the site where GDDRACDD is running. In that case, GDDRACDD uses the first gatekeeper listed as a discovery path for the named storage system, unless there is an explicit PATH parameter for the same storage system.

If you specify any SYMM parameter in ACDDPARM, then any SYMM parameter in the GDDRPARM file is ignored.

When using GDDRACDD SYMM parameters, ensure they specify the same gatekeeper devices as those on the corresponding SYMM parameters in the GDDRPARM file. This ensures that the RDF.DEVICES, DLM.DEVICES, and STDBCV ranges returned by GDDRACDD will pass validation. "Restoring normal operations after MSC THAW" on page 590 provides further details.

Example:

```
SYMM      DC1(H)    GDDR 000292601023,2000-200F
```

SNAPNAME

The optional SNAPNAME parameter specifies the 1-32 character name of the snapshot to be used by GDDR for the indicated set of TF target devices (GOLD vs. TEST, INT vs. EXT). For example:

```
SNAPNAME            GDDR GOLD,INT,GDDR_INT_MY_SNAP
```

At least one SNAPNAME parameter must be provided when a SOFTLINK parameter with value YES is specified. SNAPNAME parameters have the same environmental requirements as indicated for the SOFTLINK parameter. Note that SNAPNAMEs defined to GDDR are not qualified by SITE. A set of TF target devices is managed using the same snapshot name, regardless of the site to which it belongs.

GDDR-managed snapshot names have the following requirements:

- For internal devices use prefix: GDDR_INT_

GDDR_INT_* snapshot names must be used for devices in or paired with consistency protected SRDF groups (ConGroup and/or MSC).

- For external devices use prefix: GDDR_EXT_

  GDDR_EXT_* snapshot names must be used for devices in or paired with ADCOPY-DISK SRDF groups.

You can append up to 23 characters to form a syntactically valid snapshot name.

`SOFTLINK DC`*`n`* `GDDR {YES[,TarGeTLESS]|NO}`

The optional SOFLTINK parameter specifies whether SnapVX softlinking support is required for a site.

"SnapVX softlinking support" on page 77 discusses SnapVX softlinking support.

SOFTLINK requires at least one SNAPNAME parameter to be specified.

---

**Note:** SOFTLINK parameters are not supported when running GDDRACDD in legacy BCV discovery/validation mode using the BCV(Y,INC) argument.

---

`TarGeTLESS`

Enables you to define a completely targetless SnapVX configuration.

---

**Note:** "Targetless SnapVX support" on page 77 describes this GDDR feature.

---

TARGETLESS is ignored if the SOFTLINK parameter is set to NO.

.Aliases include TGTLESS or TLESS.

```
TFINDER DCn|ALL  GDDR
[{MIRR|CLON},]
{RA({INT|EXT|ALL})|RA(srdfgrp,srdfgrp,...)|NONE},
GOLD({WARN|ERROR}),TEST({WARN|ERROR})
[,GK(gk,gk,...)]
```

Determines which types and sets of BCVs should be discovered.

---

**Note:** See "GOLD and TEST sets of BCVs" on page 75 for information about OLD and TEST sets of BCVs.

---

The optional TFINDER parameter serves the following purposes:

- For VMAX 10K, 20K, and 40K systems, TFINDER selects the Legacy TF method (TF/Mirror or TF/Clone) to be used by site.
  - If SnapVX with softlinking is required, GDDR uses TF/Clone on VMAX 10K, 20K, and 40K systems, ignoring the Legacy TF method selection.
  - If TF Clone is selected as the Legacy TF method, then GDDR will use SnapVX with softlinking on PowerMax, VMAX All Flash, or VMAX3 systems even if softlinking support is not required by the user.
- Limits discovery of TimeFinder devices to specified sites and storage systems.
- Enforces selection of the gatekeeper address to be used by GDDR on TimeFinder commands for the listed storage system.

TFINDER parameters are ignored when running with BCV(N) or GLOBALS(CHECK). If running with GLOBALS(CHECK), GDDRACDD will fabricate appropriate TFINDER parameters, based on:

- Call Override bytes 12 (DC4), 18/19/20 (DC1/2/3)
- The Manage EXT BCV configuration option: PARMS.BCVTYPE.EXT=Y/N and associated call-override-like variable: USEBCV.EXT
- The Manage TEST BCV configuration option: PARMS.BCVSET.TEST=Y/N and associated call-override-like variable USEBCV.TEST.DC*n*
- The selected Legacy TF method: MFEOPTS.TFMETHOD.*site*=MIRR|CLON

Each TFINDER parameter must specify the following:

- A Site (or ALL) as the second word.
- The fourth word can be either the word NONE (excludes the site from BCV discovery) or the following subparameters:
  - MIRR or CLON as the selected Legacy TF method (optional)
  - SRDF group specification (required)
  - BCV set specification (GOLD and/or TEST) (required)
  - GK specification (optional)

If no TFINDER subparameters are specified, GDDRACDD will discover GOLD-set BCV devices only, for GDDR internal devices only, and use TF/Mirror as the Legacy TF method for all sites.

Examples:

- Manage GOLD-set internal BCV devices uniformly at all sites, but none at DC1. Issue warning-level messages if the GOLD-set is found to be incomplete. Use TF/Clone as the Legacy TF method for all sites:

```
TFINDER  ALL      GDDR CLON,RA(INT),GOLD(WARN)
TFINDER  DC1      GDDR NONE
```

- Manage BCV devices at DC3 only, discover both a GOLD set and a TEST set, for both internal and external devices, issue error messages if the GOLD set is incomplete, and warning messages if the test set is incomplete. Use TF/Mirror as the Legacy TF method.

```
TFINDER  DC3       GDDR RA(ALL),GOLD(ERROR),TEST(WARN)
```

The TFINDER parameter has subparameters listed further. Specifying no TFINDER subparameters has the same effect as use of the following default values:

```
TFINDER  ALL     GDDR MIRR,RA(INT),GOLD(WARN)
```

This statement means:

- Generate PARMS.STDBCV.<*t*>.DCn or PARMS.BCV.DCn.ONLY for internal SRDF groups for all systems at all sites.
- Use the first GDDRPARM SYMM pool gatekeeper for each system.

- Issue warning messages if any internal devices are found not to have a BCV associated.
- Use TF/Mirror as the Legacy TF method on VMAX 10K, 20K, 40K systems at all sites (unless SnapVX with softlinking is required, in which case TF/Clone is enforced on VMAX 10K, 20K, 40K systems).

DC*n*|ALL

This required subparameter limits BCV discovery to the sites listed on TFINDER parameters. If TFINDER parameters are coded with 'ALL' sites, then any other TFINDER parameters not specifying 'ALL' site selection must specify the SRDF group as NONE.

MIRR|CLON

This optional parameter defines the legacy TimeFinder method. The default value is MIRR (for TF/Mirror).

GDDR supports TF/Mirror and SnapVX on PowerMax, VMAX All Flash, or VMAX3 systems, as well as the legacy local replication methods TF/Mirror and TF/Clone on VMAX 10K, 20K, 40K systems.

The specification of a legacy TF method affects GDDR behavior during scripts as follows:

– If SnapVX softlinking support is required, then that method will be used on PowerMax, VMAX All Flash, or VMAX3 ystems, while TF/Clone will be used on VMAX 10K, 20K, 40K systems, ignoring any specification of a Legacy TF method.

– If SnapVX softlinking is not required, the behavior is determined by the specification for the Legacy TF method.

  If the Legacy TF method is MIRR, GDDR issues TF/Mirror commands across all PowerMax/VMAX hardware generations. TimeFinder software translates these to SnapVX commands using hard linking on PowerMax, VMAX All Flash, or VMAX3 systems.

  If the Legacy TF method is CLON, GDDR issues TF/Clone commands on VMAX 10K, 20K, 40K systems and SnapVX commands with softlinking on PowerMax, VMAX All Flash, or VMAX3 systems.

RA({INT|EXT|ALL})|RA(srdfgrp,srdfgrp,...)|NONE

This required subparameter selects SRDF group or groups.

It limits the BCV set selection to the GDDR-managed SRDF groups specified, either by type or by SRDF group number.

**Note:** It is not recommended to use the RA sub-parameter to restrict BCV discovery to certain SRDF groups in a system. This capability is provided for lab backwards compatibility with traditional GDDR behavior.

Possible values are as follows:

– NONE: excludes the site from BCV discovery

- `RA(INT|EXT|ALL)` : discover BCVs for internal, external, or all GDDR-managed SRDF groups.
- `RA(srdfgrp,srdfgrp,...)`: lists SRDF group numbers.

The SRDF group selection can only be specified with an explicit list of SRDF groups, if there is only 1 GK( ) coded, and the site is not specified as ALL.

You can use RA(INT|EXT|ALL) parameters to set defaults for a site or a specific system, along with parameters for specific groups for which you want a different BCV discovery.

`GOLD({WARN|ERROR}),TEST({WARN|ERROR})`

This required subparameter indicates which sets of BCVs GDDRACDD should discover and sets the BCV protection validation messaging level.

**Note:** "GOLD and TEST sets of BCVs" on page 75 describes the BCV sets.

This subparameter is required unless the SRDF group specification is NONE.

During discovery of the configuration, GDDRACDD validates that ALL devices in selected SRDF groups have an associated BCV in the sets selected here. If exceptions are found, GDDRACDD will issue either warning or error level messages, depending on the specification in this subparameter.

Either the GOLD set or the TEST set (or both) must be selected. If both GOLD(...) and TEST(...) are specified:

- If ACDD finds 2 BCVs associated with a STD device, then the low BCV device number goes in the GOLD set and the high BCV number goes in the TEST set.
- If only 1 BCV is found for a STD device, it goes in the GOLD set and there is a W or E-level message for the Test set.

If only GOLD(...) or TEST(...) is specified, then if more than 1 BCV is found associated with a STD device, the lowest BCV device number goes in the selected set.

`GK(gk,gk,...)`

This optional subparameter specifies serves two purposes:

- Limits BCV discovery at the affected site to the storage systems identified by the listed gatekeepers.
- Forces GDDRACDD to use the listed gatekeeper for all BCV parameters defining BCVs in the affected storage system. If no GK subparameter is specified, GDDRACDD uses the first gatekeeper from the pool defined in the GDDRPARM file for the affected storage systems.

By default, no selected gatekeepers are defined, and GDDRACDD selects all storage systems at the specified site.

If more than one gatekeeper is listed, then the RA subparameter must be specified as INT, EXT or ALL. You can include as many TFINDER parameters for the same site as needed to list all required gatekeepers.

**Note:** It is not recommended to use the GK subparameter to restrict BCV discovery to certain storage systems at a site. This possibility is provided for backwards compatibility with traditional GDDR behavior.

**Global variables for BCV(Y) discovery**

When GDDRACDD runs in Discovery mode and BCV(Y), it prints global variables to define the discovered BCV devices to GDDR in the E05TFDEV dataset. The type of variables printed by GDDRACDD depends on how it is invoked.

When running with BCV(Y), GDDRACDD writes a new global variable type to E05TFDEV:

```
PARMS.STDBCV.t.DCn.n=gk,bcvl-bcvh,stdl-stdh,h
```

Where *t* can have one of the following values:

- 1 = GOLD set BCV devices associated with GDDR-managed internal devices
- 2 = GOLD set BCV devices associated with GDDR-managed external devices
- 3 = TEST set BCV devices associated with GDDR-managed internal devices
- 4 = TEST set BCV devices associated with GDDR-managed external devices

*stdl-stdh* is the GDDR-managed STD device range associated with the listed BCV range.

*h* is an indication of the host type: C for CKD, F for FBA.

VGROUP

The VGROUP parameter defines a virtual SRDF group to GDDRACDD. For example:

```
VGROUP   50       GDDR 123456789012,123456789013
```

These virtual groups have no meaning outside of GDDRACDD, and their use is limited to situations where GDDRACDD needs to run in an incomplete configuration.

Virtual groups are allowed in 4-site configurations with the same caveats as noted for the GENERATE parameter.

Virtual groups can only be referenced on GENERATE parameters.

Each VGROUP parameter specifies an SRDF group number, and a pair of storage systems to be considered connected by the specified SRDF group number. The group cannot exist in either of the storage systems. The virtual group must connect either DC1-DC4 or DC2-DC3.

Consider the following example for an SRDF/SQAR with AutoSwap configuration:

```
CSYSSITE O016     GDDR DC1
CSYSSITE VC1A     GDDR DC2
CSYSSITE O01E     GDDR DC3
CSYSSITE O01F     GDDR DC4
SYMM     DC1(H)   GDDR 123456789011,1000-100F
SYMM     DC2      GDDR 123456789011,1000-100F
SYMM     DC2(H)   GDDR 123456789022,B211-B212,B214-B216,B218
SYMM     DC2(H)   GDDR 123456789022,B21C-B21D
SYMM     DC1      GDDR 123456789022,B211-B212,B214-B216,B218
```

```
SYMM      DC1       GDDR 123456789022,B21C-B21D
SYMM      DC3(H)    GDDR 123456789033,5000-500F
SYMM      DC4(H)    GDDR 123456789044,8000-800F
INCLUDE   DC1       GDDR I,1008,12,13
INCLUDE   DC2       GDDR I,B214,24
INCLUDE   DC3       GDDR I,500E,34
```

GDDRACDD will be allowed to run on systems named O016, VC1A, O01E and O01F. If GDDRACDD detects it is running on VC1A, it knows it is running at DC2.

Four storage systems are defined for discovery, one at each site. The storage system at DC2 has two SYMM parameters to fit the gatekeeper pool specification. Among the SYMM parameters without the (H) Home indicator, only those referring to the site where it is running are used by GDDRACDD.

For all the SRDF groups found in these four storage systems, you only want to generate RDF.DEVICES parameters for the devices in groups 12 and 13 in the system at DC1, group 24 in the system at DC2 and group 34 in the system at DC3. Referring to Figure 126 on page 431, group 12 is automatically included at DC2, group 13 is automatically included at DC3, and groups 24 and 34 are automatically included at DC4.

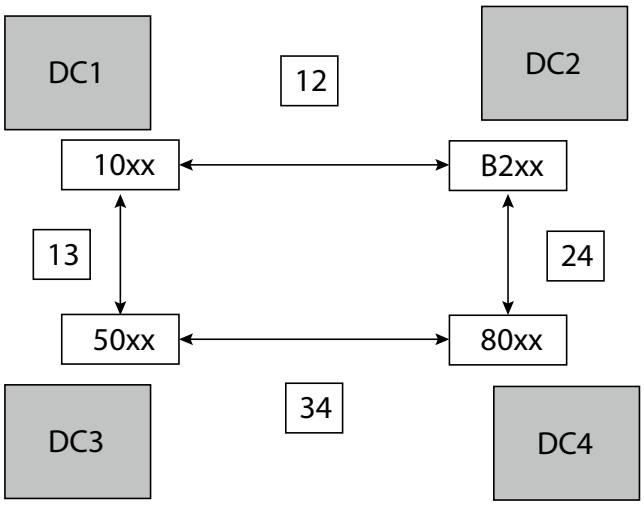



**Figure 126** SRDF/SQAR GDDRACDD parameter example

No external RDF.DEVICES parameters will be generated.

GDDRACDD will automatically select a discovery path for all storage systems.

If GDDRMAIN is running and has the expected CSYSSITE and SYMM parameters, and COMP(Y) is not specified in the arguments, the minimal required ACDDPARM deck for this configuration would be as follows:

```
INCLUDE   DC1       GDDR I,1008,12,13
INCLUDE   DC2       GDDR I,B214,24
INCLUDE   DC3       GDDR I,500E,34
```

The same result would be achieved with:

```
INCLUDE   DC2       GDDR I,B211,12
INCLUDE   DC3       GDDR I,5005,13
INCLUDE   DC4       GDDR I,8007,34,24
```

## Output

All messages in JESMSGLG are also visible in SYSTSPRT. In the sample message text below, RDF.DEVICES can be replaced by DLM.DEVICES in relevant cases.

**Note:** Depending on the Mainframe Enablers version, GDDRACDD displays PowerMax/VMAX device numbers and SRDF groups either as 4/2 digits (Mainframe Enablers V7.6) or 8/8 digits (Mainframe Enablers 8.0 and later). These displays are no indication of current or future limits on the number of PowerMax/VMAX devices and SRDF groups supported by Dell EMC. In this guide, the 4/2 notation is used.

### Information about arguments

GDDRACDD will always issue a message about the Range size and Alignment option used during each GDDRACDD run. The range size you see in the GDDRACDD output message information could be different from the one you specified in the run time arguments. If ALIGN is used, then range_size is rounded up to the next higher supported alignment boundary. If you specify ALIGN and a range size not supported with ALIGN, an additional informational message is issued:

```
GDDP415I GDD5 GDDRACDD using Range size 256, ALIGN, NOSIZE, NORAID,
and NOMETA
GDDP415I GDD5 Range size 32 overridden due to ALIGN
```

Depending on the presence/absence of the optional NOSIZE argument:

```
GDDP415I GDD5 GDDRACDD RDF.DEVICES parameters will span different
device sizes
GDDP415I GDD5 GDDRACDD RDF.DEVICES parameters will NOT span different
device sizes
```

Depending on the presence/absence of the optional NORAID argument:

```
GDDP415I GDD5 GDDRACDD RDF.DEVICES parameters will span different
protection types
GDDP415I GDD5 GDDRACDD RDF.DEVICES parameters will NOT span different
protection types
```

Depending on the presence/absence of the optional NOMETA argument:

```
GDDP415I GDD5 GDDRACDD RDF.DEVICES parameters will span meta and
non-meta devices
GDDP415I GDD5 GDDRACDD RDF.DEVICES parameters will NOT span meta and
non-meta devices
```

If COMP(Y) is specified:

```
GDDP415I GDD5 GDDRACDD running in Compatibility mode
```

If GLOBALS(CHECK) is specified:

```
GDDP415I GDD5 GDDRACDD will perform validation of existing
RDF.DEVICES parameters
```

Depending on whether GROUPS(BREAK) is specified:

```
GDDP415I GDD5 Device range breaks in one ra-group will [NOT] affect
device ranges in a second group
```

If FORCE is specified:

```
GDDP415I GDD5 RDF.DEVICES parms will be generated, even if
configuration errors are found
```

If NOSORT is specified:

```
GDDP415I GDD5 RDF.DEVICES parms will NOT be sorted for optimal
performance
```

If SCAN is specified:

```
GDDP415I GDD5 GDDRACDD will stop after validation of parameters
```

Depending on the CONFIG(*site*,*topology*) argument:

```
GDDP415I GDD5 Using Configuration description: site,topology
```

## Recognized ACDDPARM statements

Recognized ACDDPARM statements, information retrieved from the GDDRPARM file, global variables used for "validation mode", and discovery path information

All recognized ACDDPARM statements are echoed here, grouped by type. Verify that this matches the input deck you thought you specified. This section also shows ACDDPARM lines that were ignored (except comments).

**Note:** The example illustrates an SRDF/Star-A configuration.

```
GDDP415I GDD5 GDDMPARM:       Sitelist: DC1 DC3 DC4
GDDP415I GDD5 GDDMPARM:  and C-systems: LB01M3A LB01M34 LB01K145
GDDP415I GDD5
GDDP415I GDD5 GDDMPARM:  DASD sitelist: DC1 DC3 DC4
GDDP415I GDD5 Symmetrixes to be discovered
GDDP415I GDD5    Site  Symmetrix     Gatekeepers
GDDP415I GDD5    DC1   000197700135  19B0-19BF
GDDP415I GDD5    DC3   000195700866  3C90-3C9F
GDDP415I GDD5    DC4   000197200652  1371-137F
GDDP415I GDD5
GDDP415I GDD5 Selected Discovery paths
GDDP415I GDD5 Site Symmetrix    Path
GDDP415I GDD5  DC1 000197700135 19B0
GDDP415I GDD5  DC3 000195700866 19BF.D6
GDDP415I GDD5  DC4 000197200652 19BF.20
GDDP415I GDD5
GDDP415I GDD5 (SIM) ACDDPARM:  INCLUDE  DC1      GDD5 I,19B0,0000007D
GDDP415I GDD5 (SIM) ACDDPARM:  INCLUDE  DC3      GDD5 I,3C90,0000007D
GDDP415I GDD5 (SIM) ACDDPARM:  INCLUDE  DC1      GDD5 I,19B0,0000007A
...
GDDP415I GDD5 (SIM) ACDDPARM:  INCLUDE  DC4      GDD5 E,1371,0000008C
GDDP415I GDD5
GDDP415I GDD5 (SIM) ACDDPARM:  GENERATE DC4      GDD5
   E,1371,(0000008B,0000008C)
```

If GENERATE parameters are present for a 2-site configuration, a warning message is issued:

```
GDDP415W GDD5 GENERATE parameters ignored with 2-site configurations
```

If GDDRACDD runs with GLOBALS(CHECK), it echoes the relevant existing global variables:

**Note:** The example illustrates an SRDF/Star-A configuration.

```
GDDP415I GDD5 CONFIG: Dasd Sitelist = DC1 DC3 DC4
GDDP415I GDD5 CONFIG: C-Sitelist = DC1 DC3 DC4
GDDP415I GDD5 CONFIG: SRDFS=0
GDDP415I GDD5 CONFIG: AutoSwap=0
GDDP415I GDD5 CONFIG: SRDFA=1
GDDP415I GDD5 CONFIG: STAR=0
GDDP415I GDD5 CONFIG: SQAR=0
GDDP415I GDD5 CONFIG: STAR-A=1
GDDP415I GDD5 CONFIG: R22=1
GDDP415I GDD5 CONFIG: Master C-system = LB01M3A at DC1
GDDP415I GDD5 CONFIG: Primary DASD Site = DC1
GDDP415I GDD5 CONFIG: Tertiary DASD Site = DC3
GDDP415I GDD5 CONFIG: Quaternary DASD Site = DC4
GDDP415I GDD5 CONFIG: Current Topology = CONCURRENT
GDDP415I GDD5 CONFIG: Expecting Link-Block at DC4 (from DC3)
GDDP415I GDD5
GDDP415I GDD5 CONFIG: Expected SRDF Device Pairing:
GDDP415I GDD5 CONFIG: DC1-DC3: R1
GDDP415I GDD5 CONFIG: DC1-DC4: R1
GDDP415I GDD5 CONFIG: DC3-DC4: R1
GDDP415I GDD5 CONFIG: DC3-DC1: R2
GDDP415I GDD5 CONFIG: DC4-DC1: R2
GDDP415I GDD5 CONFIG: DC4-DC3: R2
GDDP415I GDD5
GDDP415I GDD5 CONFIG: Existing RDF.DEVICES parameters
GDDP415I GDD5         DC1-DC3
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC1.DC3.1=19B0,0000007D,000002F3-000002F7,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC1.DC3.2=19B1,0000007A,000005B3-000005C2,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC1.DC3.3=19B2,0000008A,00000293-000002A2,...
GDDP415I GDD5         DC3-DC1
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC3.DC1.1=3C90,0000007D,00000802-00000806,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC3.DC1.2=3C91,0000007A,000037FD-0000380C,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC3.DC1.3=3C92,0000008A,0000380D-0000381C,...
GDDP415I GDD5         DC1-DC4
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC1.DC4.1=19B0,0000007E,000002F3-000002F7,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC1.DC4.2=19B1,0000007B,000005B3-000005C2,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC1.DC4.3=19B2,0000008B,00000293-000002A2,...
GDDP415I GDD5         DC4-DC1
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC4.DC1.1=1371,0000007E,000002F3-000002F7,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC4.DC1.2=1372,0000007B,00000223-00000232,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC4.DC1.3=1373,0000008B,00000293-000002A2,...
GDDP415I GDD5         DC3-DC4
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC3.DC4.1=3C90,0000007F,00000802-00000806,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC3.DC4.2=3C91,0000007C,000037FD-0000380C,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC3.DC4.3=3C92,0000008C,0000380D-0000381C,...
GDDP415I GDD5         DC4-DC3
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC4.DC3.1=1371,0000007F,000002F3-000002F7,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC4.DC3.2=1372,0000007C,00000223-00000232,...
GLOBAL.GDD5.PARMS.RDF.DEVICES.DC4.DC3.3=1373,0000008C,00000293-000002A2,...
```

If invoked with BCV(Y) and GLOBALS(CHECK) and all relevant call override bytes are 0:

```
GDDP415W GDD5 CONFIG: Not verifying BCV parameters for any sites
```

If ACDD runs with GLOBALS(CHECK) in a configuration with SNAP-VX support configured:

```
GDDP415I GDD5 Supporting SNAP-VX Softlink at DC1: YES
```

```
GDDP415I GDD5 SNAP-VX snapshotname for GOLD set, Internal: G_I_SNAP
```

If GDDRACDD runs with GLOBALS(CHECK) it fabricates INCLUDE, GENERATE, TFINDER, and where applicable SOFTLINK and SNAPNAME parameters as necessary. Those are echoed to the joblog:

```
GDDP415I GDD5 (SIM) ACDDPARM:  INCLUDE  DC1      GDDR I,2000,10
GDDP415I GDD5 (SIM) ACDDPARM:  INCLUDE  DC1      GDDR I,2000,20
...
GDDP415I
GDDP415I GDD5 (SIM) ACDDPARM:  GENERATE DC3      GDDR I,A000,(20,30)
GDDP415I GDD5 (SIM) ACDDPARM:  GENERATE DC3      GDDR E,A000,(F9,31)
GDDP415I
GDDP415I GDD5 (SIM) ACDDPARM:  TFINDER  DC3      GDDR MIRR,RA(INT),GOLD(WARN)
GDDP415I
GDDP415I GDD5 (SIM) ACDDPARM:  SOFTLINK DC3      GDDR YES
GDDP415I
GDDP415I GDD5 (SIM) ACDDPARM:  SNAPNAME DC3      GDDR GOLD,INT,GDDR_INT_GOLD_SNAP
```

Several types of input errors are flagged by GDDP400E messages (RDF.DEVICES can be replaced by DLM.DEVICES where relevant). See GDDP400E message description in the *GDDR Message Guide* for information about specific errors.

```
GDDP400E Validation error: ,,,
```

### GDDRACDD site awareness

Based on the detected system name and the CSYSSITE parameters in ACDDPARM, GDDRACDD is able to determine which GDDR site it runs on, and issues the following message:

```
GDDP415I GDD5 GDDRACDD running at DC1
```

GDDP400E will be issued in one of two formats (depending on when the error is caught), if the current system does not match one of the system names listed in CSYSSITE parameters:

```
GDDP400E Validation error: Current system: VC1A not defined in
   csyslist: 0016 VC1B 001E
GDDP400E Validation error: GDDRACDD not supported at site-id (not a
   configured site)
```

### Wizard integration

If the JCL used to run GDDRACDD does not have an E04SRDFD or E05TFDEV DD statement, one or both of the following messages will be issued:

```
GDDP415I GDD5 E04SRDFD not allocated, RDF.DEVICES parameters will not
   be available to GDDR Parameter Wizard

GDDP415I GDD5 E05TFDEV not allocated, BCV.DCn.ONLY
parameters will not be available to GDDR Parameter Wizard
```

If the JCL used to run GDDRACDD has some or all of the optional DD-cards, but GDDRACDD fails to open one of them, the following message will be issued:

```
GDDP415W GDD5 Failed to open ddname (mode) with RC=<rc>
```

### Messages related to storage system discovery

PowerMax/VMAX device discovery is done one site at a time, one storage system at a time.

The GDDRACDD JESMSGLG shows progress messages as each storage system is being discovered:

```
GDDP402I GDD5 Beginning Discovery of Symmetrix units
GDDP410I GDD5 Site: DC1 - Device ranges in Symm: 000197700135 (5977) GK: 19B0
GDDP402I GDD5 Site: DC1 Symm: 000197700135 SP132 usage: 4022283
...
```

Message GDDP410I identifies the site-location of the discovered storage system, its serial number and operating environment level, as well as its discovery path from the site where GDDRACDD is running. It signals the start of discovery for the listed storage system.

Message GDDP402I shows the amount of virtual storage used for this storage system. It signals the completion of discovery for the listed storage system.

## Generated RDF.DEVICES and DLM.DEVICES parameters

The final section of the JESMSGLG is a complete listing of the RDF.DEVICES and DLM.DEVICES parameters generated for the discovered storage systems. The parameters are listed by site-pair, once in each direction.

```
GDDP419I GDD5 RDF.DEVICES Parameters for Discovered Symmetrixes
GDDP419I GDD5
GDDP419I GDD5 Site pair DC1-DC2
GDDP419I GDD5 PARMS.RDF.DEVICES.DC1.DC2.1
   =2000,10,009E-00CF,009E-00CF,10,6000,0  (     50 CS )
GDDP419I GDD5 PARMS.RDF.DEVICES.DC1.DC2.2
   =2001,10,00D4-015B,00D4-015B,10,6001,0  (    136 CS )
...
```

## Messages in SYSTSPRT

All messages visible in JESMSGLG are also echoed to SYSTSPRT. SYSTSPRT has some additional messages, not visible in JESMSGLG. The following sections discuss those elements in SYSTSPRT not visible in JESMSGLG.

## Initial device range discovery

Messages in this section could be used to piece together RDF.DEVICES or DLM.DEVICES parameters if the actual device parameter generation by GDDRACDD fails. The same information is reported in several stages of discovery.

### SRDF device pairs by site pair and by group as seen from either side

```
GDDR DC1-DC2 RDF device pairs in group 000292601023.10 as seen from DC1
009E-009E  ===>  009E-009E
009F-00CF  ===>  009F-00CF
…
GDDR DC1-DC2 RDF device pairs in group 000292601156.10 as seen from DC2
009E-00CF  <===  009E-00CF
00D4-0113  <===  00D4-0113

...
```

### Discovered ranges

```
Discovered RDF device ranges
RDF device ranges for site-pair DC1-DC2
   > 000292601023 Group: 10 ---> 000292601156
         1 009E-009E  <--->  009E-009E     1 (CS) (CS) Groups: (10   ) (10   )
         2 009F-00CF  <--->  009F-00CF    49 (CS) (CS) Groups: (10 20) (10   )
...
```

**Discovered and generated ranges**

```
Discovered and generated RDF device ranges
RDF device ranges for site-pair DC1-DC2
   > 000292601023 Group: 10 ---> 000292601156
         1 009E-009E  <--->  009E-009E     1 (CS) (CS) Groups: (10   ) (10   )
         2 009F-00CF  <--->  009F-00CF    49 (CS) (CS) Groups: (10 20) (10 30)
…
```

## RDF.DEVICES parameters

Messages in this section can used to piece together RDF.DEVICES parameters if the actual device parameter generation by GDDRACDD fails. The same information is reported in several stages of parameter generation (initial, sorted, and those with gatekeepers from pools). Each of these messages can be repeated for DLM.DEVICES parameters, if applicable.

```
Initial RDF.DEVICES Parameters
   Site pair DC1-DC2
PARMS.RDF.DEVICES.DC1.DC2.1 =2000,10,009E-00CF,009E-00CF,10,6000,0  (     50 CS )
PARMS.RDF.DEVICES.DC1.DC2.2 =2000,10,00D4-015B,00D4-015B,10,6000,0  (    136 CS )
```

## Messages in SYMMDEVS

If the SYMMDEVS DD card is present in the GDDRACDD JCL, it contains the device ranges report as created during discovery of the devices. It consists of a sequence of the following messages:

```
GDDP402I Beginning Discovery of Symmetrix units
GDDP410I Site: DC1 - Device ranges in Symm: 000192600975 (5876) GK: 0E40
GDDP402I Site: DC1 Symm: 000192600975 SP132 usage: 1769908
GDDP410I Site: DC1 - Device ranges in Symm: 000192600975 (5876) GK: 0E40
GDDP411I LDEV-HDEV-COUNT-HTM-MBR-  CYLS  -MR- RDF-RDFGROUPS
GDDP412I 0000-00C7-  200-SVA-----    5120---------/-
GDDP412I 00C8-00CB-    4-SFI-----    8738--1------/-
GDDP412I 00CC-00CC-    1-CS------   10017--5-R11--D1/CE---/-----/--W-/0003----/192604665/00CC-00CC
GDDP412I                                    D8/-----/---D-/--W-/--------/192604809/00CC-00CC
```

Message GDDP410I identifies the site-location of the discovered storage system, its serial number and operating environment level, as well as its discovery path from the site where GDDRACDD is running.

Message GDDP402I shows the amount of virtual storage used for this storage system.

Message GDDP411I is a header preceding the GDDP412I messages. For detailed explanation of fields in message GDDP412I, see the *GDDR Message Guide*.

## Messages in RDFGROUP

If the RDFGROUP DD card is present in the GDDRACDD JCL, it contains a summary report at the SRDF group level. The report is presented in two parts:

- Part 1: List of SRDF groups linking GDDR-managed storage systems
- Part 2: Unique GDDR SRDF groups by site-pair and type

Part 1 lists all SRDF groups between storage systems for which a SYMM parameter is found, organized by site-pair. For each group an indication is given of its inclusion type (Internal or External or None). If applicable, the report indicates that a group is for DLm backend devices ("- DLm"). Offline groups are marked as "<<< OFFLINE <<<". Virtual groups are marked "- VIRTUAL". Groups connecting same-site storage systems are marked "- ignored".

Part 1 example:

**Note:** The example illustrates an SRDF/Star-A configuration.

```
GDDP417I GDD5 List of RDFgroups linking GDDR Managed Symmetrixes
GDDP417I GDD5
GDDP417I GDD5   Symmetrix units at DC1
GDDP417I GDD5
GDDP417I GDD5     Site pair DC1-DC3
GDDP417I GDD5     19B0 000197700135 (5977) <-- 0000001A --> 000195700866 (5876) (3C90) (Non-GDDR)
GDDP417I GDD5     19B0 000197700135 (5977) <-- 0000004B --> 000195700866 (5876) (3C90) (Non-GDDR)
...
GDDP417I GDD5     Site pair DC1-DC4
GDDP417I GDD5     19B0 000197700135 (5977) <-- 00000020 --> 000197200652 (5978) (1371) (Non-GDDR)
GDDP417I GDD5     19B0 000197700135 (5977) <-- 00000021 --> 000197200652 (5978) (1371) (Non-GDDR)
...
GDDP417I GDD5   Symmetrix units at DC3
GDDP417I GDD5
GDDP417I GDD5     Site pair DC3-DC1
GDDP417I GDD5     3C90 000195700866 (5876) <-- 0000001A --> 000197700135 (5977) (19B0) (Non-GDDR)
GDDP417I GDD5     3C90 000195700866 (5876) <-- 0000004B --> 000197700135 (5977) (19B0) (Non-GDDR)
...
GDDP417I GDD5     Site pair DC3-DC4
GDDP417I GDD5     3C90 000195700866 (5876) <-- 0000007C --> 000197200652 (5978) (1371) (Internal)
GDDP417I GDD5     3C90 000195700866 (5876) <-- 0000007F --> 000197200652 (5978) (1371) (Internal)
GDDP417I GDD5
GDDP417I GDD5   Symmetrix units at DC4
GDDP417I GDD5
GDDP417I GDD5     Site pair DC4-DC1
GDDP417I GDD5     1371 000197200652 (5978) <-- 00000020 --> 000197700135 (5977) (19B0) (Non-GDDR)
GDDP417I GDD5     1371 000197200652 (5978) <-- 00000021 --> 000197700135 (5977) (19B0) (Non-GDDR)
...
GDDP417I GDD5     Site pair DC4-DC3
GDDP417I GDD5     1371 000197200652 (5978) <-- 0000007C --> 000195700866 (5876) (3C90) (Internal)
GDDP417I GDD5     1371 000197200652 (5978) <-- 0000007F --> 000195700866 (5876) (3C90) (Internal)
...
```

Part 2 lists all INCLUDE-d SRDF groups, first the internal groups and then the external groups. For each type, the list is organized by site-pair.

Part 2 example:

**Note:** The example illustrates an SRDF/Star-A configuration.

```
GDDP417I GDD5 Unique GDDR ra-groups by site-pair and type
GDDP417I GDD5
GDDP417I GDD5  RDF Devices
GDDP417I GDD5
GDDP417I GDD5  RA-groups of type Internal
GDDP417I GDD5
GDDP417I GDD5   DC1  GK     RA     GK DC3 (  DC1 Symm  -  DC3 Symm  )
GDDP417I GDD5       19B0 0000007D 3C90    (000197700135-000195700866)
GDDP417I GDD5       19B0 0000007A 3C90    (000197700135-000195700866)
GDDP417I GDD5
GDDP417I GDD5   DC1  GK     RA     GK DC4 (  DC1 Symm  -  DC4 Symm  )
GDDP417I GDD5       19B0 0000007E 1371    (000197700135-000197200652)
GDDP417I GDD5       19B0 0000007B 1371    (000197700135-000197200652)
GDDP417I GDD5
GDDP417I GDD5   DC3  GK     RA     GK DC4 (  DC3 Symm  -  DC4 Symm  )
GDDP417I GDD5       3C90 0000007F 1371    (000195700866-000197200652)
GDDP417I GDD5       3C90 0000007C 1371    (000195700866-000197200652)
GDDP417I GDD5
GDDP417I GDD5  RA-groups of type External
GDDP417I GDD5
GDDP417I GDD5   DC1  GK     RA     GK DC3 (  DC1 Symm  -  DC3 Symm  )
GDDP417I GDD5       19B0 0000008A 3C90    (000197700135-000195700866)
GDDP417I GDD5
GDDP417I GDD5   DC1  GK     RA     GK DC4 (  DC1 Symm  -  DC4 Symm  )
GDDP417I GDD5       19B0 0000008B 1371    (000197700135-000197200652)
GDDP417I GDD5
GDDP417I GDD5   DC3  GK     RA     GK DC4 (  DC3 Symm  -  DC4 Symm  )
GDDP417I GDD5       3C90 0000008C 1371    (000195700866-000197200652)
```

### Messages in CREPAIR

If the CREPAIR DD card is present in the GDDRACDD JCL, it contains a set of CREATEPAIR commands which could be used the reconstruct the GDDR-managed environment from scratch. The content of the report is influenced by the CONFIG(...) argument. For example:

**Note:** The example illustrates an SRDF/Star-A configuration.

```
GDDP418I GDD5 Createpair commands by site-pair and type
GDDP418I GDD5
GDDP418I GDD5  >>> Intended configuration: 3-site, CONCURRENT, Primary DASD site: DC1
GDDP418I GDD5
GDDP418I GDD5  RDF Devices
GDDP418I GDD5
GDDP418I GDD5  RA-groups of type Internal
GDDP418I GDD5
GDDP418I GDD5   Sitepair DC1-DC3
GDDP418I GDD5    Use these commands from a host with channel access to the Symmetrix unit at DC1
GDDP418I GDD5    Caution: use LCLISR2 if you want the devices at DC1 to be R2 devices
GDDP418I GDD5 SC VOL,LCL(19B0,0000007D),CREATEPAIR(ADCOPY-DISK,SUSPEN-
GDDP418I D,STAR-A,LCLISR1),000002F3-000002F7,00000802
GDDP418I GDD5 SC VOL,LCL(19B1,0000007A),CREATEPAIR(ADCOPY-DISK,SUSPEN-
GDDP418I D,STAR-A,LCLISR1),000005B3-000005C2,000037FD
GDDP418I GDD5
GDDP418I GDD5   Sitepair DC3-DC1
...
```

### Messages in FBAMETA

If the FBAMETA DD card is present in the GDDRACDD JCL, it contains an overview of GDDR-managed FBA meta devices. It shows meta-head versus meta-member relations as well as SRDF device pairing information for FBA meta devices. For example:

```
GDDP422I
GDDP422I FBA Meta Devices in GDDR managed ra-groups
GDDP422I
GDDP422I  RA-groups of type Internal
GDDP422I
GDDP422I   Site: DC1
GDDP422I     Symmetrix: 000292601023
GDDP422I             Head: 0220 Members: 0224 0221 0222
GDDP422I   Group: 10       0220          0221 0222 0223
GDDP422I   Group: 20       0220          0221 0222 0223
GDDP422I             Head: 0223 Members: 0226 0225 0227
GDDP422I   Group: 10       0224          0225 0226 0227
...
```

### Messages in TFLIST

If the TFLIST dd-card is present in the GDDRACDD JCL, it contains an overview of GDDR-managed BCV devices. It shows BCV to STD pairing information, as well as information pertaining to the requested BCV types and sets. For example:

```
GDDP460I GDD5 Site: DC1 - GDDR BCV Device ranges in Symmetrix: 000195700866 (5876) GK: 19BF.D6
GDDP461I GDD5    BCV  -   STD  -  LBCV  -H-  STATUS -SDDF-INT-EXT-GOLD-TEST-IMPLEMENTATION
GDDP462I GDD5 00000780-000037FD-00000780-C-SPLIT    -25FD- Y - N -WARN-    -CLON
GDDP462I GDD5 00000781-000037FE-00000781-C-SPLIT    -25FD- Y - N -WARN-    -CLON
```

If SnapVX is used, the GDDP461I and GDDP462I messages look similar to the following:

```
GDDP460I GDD5 Site: DC1 - GDDR BCV Device ranges in Symmetrix: 000197700135 (5977) GK: 19B0
GDDP461I GDD5    BCV  -   STD  -LAST SNAPSHOT NAME            -H-  STATUS -  SNID  -INT-EXT-GOLD-TEST-IMPLEMENTATION
GDDP462I GDD5 0000029A-0000029A-ADAD0001                     -C-ACTIVATE -ADAD0001- N - Y -WARN-WARN-SNVX: GDDR_EXT_STARA

GDDP462I GDD5 0000029B-0000029B-ADAD0001                     -C-ACTIVATE -ADAD0001- N - Y -WARN-WARN-SNVX: GDDR_EXT_STARA
```

```
GDDP462I GDD5 0000029C-0000029C-ADAD0001                -C-ACTIVATE -ADAD0001- N - Y -WARN-WARN-SNVX: GDDR_EXT_STARA
```

**Note:** For an explanation of message fields, see the *GDDR Message Guide*.

## Exception reporting

### Asymmetrical configurations

If GDDRACDD discovers asymmetry in the PowerMax/VMAX device numbers and the ALIGN option is specified, an warning message is issued:

```
GDDP415W
GDDP415W ALIGN was specified or defaulted, but an asymmetrical device
         configuration was discovered
GDDP415W Generated RDF.DEVICES parameters may not be aligned
GDDP415W
```

This message is issued to explain why generated parameters may not be aligned, which may be unexpected if you requested them to be aligned. It is of no further consequence.

### Configuration validation

Configuration errors are flagged with message GDDP420E. Some of these cause RC=8 and others cause a higher return code. For the list of errors, see description of message GDDP420E in the *GDDR Message Guide*.

**Note:** Items flagged with RC=8 may lead to the generation of RDF.DEVICES parameters, if GDDRACDD is run with the FORCE argument. There is no guarantee of any kind that the generated parameters are usable or even correct.

```
GDDP420E Error: ...
```

If running with BCV(Y) and one or more TFINDER parameters specify GOLD(ERROR) or TEST(ERROR):

```
Site: site-id Symm: serial Device symmdev requires n BCV devices. Found
   {NONE | n (bcv#)}
```

The following configuration issues are flagged with message GDDP421W, and cause RC=4. These messages should be reviewed as they could either result in incorrect RDF.DEVICES parameters or cause issues during GDDR scripts.

```
GDDP421W Warning: text
```

*text* can be one of the following:

- `Site: site-id Symm: serial Meta-member: symmdev has different RDF-mirror flags than head symmdev (<mbr-flags versus <head-flags)`
- `Site: site-id Symm: serial Meta-member: symmdev has different ADCOPY flag than head symmdev (<mbr-flag versus <head-flag)`
- `Site: site-id Symm: serial Meta-member: symmdev has different CG flags than head symmdev (<mbr-flags versus <head-flags)`
- `Site: site-id Symm: serial GK: gk RA: srdfgrp. NON-GDDR group srdfgrp found for GDDR-managed device range dev1-dev2`

If running with BCV(Y) and one or more TFINDER parameters specify GOLD(WARN) or TEST(WARN):

- Site: *site-id* Symm: *serial* Device *symmdev* requires *n* BCV devices. Found {NONE | *n* (*bcv#*)}

The following configuration issues are flagged with message GDDP421W, and cause RC=0. These messages occur when running with BCV(Y) and should be reviewed as they could either result in incorrect BCV-related parameters or cause issues during GDDR scripts. These messages are prefixed with the selected TF method for the affected storage system (MIRR, CLON, or SNVX).

GDDP421W Warning: *text*

*text* can be one of the following:

- BCV Parameter *value* defines non-existing device *dev#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines special device *dev#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines non-BCV device *dev#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines BCV *bcv#*, which has state INVALID in Symmetrix *serial* at *site*
- BCV Parameter *value* defines BCV *bcv#*, which was never attached in Symmetrix *serial* at *site*
- BCV Parameter *value* defines BCV *bcv#*, which is associated with non-existing STD device *std#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines BCV *bcv#*, which is associated with special STD device *std#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines BCV *bcv#*, which is associated with CKD Meta member STD device *std#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines BCV *bcv#*, which is associated with non-GDDR STD device *std#* in Symmetrix *serial* at *site*
- BCV Parameter *value* defines superfluous BCV *bcv#*, which is associated with STD device *std#* in Symmetrix *serial* at *site*. No BCV devices requested for STD device *std#*
- BCV Parameter *value* defines superfluous BCV *bcv#*, which is associated with STD device *std#* in Symmetrix *serial* at *site*. Requested BCV devices for STD device *std#*: *nn* (*bcv#*)
- No BCV Parameter for device *bcv#*, which is associated with GDDR STD device *std#* in Symmetrix *serial* at *site*
- BCV Parameter <value> associates BCV device *bcv#* with STD device *std#*, but this BCV is found associated with device *dev#* in Symmetrix *serial* at *site*
- Duplicate definition of BCV device *dev#* in Symmetrix *serial* at *site* in parameters *prm1* and *prm2*
- *pair-state* error detected for device *dev#* | STD-BCV pair *std#-bcv#* in Symmetrix *serial* at *site*
- Incorrect state *state* for target device *dev#* in Symmetrix *serial* at *site*: expected one of: *states*
- Device *bcv#* in Symmetrix *serial* at *site* should not be SNVX target for device *std#*: both are GDDR managed RDF devices
- Device *dev#* in Symmetrix *serial* at *site* should not use snapshot name *name*: reserved for GDDR
- Incorrect snapshot name *name* for target device *bcv#* in Symmetrix *serial* at *site*: should begin with *prefix*
- BCV device *bcv#* was found local replication source for GDDR managed RDF devices *std#1 std#2* in Symmetrix *serial* at *site*. Ignoring this BCV

# GDDR Environment Check utility (GDDRECHK)

The GDDR Environment Check utility performs the following actions:

- Check that GDDR modules are accessible and properly authorized.
- Check that all libraries in the GDDRPROC STEPLIB concatenation are APF-authorized.
- Check access to various RACF resources to ensure those resources were properly defined.

  **Note:** The job runs tests using the RACF logon ID from which the job was submitted.

- Validate GDDRPROC maintenance level against that of GDDRMAIN, and show the GDDR module maintenance level report. This can be used to determine if there are any incorrect libraries in the GDDRPROC STEPLIB concatenation.

## RACF security verification

GDDRECHK provides a means for the RACF administrator to verify that the site's security rules are properly defined for support of GDDR processing.

The utility validates authorizations using a control statement specifying group (GDDR$STC, GDDR$ADM, and GDDR$USR), site, CPC, and LPAR. The GDDRECHK job validates authorization of the user associated with the job against the RACF definitions on the system where the job is run. Therefore, the GDDRECHK job must be submitted on each C-system and managed system. It is recommended to use the USER job card parameter with the GDDRECHK job when the security administrator validates authorization for GDDR administrators and operations personnel.

**Note:** Table 15 on page 136 lists the members provided for use in defining and validating security authorization.

The GDDRECHK utility checks that the security rules (RACF/ACF2) are set up correctly for a given security ID (specified on the job card USER= parameter, or defaulted to the TSO user submitting the job) and a given GDDR security role (specified in the PARM='GROUP(GDDR$xxx)' field). GDDRECHK needs to be run separately for every combination of GROUP(GDDR$xxx) and user ID, to verify all security has been implemented correctly.

## Running GDDRECHK utility

To run the GDDRECHK utility:

1. Customize the job card to match your site requirements. If the verification is to be performed for a user ID other than the submitter, include the job card parameter USER= to specify the required user ID.
2. Customize the JCLLIB statement to specify the library containing your customized GDDRPROC.
3. Specify a value for GRP= from those listed in Table 12, "RACF permissions," on page 131.

For example:

- GRP=GDDR$ADM for GDDR administrators
- GRP=GDDR$USR for GDDR operations users
- GRP=GDDR$STC for GDDR started tasks
- GRP=GDDR$REV for GDDR review users (optional)
- GRP= for non-GDDR users to check all that GDDR resources have been properly protected.

4. Specify the CP1= value that was used from SAMPLIB(GDDIRACF), or specify a list of CPC names.

   **Note:** This step is not applicable to managed systems.

5. Specify the CP2= value that was used from SAMPLIB(GDDIRACF), or specify a list of CPC names.

   **Note:** This step is not applicable to managed systems.

6. Specify the CP3= value that was used from SAMPLIB(GDDIRACF), or specify a list of CPC names.

   **Note:** This step is not applicable to managed systems.

7. Specify the SYS= value that was used from SAMPLIB(GDDIRACF), or specify a list of system names.

   **Note:** This step is not applicable to managed systems.

8. Run the GDDRECHK job on each C-system and managed system to verify RACF definitions.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRECHK).

```
//GDDRECHK JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//         SET GRP=GDDR$ADM    <--- GDDR$ADM|GDDR$USR|GDDR$STC|GDDR$REV
//         SET CP1=CPC1        <--- CPC names to check at DC1
//         SET CP2=CPC2        <--- CPC names to check at DC2
//         SET CP3=CPC3        <--- CPC names to check at DC3
//         SET SYS=VC1B        <--- System names to check
//*
//GDDRECHK EXEC GDDRPROC,PARM='GDDRECHK'
//SYSIN    DD  *
AUTHCMD EHCMSCME
AUTHCMD EHCMSCM6
AUTHCMD EHCMSCM9
AUTHCMD SCFRDFME
AUTHCMD SCFRDFM6
AUTHCMD SCFRDFM9
```

```
AUTHPGM ECGUTIL
AUTHPGM EHCMSCME
AUTHPGM EHCMSCM6
AUTHPGM EHCMSCM9
AUTHPGM EMCTF
AUTHPGM EMCSNAP
AUTHPGM SCFRDFME
AUTHPGM SCFRDFM6
AUTHPGM SCFRDFM9

LOADAPF EHCMSCMA
LOADAPF EHCMSCMB
LOADAPF EHCMSCMC
LOADAPF EHCMSCMD
LOADAPF EHCMSCME
LOADAPF EHCMSCMR
LOADAPF EHCMSCM6
LOADAPF EHCMSCM8
LOADAPF EHCMSCM9

AUTHPGM GDDFLIST
AUTHPGM GDDRCGTP
AUTHPGM GDDRDAP1
AUTHPGM GDDRDAP3
AUTHPGM GDDRQFCN
AUTHPGM GDDRQRY5
AUTHPGM GDDRSSVI
AUTHPGM GDDRSTAT
AUTHPGM GDDRSTOK
AUTHPGM GDDRXCMD

AUTHTSF GDDBCPC2
AUTHTSF GDDBCPC3
AUTHTSF GDDBCPD2
AUTHTSF GDDBCPL2
AUTHTSF GDDBCPQ2
AUTHTSF GDDBCPS2
AUTHTSF GDDRAWTZ
AUTHTSF GDDRINF2
AUTHTSF GDDRMCS1
AUTHTSF GDDRQDE2
AUTHTSF GDDRQDE3
AUTHTSF GDDRQDE4
AUTHTSF GDDRQDE5
AUTHTSF GDDRQDE6
AUTHTSF GDDRQDE7
AUTHTSF GDDRQDE8
AUTHTSF GDDRQDE9
AUTHTSF GDDRQDEA
AUTHTSF GDDRQDEB
AUTHTSF GDDRQDEC
AUTHTSF GDDRQDED
AUTHTSF GDDRQDEE
AUTHTSF GDDRQDEF
AUTHTSF GDDRQDEG
AUTHTSF GDDRXMPS
/*
//*
//*********************************************************************
//* Check that all libraries in the GDDRPROC STEPLIB concatenation    *
//* are APF authorized                                                *
//*********************************************************************
//*
//GDDCAUTH EXEC GDDRPROC,PARM='CALL *(GDDCAUTH) ''VALIDATE'''
//*
```

```
//*******************************************************************
//* Check access to various RACF resources to ensure those resources  *
//* were properly defined                                             *
//*******************************************************************
//*
//GDDTSECR EXEC GDDRPROC,
// PARM='GDDTSECR GRP(&GRP) CP1(&CP1) CP2(&CP2) CP3(&CP3) SYS(&SYS)'
//*
//*******************************************************************
//* Validate GDDRPROC maintenance level against that of GDDRMAIN,     *
//* and display GDDR module maintenance level report                  *
//*******************************************************************
//*
//GDDRMCMD EXEC GDDRPROC,PARM='CALL *(GDDRMCMD) ''MAINT'''
//*
```

## SYSIN syntax

The syntax is as follows:

{AUTHCMD|AUTHPGM|AUTHTSF|LOADAPF|LOADNAPF} *module*

Where:

AUTHCMD

Checks the AUTHCMD list and performs the LOADAPF function for the specified module.

AUTHPGM

Checks the AUTHPGM list and performs the LOADAPF function for the specified module.

AUTHTSF

Checks the AUTHTSF list and performs the LOADAPF function for the specified module.

LOADAPF

Loads the module and checks APF authorization.

LOADNAPF

Loads the module but does not check APF authorization.

*module*

The name of the module to be loaded or checked.

## Return codes

The return codes are as follows:

| | |
|---|---|
| 0 | All checks were successful |
| 4 | Warning, one or more resources were allowed that should have been denied |
| 8 | Error, one or more resources were not protected that should have been |
| 12 | Error, access was denied to one or more resources that should be allowed, or a module was not authorized, not in IKJTSO, or load failed |
| 16 | Abend occurred, SVC dump is taken |

## Sample output

GDDRECHK produces a listing which shows the result of GDDR SAF authorizations by OPERCMDS, SURROGAT, TSOAUTH, JESJOBS, and FACILITY classes.

### Output fields

The following fields are displayed in the output:

CLASS — Specifies a valid class name defined in the RACF class descriptor table associated with the `RESOURCE NAME`. The GDDRECHK utility expects `DATASET` or `FACILITY`.

RESOURCE NAME — For a particular GDDR security role, shown in the report at the top on the line `Testing GDDR$xxx group access`, this column specifies the discrete or generic DATASET or FACILITY profile name to be validated.

ACCESS — Shows what access the user should have (`READ`, `UPDATE`, `NONE`).

GDDSECUR RESULT — Shows the response from the security system (a SAF call) to an inquiry to check the indicated `ACCESS` for the indicated resource (`FACILITY` and `RESOURCE NAME`). This can be `ALLOW`, `DENY`, or `RESOURCE NOT PROTECTED`.

COMMENT — Shows `Ok` if the access is correct (security system correctly allowed or denied access), or `*** ERROR ***` if the access is not correct (the user has too much or too little access).

- If the user has too much access, `*** ERROR ***` will be reported when either:
  - The resource is not protected at all.
  - The particular user ID has more privileges to access the indicated resource than the user should have (defined by the value specified on the PARM='GROUP(GDDR$*xxx*)' field).
- If the user has too little access, `*** ERROR ***` will be reported when the security system returns DENY (for ACCESS=READ or ACCESS=WRITE).

An `*** ERROR ***` is always returned when a resource that should be protected is not protected by your security system (security system returns: `RESOURCE NOT PROTECTED`).

All lines flagged with `*** ERROR ***` should be investigated and fixed, to get a clean report with all `Ok` values in the `COMMENT` column.

# GDDR SRDF Director Overview utility (GDDRDIRS)

The GDDR SRDF Director Overview utility (GDDRDIRS) provides an overview of the currently configured directors used for SRDF in storage systems known to GDDR. The utility can be run at any site with a GDDR configuration.

By default, the utility provides SRDF director status for all directors used by all SRDF groups in storage systems defined to GDDR.

Use of the optional GDDR argument limits the report output to only the directors and SRDF groups defined in GDDR RDF.DEVICES parameters (this includes the consistency protected SRDF groups and the external SRDF groups).

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRDIRS).

```
//GDDRDIRS JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRDIRS EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRDIRS arguments
/*
```

## Arguments

GDDR

This optional argument shows only directors and SRDF groups defined in GDDR RDF.DEVICES parameters.

## Sample output

The output samples have been edited for brevity. They include:

- GDDR SRDF directors overview for all SRDF groups (Figure 127 on page 448)
- SRDF groups by director by storage system by site (Figure 128 on page 448)
- SRDF groups by director by storage system by site, Site UNK (Figure 129 on page 449)

```
+GDDP430I GDDR SRDF Directors overview for ALL ra-groups (MFE730) running at DC2
+GDDP432I
+GDDP431I (I) DC1 1072 000190103389 (5773) 81 <-ONL-> 81 000190101982 (5073) 5072 DC2
+GDDP432I         28 (F) ONL(Y) CONN(Y)                  28 (F) ONL(Y) CONN(Y)
+GDDP432I         38 (F) ONL(Y) CONN(Y)                  38 (F) ONL(Y) CONN(Y)
+GDDP432I
+GDDP431I (I) DC1 ABC1 000192601301 (5874) 80 <-ONL-> 80 000192601314 (5074) B27A DC2
+GDDP432I         48 (F) ONL(Y) CONN(Y)                  48 (F) ONL(Y) CONN(Y)
+GDDP432I         49 (F) ONL(Y) CONN(Y)                  49 (F) ONL(Y) CONN(N)        *
+GDDP432I         58 (F) ONL(Y) CONN(Y)                  58 (F) ONL(Y) CONN(Y)
+GDDP432I         59 (F) ONL(Y) CONN(Y)                  59 (F) ONL(Y) CONN(Y)
+GDDP432I
+GDDP431I (I) DC1 1073 000190103389 (5773) 82 <-ONL-> 82 000190100868 (5773) 3072 DC3
+GDDP432I         03 (E) ONL(Y) CONN(Y)                  03 (E) ONL(Y) CONN(Y)
+GDDP432I         0E (E) ONL(Y) CONN(Y)                  3E (E) ONL(Y) CONN(Y)
+GDDP431I (E) DC1 1073 000190103389 (5773) 88 <-ONL-> 88 000190100868 (5773) 3073 DC3
+GDDP432I         03 (E) ONL(Y) CONN(Y)                  03 (E) ONL(Y) CONN(Y)
+GDDP432I         0E (E) ONL(Y) CONN(Y)                  3E (E) ONL(Y) CONN(Y)
+GDDP432I
...
```

**Figure 127** GDDR SRDF directors overview for all SRDF groups

```
...
+GDDP435I Site: DC1
+GDDP436I  Symmetrix: 000190103389 (5773) - GK: 1072 Path: 5072.01
+GDDP437I   Director: 28 (F) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I) 81
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 01 05 10 11 15 16 18 19 1A 3C 3D 3E 3F 40 4C 5A 70 80 B1 E1
+GDDP437I   Director: 38 (F) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I) 81
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 01 05 10 11 15 16 18 19 1A 40 4C 70 80 B1 E1 F1 F4
+GDDP437I   Director: 03 (E) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I) 82
+GDDP438I    RA-groups: (E) 88
+GDDP438I    RA-groups: (N) 02 04 06 1B 20 21 25 36 B3 E2 E8 F2 F6
+GDDP437I   Director: 0E (E) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I) 82
+GDDP438I    RA-groups: (E) 88
+GDDP438I    RA-groups: (N) 02 04 06 1B 20 21 25 36 B3 E2 E8 F2 F6
+GDDP437I   Director: 2E (E) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I)
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 02 06 20 21 5B E2
+GDDP437I   Director: 13 (E) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I)
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 20 21
+GDDP437I   Director: 33 (E) ONL(Y) CONN(N)   *
+GDDP438I    RA-groups: (I)
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 21
+GDDP438I
+GDDP436I  Symmetrix: 000192601301 (5874) - GK: ABC1 Path: B27A.00
+GDDP437I   Director: 48 (F) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I) 80
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 00 03 10 15 1A 1F 25 2A 41 60 70 A0 B1 C1 D0 E4 F0 F4 05 5A
+GDDP437I   Director: 49 (F) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I) 80
+GDDP438I    RA-groups: (E)
...
```

**Figure 128** SRDF groups by director by storage system by site, Site: DC1

```
...
+GDDP438I
+GDDP438I
+GDDP435I Site: UNK
+GDDP436I  Symmetrix: 000187940471 (5671) - GK: ---- Path: 5072.01.03
+GDDP437I   Director: 31 (F) ONL(Y) CONN(Y)
+GDDP438I    RA-groups: (I)
+GDDP438I    RA-groups: (E)
+GDDP438I    RA-groups: (N) 03
+GDDP438I
+GDDP436I  Symmetrix: 000187720450 (----) - GK: ---- Path: No path found
+GDDP438I
+GDDP436I  Symmetrix: 000190103189 (----) - GK: ---- Path: No path found
+GDDP438I
+GDDP438I
...
```

**Figure 129** SRDF groups by director by storage system by site, Site UNK

# GDDR MSC Configuration Validation and Cleanup utility (GDDMSCFX)

The GDDMSCFX utility validates and cleans up MSC configuration information, such as the MSC scratch area, multibox list, and other structures stored in a storage system.

GDDMSCFX provides the VALIDATE, CLEAR, DROP, and RESOLVE operation modes described in "Operation modes" on page 451. It can interrogate GDDR for the configuration or run without GDDR, as discussed in "Configuration identification" on page 450. The utility produces reports described in "GDDMSCFX reports" on page 454 and shown in "Sample output" on page 459.

You can run the GDDMSCFX utility in batch or in foreground using the MSC command in the Perform Health Check panel (C). To run the utility from the Perform Health Check panel (C), APF-authorize GDDMSCFX by adding it to the IKJTSO parameter file under AUTHPGM.

## Requirements

SCF (Symmetrix Control Facility) must be active.

If GDDR is to be interrogated for the configuration, GDDRMAIN must be active.

## Configuration identification

GDDMSCFX can run with or without GDDR. By default, if a GDDR subsystem is found matching the GDD$*nnnn* DD DUMMY statement in the GDDMSCFX JCL (or the default subsystem name of 'GDDR' if none is specified), then the utility interrogates GDDR for the configuration. If a matching GDDR subsystem is found, GDDRMAIN must be active for that subsystem. Optionally, the NOGDDR parameter may be specified to run without GDDR, ignoring any matching GDDR subsystem whether active or inactive.

When running with GDDR, the MSCGROUP and CONTROLLER parameters are optional and act as a filter, limiting the scope of the utility to the specified MSC group and/or storage system.

When running without GDDR, the MSCGROUP and CONTROLLER parameters are required. The CONTROLLER parameter provides a starting point for the utility to discover the configuration but does not limit the scope of the utility. The CONTROLLER parameter must specify any accessible storage system in the configuration to be used as a starting point (similar to a gatekeeper). This storage system may be local or remote to the system where GDDMSCFX is running. GDDMSCFX looks for SRDF groups matching the specified MSC group name on the specified storage system. Then the multibox lists are traversed until all SRDF groups in the MSC group are discovered.

GDDMSCFX can run at any site and adaptively finds paths to each storage system in the configuration. For better reliability and resiliency in a disaster situation, the utility tries all possible local and remote paths before concluding that a storage system is inaccessible. Before a path is selected, it undergoes a series of validation checks to ensure that the UCB is accessible and is in a valid state, the UCB was not swapped, I/O can be done to the UCB or via the UCB to the remote storage system (if applicable),

and the destination serial number matches the storage system it is attempting to access. If no valid paths are found, that storage system is deemed inaccessible, and the utility moves on to the next storage system in the configuration.

**Note:** Paths to each storage system are obtained from SCF.

## Operation modes

The GDDMSCFX utility provides the following operation modes:

- VALIDATE
- CLEAR
- DROP
- RESOLVE

By default the utility runs in the VALIDATE mode. All other modes imply VALIDATE.

The modes can be specified in combination. If a combination of CLEAR, DROP, and RESOLVE is specified, the report is repeated, and a header line is inserted before each phase. The order the parameters are specified is irrelevant. DROP is always performed before RESOLVE, and CLEAR is always performed last. If all the three options are specified, the order is: DROP, RESOLVE, and then CLEAR.

**Note:** CLEAR and DROP modes are available starting with GDDR 5.2 PTF GD52009. RESOLVE mode is available starting with GDDR 5.2 PTF GD52019.

In CLEAR, DROP, and RESOLVE modes, a WTOR is issued by default for each SRDF group being processed. To disable these prompts, specify the NOPROMPT option.

You can simulate CLEAR, DROP, and RESOLVE processing by setting the SIMULATE option.

### VALIDATE

The GDDMSCFX utility validates storage system information for each SRDF group against the configuration defined to GDDR (if applicable). It also validates the storage system information across SRDF groups for discrepancies, and ensures only the expected storage system information exists. Any issues or discrepancies are indicated in the report with an asterisk, in which case GDDMSCFX ends with return code 4. This helps you detect any stale storage system information that may have been left behind by a previous configuration or recovery procedure.

To run GDDMSCFX in the VALIDATE mode, specify the VALIDATE option.

For each SRDF group, the following validation checks are performed, where applicable:

- The SRDF group exists and is online
- Devices are ready (not TNR)
- The MSC scratch area exists
- The multibox list exists and contains the SRDF group

- SRDF/A is active and cycle-switching
- MSC is active
- Star/Star-A/SQAR recovery is available
- Remote is consistent
- Global consistency achieved
- Site C is not ahead of site B
- The SRDF group is not in the Transmit Idle state
- Recovery group is not used for multiple SRDF/A sessions

If GDDR is available, the following additional validation checks are performed against the GDDR configuration for each SRDF group:

- The SRDF group is known to GDDR
- The SRDF group is in the active GDDR configuration
- The following items are as GDDR expects:
  - The MSC group name in the scratch area
  - The MSC site designation (A, B, or C) in the scratch area
  - The other-side SRDF group
  - The SRDF group type (SRDF/A, SRDF/S, or Recovery)
  - The SRDF personality (R1 or R2)
  - The configuration mode (MSC, Star, Star-A, or SQAR)
  - The cycle tag mode (MSC, Star, Star-A, or SQAR)
  - The Star/SQAR mode (Star, Star-Rcvy, Star-A, Star-A-Rcvy, SQAR, SQAR-Rcvy, or MSC)

If GDDR is not available, GDDMSCFX does not perform validation against the GDDR configuration. GDDMSCFX checks that the expected storage system information exists and validates this information across SRDF groups for consistency. If an inconsistency is detected, a warning message is displayed before the configuration summary, and GDDMSCFX ends with return code 4. This may be indicative of stale box information.

If GDDR is not available, the following additional validation checks are performed for each SRDF group:

- The MSC site designation is consistent across all MSC scratch areas on a given storage system
- The configuration mode is consistent across all MSC scratch areas
- The cycle tag mode is consistent across all SRDF/A groups

Storage system information is cleared for a given SRDF group if one or more of the following conditions are true for that group, where applicable:

- The SRDF group no longer exists

- The SRDF group is not found in multibox list
- The other-side SRDF group in scratch area does not match the multibox list
- The SRDF group is not known to GDDR
- The SRDF group is not in the active GDDR configuration
- The following items are not what GDDR expects:
  - The configuration mode in the scratch area
  - The site designation in the scratch area
  - The MSC group name in the scratch area
  - The SRDF group type (SRDF/A, SRDF/S, recovery) in the scratch area
  - The other-side SRDF group in the scratch area
  - The other-side SRDF group in the multibox list
  - SRDF personality in the multibox list
  - The Star/SQAR mode

## CLEAR

The GDDMSCFX utility clears storage system information for all SRDF groups indicated as eligible for CLEAR in the VALIDATE report.

To run GDDMSCFX in the CLEAR mode, specify the CLEAR option.

For a given SRDF group, the following information can be cleared:

- MSC scratch area
- Multibox list
- Star/SQAR mode indicator
- MSC mode indicator

Storage system information is not cleared if one or more SRDF groups are found with host intervention required or SRDF/A active.

The FORCE option may be used to clear storage system information for all SRDF groups in the configuration regardless of whether stale data was found.

## DROP

The GDDMSCFX utility drops SRDF/A for all eligible SRDF groups in the configuration.

To run GDDMSCFX in the DROP mode, specify the DROP option.

SRDF/A is dropped for a given SRDF group if one or more of the following conditions are true for that group, where applicable:

- SRDF/A is not inactive
- SRDF group is in transit idle
- SRDF/A is active, but one or more R1 devices are TNR
- SRDF/A is active but not cycling

- SRDF/A is active, but a link issue was detected

If GDDMSCFX concludes that even a single SRDF group under MSC control must be dropped, then all other groups under MSC control with SRDF/A active are also dropped.

To unconditionally drop all SRDF groups in the configuration with SRDF/A active, specify the FORCE option.

RESOLVE

The GDDMSCFX utility interrogates the cycle tags and the completeness of the cycles for each SRDF/A group with the Host Intervention Required flag and commits or discards the Receive cycles, as appropriate. As the cycles are resolved, the Host Intervention Required flag is reset.

### Restrictions

- Cycle resolution is not performed if an error occurs when querying one or more SRDF groups in the MSC group.
- Cycle resolution is not performed if one or more SRDF groups in the MSC group have SRDF/A active, or have R1 cleanup or R2 restore in progress.
- The utility stops processing if an error occurs committing or discarding SRDF/A cycles.

## GDDMSCFX reports

The utility generates a report containing the following information:

- Report header providing supplementary information, such as GDDR, SymmAPI, and SCF version and maintenance levels, GDDR and SCF subsystem names, GDDMSCFX parameters, the report date and time, and so on.
- Configuration summary:
  - Configuration type and mode
  - All GDDR-managed SRDF groups for each site and their expected roles
  - All device ranges and the total device count for each SRDF group

    **Note:** Shown only if the utility interrogates GDDR for configuration information.

    Ranges containing devices in TNR status are flagged with an asterisk, and the device numbers are listed to the right.
- SRDF group details by site and then by storage system

  The report shows overall and detailed status information for each SRDF group in the configuration in addition to any non-GDDR-managed SRDF groups for which storage system information is found matching the active MSC group name for that site pair.
  - For all groups:
    - The group is online or not

  - Star/SQAR mode
  - The other-side storage system and SRDF group
- For SRDF/A and SRDF/S groups only:
  - MSC group name
  - Multibox list
- For SRDF/A groups only:
  - SRDF/A and MSC is active or not
  - The remote side is consistent or not
  - Global consistency achieved or not
  - Recovery is available or not
  - Transmit Idle is set or not
  - Last cycle switch date and time
- For SRDF/A groups under MSC control only:
  - Target time between MSC cycle switches (MSC cycle target)
- For SRDF/A groups not under MSC control only:
  - Minimum cycle time
- For SRDF/A R1 groups only:
  - Tolerance mode is set or not
  - R1 cleanup is running or not
  - Transmit cycle is empty or not
  - Capture and Transmit cycle numbers and tags
  - Multi-cycle mode is set or not
  - Secondary delay
- For SRDF/A R1 groups in multi-cycle mode only:
  - Last and average transfer times in seconds
  - Queue depth in cycles
- For SRDF/A R2 groups only:
  - Intervention is required or not
  - R2 cleanup is running or not
  - R2 restore is complete or not
  - Apply cycle is empty or not
  - Receive and Apply cycle numbers and tags
  - Cycle TOD (time of day)

Any issues (for example, stale storage system information) are flagged with an asterisk.

- SRDF/A cycle information

  For each SRDF/A group, the report includes SRDF/A cycle tags, whether the Host Intervention Required flag is set, and (when run in the RESOLVE mode) the action taken by the utility to resolve SRDF/A cycles.

  In the DROP mode, the dropped SRDF groups are also shown.

  If GDDMSCFX is run when SRDF/A is active and cycle switching, the MSC cycle tags are shown as follows:

  - For the R1 side and the R2 side, the inactive cycle tag is typically one lower than the active cycle tag. But when SRDF/A is running in multi-cycle mode, this difference may be bigger on the R1 side.
  - There is no guaranteed relationship between R1-side and R2-side cycle tags because they are queried separately and seconds might pass between the R1-side versus R2-side query.

- A summary message indicating the utility return code

The report is written to the SYSPRINT DD. If the DEBUG option is specified, diagnostic information is also written to SYSPRINT in line with the report. All GDDR global variables used by the utility are echoed to the GVAROUT DD.

## Sample JCL

Adjust the GDDMSCFX sample JCL as follows:

- Specify your GDDR LINKLIB in STEPLIB DD.
- The *nnnn* in SCF$*nnnn* DD DUMMY must match your SCF subsystem name.
- The *nnnn* in GDD$*nnnn* DD DUMMY must match your GDDR subsystem name.

  **Note:** These DDs link the utility to the correct SCF and GDDRMAIN address spaces. The default SCF subsystem name is 'EMC', in which case SCF$*nnnn* DD DUMMY is optional. The default GDDR subsystem name is 'GDDR', in which case GDD$*nnnn* DD DUMMY is optional.

- Specify the EXEC parameters.

```
//GDDMSCFX JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
...
//*
//GDDMSCFX EXEC PGM=GDDMSCFX,PARM='parameters'
//STEPLIB  DD  DISP=SHR,DSN=ds-prefix.LINKLIB  <--- Your GDDR LINKLIB
//SYSPRINT DD  SYSOUT=*
//GVAROUT  DD  SYSOUT=*
//SYSABEND DD  SYSOUT=*
//SCF$nnnn DD  DUMMY  <--- Your SCF subsystem name ('EMC' is default)
//GDD$nnnn DD  DUMMY  <--- Your GDDR subsystem name ('GDDR' is default)
//*
```

## EXEC parameters

The syntax is as follows:

```
[VALidate] [,CLeaR] [,DRoP] [,RESolve]
[,MSCGroup=msc-group-name] [,ConTRoLler=serial#]
[,SITE=site] [,SITEPair=sitepair]
[,PRompt|NOPRompt]
[,SIMulate]
[,FoRCe]
[,NOGDdr]
[,DeBuG]
```

**Note:** If GDDR is not available or the NOGDdr option is specified, the MSCGroup and ConTRoLler parameters are required.

**Note:** If GDDR is available and the RESOLVE option is specified, the ConTRoLler parameter cannot be used to limit operation to a single storage system.

**Note:** The PRompt (NOPRompt) and SIMulate options are only applicable when CLEAR, DROP, or RESOLVE is specified. The FoRCe option is only applicable when CLEAR or DROP is specified.

**Note:** Parameters can be specified as either *keyword* or *keyword*={Y|N}. For example, 'DEBUG' implies 'DEBUG=Y'. Omission of 'DEBUG' implies the default value of 'DEBUG=N'.

Where:

`CLeaR`

Runs the GDDMSCFX utility in the CLEAR mode described in "CLEAR" on page 453.

**⚠WARNING**

**Run first with the SIMULATE option specified and ensure the results are as expected.**

`ConTRoLler=serial#`

Specifies the storage system serial number.

If GDDR is available, the CONTROLLER parameter is used as a filter to limit operation to the specified storage system (drop side for the DROP mode).

**Note:** The CONTROLLER filter cannot be specified together with the RESOLVE option. SRDF/A cycle resolution cannot be limited to a single storage system because the GDDMSCFX utility must discover all SRDF groups in the MSC group.

If GDDR is not available or the NOGDDR option is set, both MSCGROUP and CONTROLLER must be specified to identify the configuration. The CONTROLLER parameter specifies the serial number of any accessible storage system in the configuration to be used as a starting point (similar to a gatekeeper). The storage system may be local or remote to the system where GDDMSCFX is running.

This parameter is required if GDDR is not available or the NOGDDR option is specified.

`DeBuG`

Enables debugging to the SYSPRINT DD.

`DROP`

Runs the GDDMSCFX utility in the DROP mode described in "DROP" on page 453.

**⚠WARNING**

**Run first with the SIMULATE option specified and ensure the results are as expected.**

`FoRCe`

If CLEAR is specified, clears storage system information for all SRDF groups in the configuration regardless of whether stale data was found.

If DROP is specified, unconditionally drops all SRDF groups in the configuration with SRDF/A active.

**⚠WARNING**

**Run first with the SIMULATE parameter specified to ensure the results are as expected.**

`MSCGroup=`*`msc-group-name`*

The MSC group name identifying the configuration.

If GDDR is available, the MSCGROUP parameter is used as a filter to limit the requested functions to the specified MSC group.

If GDDR is not available or the NOGDDR option is set, both MSCGROUP and CONTROLLER must be specified in order to identify the configuration. GDDMSCFX looks for SRDF groups matching this MSC group name on the storage system indicated by the CONTROLLER parameter. Then the multibox lists are traversed from system to system until all SRDF groups in the MSC group are discovered.

This parameter is required if GDDR is not available or the NOGDDR option is set.

`NOGDdr`

When specified, the GDDMSCFX utility does not interrogate GDDR for the configuration, as described in "Configuration identification" on page 450.

**Note:** When NOGDDR is specified, both MSCGROUP and CONTROLLER parameters are required.

PRompt | NOPRompt

If CLEAR, DROP, or RESOLVE is specified, this option determines whether to issue a prompt before processing each SRDF group. NOPrompt suppresses all prompts.

**⚠WARNING**

**When set to NOPRompt, run the utility with the SIMULATE parameter specified to ensure the results are as expected.**

RESolve

Runs the GDDMSCFX utility in the RESOLVE mode described in "RESOLVE" on page 454.

**⚠WARNING**

**Run first with the SIMULATE option specified and ensure the results are as expected.**

SIMulate

If CLEAR, DROP, or RESOLVE is specified, this option simulates processing without making any changes.

SITE=*site*

Limits operation to the specified site (the drop side for the DROP mode, the SRDF/A R2 site for the RESOLVE mode).

If GDDR is available, the SITE parameter uses the GDDR site designation and valid values for *site* are DC1, DC2, DC3, DC4 (recommended) or A, B, C, D.

If GDDR is not available, the SITE parameter uses the MSC site designation and valid values for *site* are A, B, C, D.

SITEPair=*sitepair*

Limits operation to the specified site pair.

If GDDR is available, the SITEPAIR parameter uses the GDDR site designation and valid values for *sitepair* are DC*n*-DC*m* (recommended) or A-B, A-C, B-C, B-D, C-D.

If GDDR is not available, the SITEPAIR parameter uses the MSC site designation and valid values for *sitepair* are A-B, A-C, B-C, B-D, C-D.

VALidate

Runs the GDDMSCFX utility in the VALIDATE mode described in "VALIDATE" on page 451. This is the default mode.

## Sample output

*Example 1: VALIDATE mode, GDDR available*

In the following example, GDDR was available and was interrogated for the configuration. The utility was run in the VALIDATE mode and ended with return code 0.

**Note:** The example illustrates an SRDF/Star-A configuration.

```
*****************************************
```

```
* GDDMSCFX-11/29/18-15.33-GDDR520-GD52024 *
*******************************************

Parameters            : VALidate
SCF version           : 8.3.0 SF83033
API version           : 8.3.0 BASE (02/06/18)
SCF subsystem name    : DVT
GDDR version          : 5.2.0 GD52024 (11/29/18)
GDDR subsystem name   : GDDR
GDDMSCFX PTFs applied : 024 009 019
Running at site       : DC1 (primary DASD site)
Running on system     : LB04M41 (master C-system)
Date and time         : 11/29/18 15:34:01

Configuration Summary
=====================

 Config type: Con Star-A
 Config mode: Star-A

  Site A (DC1)  SRDF/A  Site C (DC3)
  ============ ======== ============
  000195700794 80 -> 80 000195700987
     04D0-04EF    ->    04D0-04EF
     051A-0538    ->    051A-0538
     3350-3384    ->    3350-3384
     3C0D-3C56    ->    3C0D-3C56
    -----------         -----------
    126 devices        126 devices

  Site A (DC1)  SRDF/A  Site D (DC4)
  ============ ======== ============
  000195700794 81 -> 81 000195700866
     04D0-04EF    ->    04D0-04EF
     051A-0538    ->    051A-0538
     3350-3384    ->    3350-3384
     3C0D-3C56    ->    3C0D-3C56
    -----------         -----------
    126 devices        126 devices

  Site C (DC3) Recovery Site D (DC4)
  ============ ======== ============
  000195700987 82 -> 82 000195700866
     04D0-04EF    ->    04D0-04EF
     051A-0538    ->    051A-0538
     3350-3384    ->    3350-3384
     3C0D-3C56    ->    3C0D-3C56
    -----------         -----------
    126 devices        126 devices

Site A (DC1) (MSC Site A)
=========================

 Controller 0001957-00794 (5876.309)
 -----------------------------------

  RDF group 80 (SRDF/A R1)

   Online                    : Y
   SRDF/A active             : Y
   MSC active                : Y
   Remote consistent         : Y
   Global consistency        : Y
   Star-A recovery available : Y
   Transmit idle             : N
   Tolerance mode            : N
```

```
   R1 Cleanup running        : N
   Transmit cycle empty      : N
   Multi-cycle mode          : N
   Capture cycle tag         : C1000000000EB744
   Transmit cycle tag        : C1000000000EB743
   Secondary delay           : 11/29/18 15:33:55 (7 seconds)
   Time of last cycle switch : 11/29/18 15:34:00 (2 seconds)
   MSC cycle target          : 5 seconds
   Mode                      : Star-A
   MSC group name            : DVTSTR13
   Other side RDF group      : 000195700987/80

   Multi-box list:

     Site A (DC1)         Site C (DC3)
    ---------------      ---------------
    000195700794/80 ---> 000195700987/80

  RDF group 81 (SRDF/A R1)

   Online                    : Y
   SRDF/A active             : Y
   MSC active                : Y
   Remote consistent         : Y
   Global consistency        : Y
   Star-A recovery available : Y
   Transmit idle             : N
   Tolerance mode            : N
   R1 Cleanup running        : N
   Transmit cycle empty      : Y
   Multi-cycle mode          : N
   Capture cycle tag         : C1000000000EB732
   Transmit cycle tag        : C1000000000EB731
   Secondary delay           : 11/29/18 15:33:55 (7 seconds)
   Time of last cycle switch : 11/29/18 15:34:00 (2 seconds)
   MSC cycle target          : 5 seconds
   Mode                      : Star-A
   MSC group name            : DVTSTR14
   Other side RDF group      : 000195700866/81

   Multi-box list:

     Site A (DC1)         Site D (DC4)
    ---------------      ---------------
    000195700794/81 ---> 000195700866/81

Site C (DC3) (MSC Site C)
=========================

 Controller 0001957-00987 (5876.309)
 -----------------------------------

  RDF group 80 (SRDF/A R2)

   Online                    : Y
   SRDF/A active             : Y
   MSC active                : Y
   Remote consistent         : Y
   Global consistency        : Y
   Star-A recovery available : Y
   Transmit idle             : N
   Intervention required     : N
   R2 Cleanup running        : N
   R2 Restore complete       : Y
   Apply cycle empty         : Y
   Receive cycle tag         : C1000000000EB743
```

```
   Apply cycle tag           : C1000000000EB742
   Cycle time of day         : 11/29/18 15:33:55 (8 seconds)
   Time of last cycle switch : 11/29/18 15:34:00 (3 seconds)
   MSC cycle target          : 5 seconds
   Mode                      : Star-A
   MSC group name            : DVTSTR13
   Other side RDF group      : 000195700794/80

   Multi-box list:

     Site A (DC1)         Site C (DC3)
    ---------------      ---------------
    000195700794/80 ---> 000195700987/80

  RDF group 82 (Recovery R1)

   Online                    : Y
   Mode                      : Star-A-Rcvy
   Other side RDF group      : 000195700866/82

   No scratch area or MB list for Recovery groups (referenced in RDF group 80 scratch
    area)

Site D (DC4) (MSC Site C)
=========================

 Controller 0001957-00866 (5876.309)
 -----------------------------------

  RDF group 81 (SRDF/A R2)

   Online                    : Y
   SRDF/A active             : Y
   MSC active                : Y
   Remote consistent         : Y
   Global consistency        : Y
   Star-A recovery available : Y
   Transmit idle             : N
   Intervention required     : N
   R2 Cleanup running        : N
   R2 Restore complete       : Y
   Apply cycle empty         : Y
   Receive cycle tag         : C1000000000EB732
   Apply cycle tag           : C1000000000EB731
   Cycle time of day         : 11/29/18 15:34:00 (5 seconds)
   Time of last cycle switch : 11/29/18 15:34:05 (0 seconds)
   MSC cycle target          : 5 seconds
   Mode                      : Star-A
   MSC group name            : DVTSTR14
   Other side RDF group      : 000195700794/81

   Multi-box list:

     Site A (DC1)         Site D (DC4)
    ---------------      ---------------
    000195700794/81 ---> 000195700866/81

  RDF group 82 (Recovery R2)

   Online                    : Y
   Mode                      : Star-A-Rcvy
   Other side RDF group      : 000195700987/82

   No scratch area or MB list for Recovery groups (referenced in RDF group 81 scratch
    area)
```

```
SRDF/A Cycle Summary
====================

      R1              R2                    _______________R1_______________ _______________R2_______________
 Controller/Grp#  Controller/Grp# MSC Group  Capture Cycle    Transmit Cycle   Receive Cycle     Apply Cycle    HIR
 --------------  --------------- ---------- ---------------- ---------------- ---------------- ---------------- ---
 000195700794/80->000195700987/80 DVTSTR13  C1000000000EB744 C1000000000EB743 C1000000000EB743 C1000000000EB742  N
 000195700794/81->000195700866/81 DVTSTR14  C1000000000EB732 C1000000000EB731 C1000000000EB732 C1000000000EB731  N

  Note: SRDF/A is actively cycle-switching. Consistency of cycle tags across R1 and R2 cannot be guaranteed.

GDDMSCFX ended with RC 0
```

*Example 2: VALIDATE mode, SRDF group in unexpected state*

The utility was run in the VALIDATE mode and ended with return code 4, as one or more SRDF groups were in an unexpected state.

**Note:** The example illustrates an SRDF/A configuration.

The Configuration Summary indicates that some SRDF/A R1 devices are in TNR status:

```
Site A (DC1)  SRDF/A  Site B (DC3)
  ============ ======== ============
  000197200654 5C -> 5C 000195700866
   * 009D-00AC    ->    1881-1890     * TNR devices: 009D-00AC
    -----------        -----------
    16 devices         16 devices
  000197200654 5A -> 5A 000197700135
   * 0BA0-0BAF    ->    00B0-00BF     * TNR devices: 0BA0-0BAF
    -----------        -----------
    16 devices         16 devices
```

The body of the report also indicates some issues. SRDF/A is not active; therefore, there is a delay in cycle switching. There is also a warning message summarizing the number of TNR devices in each SRDF group.

```
RDF group 5C (SRDF/A R1)

   Online                     : Y
   SRDF/A active              : N *
   MSC active                 : Y
   Remote consistent          : Y
   Global consistency         : Y
   MSC recovery available     : Y
   Transmit idle              : N
   Tolerance mode             : N
   R1 Cleanup running         : N
   Transmit cycle empty       : Y
   Multi-cycle mode           : Y
   Last transfer time         : 0 seconds
   Average transfer time      : 0 seconds
   Queue depth                : 0 cycles
   Capture cycle tag          : C0000000000070BD
   Transmit cycle tag         : C0000000000070BE
   Secondary delay            : 11/29/18 16:14:30 (76 seconds) *
   Time of last cycle switch  : 11/29/18 16:14:35 (71 seconds) *
   MSC cycle target           : 5 seconds
   Mode                       : MSC
   MSC group name             : LB01M313
   Other side RDF group       : 000195700866/5C

   * 16 (out of 16) devices found in TNR status *

   Multi-box list:

     Site A (DC1)          Site B (DC3)
    ---------------       ---------------
    000197200654/5A ---> 000197700135/5A
    000197200654/5C ---> 000195700866/5C
```

*Example 3: VALIDATE mode, GDDR not available*

The utility was run in the VALIDATE mode with PARM='MSCG=DVTSTR14, CTRL=000195700794'. It ended with return code 0.

**Note:** The example illustrates an SRDF/Star-A configuration.

```
*****************************************
* GDDMSCFX-11/29/18-15.33-GDDR520-GD52024 *
*****************************************

Parameters            : VALidate MSCGroup(DVTSTR14) ConTRoLler(0001957-00794)
SCF version           : 8.3.0 SF83033
API version           : 8.3.0 BASE (02/06/18)
SCF subsystem name    : DVT
GDDR version          : 5.2.0
GDDR subsystem name   : n/a
GDDMSCFX PTFs applied : 024 009 019
Running on system     : LB04M41
Date and time         : 11/29/18 18:43:08

Configuration Summary
=====================

 Config mode : Star-A
 Topology    : Concurrent

 RDF groups in MSC group DVTSTR14:

     Site A     SRDF/A     Site C
  ============ ======== ============
  000195700794 81 -> 81 000195700866

Site A
======

 Controller 0001957-00794 (5876.309)
 -----------------------------------

  RDF group 81 (SRDF/A R1)

   Online                    : Y
   SRDF/A active             : Y
   MSC active                : Y
   Remote consistent         : Y
   Global consistency        : Y
   Star-A recovery available : Y
   Transmit idle             : N
   Tolerance mode            : N
   R1 Cleanup running        : N
   Transmit cycle empty      : N
   Multi-cycle mode          : N
   Capture cycle tag         : C1000000000EC011
   Transmit cycle tag        : C1000000000EC010
   Secondary delay           : 11/29/18 18:43:05 (6 seconds)
   Time of last cycle switch : 11/29/18 18:43:10 (1 seconds)
   MSC cycle target          : 5 seconds
   Mode                      : Star-A
   MSC group name            : DVTSTR14
   Other side RDF group      : 000195700866/81

   Multi-box list:

        Site A              Site C
    ---------------      ---------------
    000195700794/81 ---> 000195700866/81
```

```
Site C
======

 Controller 0001957-00866 (5876.309)
 -----------------------------------

  RDF group 81 (SRDF/A R2)

   Online                    : Y
   SRDF/A active             : Y
   MSC active                : Y
   Remote consistent         : Y
   Global consistency        : Y
   Star-A recovery available : Y
   Transmit idle             : N
   Intervention required     : N
   R2 Cleanup running        : N
   R2 Restore complete       : Y
   Apply cycle empty         : Y
   Receive cycle tag         : C1000000000EC010
   Apply cycle tag           : C1000000000EC00F
   Cycle time of day         : 11/29/18 18:43:06 (8 seconds)
   Time of last cycle switch : 11/29/18 18:43:11 (3 seconds)
   MSC cycle target          : 5 seconds
   Mode                      : Star-A
   MSC group name            : DVTSTR14
   Other side RDF group      : 000195700794/81

   Multi-box list:

        Site A              Site C
    ---------------      ---------------
    000195700794/81 ---> 000195700866/81

  RDF group 82 (Recovery)

   Online                    : Y
   Mode                      : Star-A-Rcvy
   Other side RDF group      : 000195700987/82

   No scratch area or MB list for Recovery groups (referenced in RDF group 81 scratch
    area)

SRDF/A Cycle Summary
====================

       R1              R2                  _______________R1_______________ _______________R2_______________
 Controller/Grp#  Controller/Grp# MSC Group  Capture Cycle    Transmit Cycle   Receive Cycle     Apply Cycle    HIR
 ---------------  ---------------- --------- ---------------- ---------------- ---------------- ---------------- ---
 000195700794/81->000195700866/81 DVTSTR14  C1000000000EC011 C1000000000EC010 C1000000000EC010 C1000000000EC00F  N

GDDMSCFX ended with RC 0
```

*Example 4: CLEAR mode*

When GDDMSCFX is run in the CLEAR mode, messages are issued in line with the report indicating which storage system information is cleared for each SRDF group.

For an SRDF/A R1 group:

```
Scratch area cleared for RDF group 81

Multi-box list cleared for RDF group 81

Star mode cleared for RDF group 81
```

```
   MSC mode cleared for RDF group 81
```

For an SRDF/A R2 group:

```
   Scratch area cleared for RDF group 80

   Multi-box list cleared for RDF group 80

   Star mode cleared for RDF group 80
```

For a recover group:

```
   Star-Rcvy mode cleared for RDF group 82
```

*Example 5: DROP mode*

When GDDMSCFX is run in the DROP mode, messages are issued in line with the report indicating which SRDF groups are dropped:

```
GDDUM20I SRDF/A dropped for DC1 controller 0001957-00794 RDF group 80
GDDUM20I SRDF/A dropped for DC1 controller 0001957-00794 RDF group 81
GDDUM20I SRDF/A dropped for DC3 controller 0001957-00987 RDF group 80
GDDUM20I SRDF/A dropped for DC4 controller 0001957-00866 RDF group 81
```

*Example 6: RESOLVE mode*

The example shows that the Receive cycles were complete for both SRDF groups in the MSC group, so the cycles were committed.

**Note:** This example illustrates an SRDF/A configuration.

```
...
SRDF/A Cycle Resolution Results
==============================

      R1              R2                    _____________R1_____________ _____________R2_____________
 Controller/Grp#  Controller/Grp# MSC Group  Capture Cycle    Transmit Cycle   Receive Cycle    Apply Cycle    HIR   Action
 --------------  --------------- --------- ---------------- ---------------- ---------------- ---------------- --- ---------
 000197200654/5C->000195700866/5C LB01M313  C0000000000070BE C0000000000070BD C0000000000070BD C0000000000070BC  Y  Committed
 000197200654/5A->000197700135/5A LB01M313  C0000000000070BE C0000000000070BD C0000000000070BD C0000000000070BC  Y  Committed
```

*Example 7: SIMULATE option specified*

If the SIMULATE option is specified, "* Simulated *" is displayed after each message, for example:

```
Scratch area cleared for RDF group 81    * Simulated *
```

*Example 8: PROMPT option specified*

If GDDMSCFX is run with the PROMPT option specified (or defaulted), a WTOR is issued before each SRDF group is processed. This allows you to target specific SRDF groups. Also, a WTO is issued confirming each operator response. These messages are in addition to those issued in line with the GDDMSCFX report.

**Note:** The following examples illustrate the CLEAR mode.

Enter Yes to clear the storage system information for the indicated SRDF group, for example:

```
*75 GDDUM01A Clear box info for DC1 controller 0001957-00794 RDF group
   81? (Reply Yes, No, Cancel, or NOPRompt)
R 75,Y
 GDDUM02I Operator replied Yes - Clearing box info for DC1 controller
   0001957-00794 RDF group 81
```

Enter No to bypass the indicated SRDF group, for example:

```
*76 GDDUM01A Clear box info for DC1 controller 0001957-00794 RDF group
   80? (Reply Yes, No, Cancel, or NOPRompt)
R 76,N
 GDDUM04I Operator replied No - Box info not cleared for DC1 controller
   0001957-00794 RDF group 80
```

Enter NOPRompt to clear the storage system information for the indicated SRDF group and all remaining groups without further prompting, for example:

```
*77 GDDUM01A Clear box info for DC3 controller 0001957-00987 RDF group
   80? (Reply Yes, No, Cancel, or NOPRompt)
R 77,NOPR
 GDDUM03I Operator replied NOPRompt - Proceeding without prompts
```

Enter Cancel to bypass the indicated SRDF group and all remaining groups, for example:

```
*78 GDDUM01A Clear box info for DC1 controller 0001957-00794 RDF group
   81? (Reply Yes, No, Cancel, or NOPRompt)
R 78,C
 GDDUM05I Operator replied Cancel - Processing terminated
```

## Return codes

The return codes are as follows:

0 Processing completed successfully, including the following:

- All expected storage system information was found
- All SRDF groups are in the expected state
- If CLEAR, DROP, or RESOLVE was specified: all eligible SRDF groups were processed successfully

4 Processing completed with warning(s), for example:

- Stale or unexpected storage system information was found
- Expected storage system information was not found
- One or more SRDF groups or devices are in an invalid state

8 Processing completed with error(s), for example:

- Discovery for the specified MSC group failed because the required storage system information was not found
- MSC group names for the configured site pair(s) are not defined to GDDRMAIN
- One or more sites or storage systems referenced in SRDF devices parameters are not defined to GDDRMAIN
- A SymmAPI error occurred
- The operating environment level of one or more storage systems is too low
- CLEAR, DROP, or RESOLVE is prohibited, as one or more SRDF groups are in an unexpected state

12 Processing failed unexpectedly, for example:

- SCF is not active
- SymmAPI version is too low
- GDDRMAIN is not active
- The GDDR subsystem name could not be determined
- The GDDR data space was not found
- A 'GDDR' license was not found
- A parse error occurred
- A post-parse error occurred
- Global variable read error occurred
- RESOLVE failed because the required storage system information was not found
- RESOLVE failed because of a possible data consistency issue

16 Abend occurred, SVC dump is taken

20 Operator replied Cancel to WTOR

# GDDR SDDF Session Verification utility (GDDRSDDF)

The GDDR SDDF Session Verification utility (GDDRSDDF) verifies status of the SDDF sessions in your storage systems. The utility only queries the SDDF session information and does not affect the state of the GDDR-managed configuration. It creates a GDDR run-type global variable which is not currently used by any other GDDR process.

## Requirements

- SCF must be up and running with the SCF$*nnnn* connector specified in your customized GDDRPROC JCL, which is pointed to by the JCLLIB statement.
- GDDRMAIN must be up and running, and connected with the same SCF instance.
- RDF.DEVICES parameters reflecting site pairs as expected for the defined type of configuration.
- Channel and SRDF group connectivity to the storage systems relevant to the queries.

The utility can run on any site in the GDDR configuration, but is recommended to run on one of the sites expected to have SDDF sessions: DC3 (Site C) or DC4 (Site D).

The utility only needs to run on one of the sites. It retrieves information from both relevant sites at run-time.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRSDDF).

```
//GDDRSDDF JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRSDDF EXEC GDDRPROC
//SDDFMSGS DD  SYSOUT=*
//SDDFLIST DD  SYSOUT=*
//DIFFLIST DD  SYSOUT=*
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRSDDF arguments
/*
```

## Optional DD cards

The following optional DD cards can be specified:

- SDDFMSGS for device-level error messages
- SDDFLIST for SDDF session details for GDDR-managed devices
- DIFFLIST to provides an estimate of the size of the differential between the 2 sites on the next CREATEPAIR or RESUMEPAIR DIFFERENTIAL operation.

If SDDFLIST and/or DIFFLIST is not allocated, the affected report is not written, but the utility is expected to run normally, and will issue appropriate messages in the joblog or in the SDDFMSGS dataset to warn you of any abnormalities in the SDDF sessions.

If SDDFMSGS is not allocated, only summary messages are written that do not provide device-level details.

## Arguments

The syntax is as follows:

```
GDDRRXST GDDRSDDF
[N(run-count)]
[I(minutes)]
[D({Y|N})]
[P(current-primary-site)]
[C({STAR|SQAR})]
```

**Note:** The `T({COncurrent|CAscaded})` argument is ignored in SQAR configurations.

Where:

`C({STAR|SQAR})`

Specifies the configuration type: STAR or SQAR.

Default: The configuration type known to GDDR.

This argument is provided for completeness but is of very limited use. This argument is intended for lab environments. If you use this argument to specify a different type of configuration than the one known to GDDR, it must have an equal or lesser number of sites than the GDDR configuration. Also RDF.DEVICES parameters must exist for site pairs expected for the specified type of configuration.

`D({Y|N})`

Specifies whether or not you want the utility to estimate the size of the differential at runtime. Valid values are Y or N.

Default: N

It is recommended to use the default value.

The estimate provided is an approximation at best, reflecting only the situation at run-time of the utility. The size of the differential seen during a GDDR script which has a CREATEPAIR or RESUMEPAIR DIFFERENTIAL command may vary.

D(Y) should never be used when SRDF/A and MSC are expected to be cycling normally. Results are unpredictable, and the provided estimate will be grossly incorrect. You will almost certainly get unnecessary Full PUSH warnings.

Specifying D(Y) will cause several I/Os per device in the GDDR-managed configuration and will slow utility processing considerably.

`I(minutes)`

Specifies the number of minutes between SDDF session checks (runs). Any numeric value, even non-integer, is allowed (for example, 1.5). The utility performs a run, then waits for a number of seconds corresponding to the specified number of minutes.

Default: 15

The right interval size will be a function of the configuration. If the utility issues messages indicating that SDDF session time stamps have not changed, it may be necessary to increase the interval size.

A convenient way of determining an optimal interval size, is to start with a small value, for example, 2 minutes, and a value of 10 or more for the N argument. If the MSC software is currently performing normal SDDF activity, then the GDDR SDDF Session Verification utility will issue messages on each run about a decreasing number of devices for which SDDF session time stamps have not changed. The right interval is then equal to the short interval, multiplied by the lowest run count for which no more devices are said to have unchanged SDDF session time stamps.

`N(`*`run-count`*`)`

Specifies the number of SDDF session checks (runs) the utility should perform.

Default: 2

The utility can be run in two situations:

- When SRDF/A and MSC are expected to function normally. In this case, run with N(2). This allows the utility to verify that the SDDF session time stamps change over time when compared to the first run. Any higher number can be specified, thus allowing the utility to run for longer periods of time, for example, during MSC activation or termination, monitoring progress of the operation.
- When SRDF/A is known to be dropped. In this case, run with N(1). There is no point looking for changes in SDDF session time stamps in this context.

`P(`*`current-primary-site`*`)`

Specifies the current primary DASD site. The value specified must be DC1, DC2, DC3, or DC4, and the named site must be a GDDR-configured site.

Default: The current primary DASD site currently known to GDDR.

It is recommended to use the default. This argument is intended for lab environments. It should only be necessary to deviate from the default in a transient state of the configuration, when GDDR variables do not properly reflect the site roles you want to check SDDF sessions for.

## Sample output

This section provides sample reports produced by the GDDR SDDF Session Verification utility.

**Note:** These sample reports have been edited for brevity.

The following GDDR script joblog shows MSC,REFRESH processing in preparation of MSC activation.

```
For context: Log of an MSC Activation occurring during a GDDR script.
15.57.46 J
OB00909  +GDDR705I GDDR Rerun state checker has initialized GLOBAL.GDDR.RUN.GDDRPA29.GDDRMSCP_RESREFR_DC1_STAR_.RC
15.57.46 JOB00909  +GDDR721I GDDR Starting Activate SRDF/Star on Primary MSC Server for System B30D
15.57.46 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE INITIAL PRIMARY ???? STARFIRE  012
   012              CONCURRENT INIT_ERROR SITEAV41 NNN NOT_STARTED NNNNYNNN00 ????
15.57.46 JOB00909  +GDDR721I GDDR Starting Issue MSC Command (REFRESH)
15.57.46 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE INITIAL PRIMARY ???? STARFIRE  014
   014              CONCURRENT INIT_ERROR SITEAV41 NNN NOT_STARTED NNNNYNNN00 ????
15.57.46 JOB00909 +GDDB046I GDDR Waiting for MSC,REFRESH to complete for group SRDFMSC1. Please Reply (Yes/Cancel).
15.57.46 JOB00909 @64 GDDB046I GDDR Waiting for MSC,REFRESH to complete for group SRDFMSC1. Please Reply
(Yes/Cancel).
15.57.46 JOB00909  +GDDR670I GDDR Issuing command F GDDRSCF,MSC,REFRESH,MSCG(SRDFMSC1)
15.58.05 JOB00909  +GDDR818I MSC State for SRDFMSC1: ENABLED NOTFOUND NOTFOUND ???? ???? ???? ???? ???? NNN
NOT_STARTED ???? ????
15.58.05 JOB00909  +GDDR639I GDDR Completed Issue MSC Command with rc 0
```

The following GDDR script joblog shows activation of MSC, GLOBAL,PARM_REFRESH processing.

```
FMSC Activation, continued
15.58.05 JOB00909  +GDDR721I GDDR Starting Issue MSC Command (PARM_REFRESH)
15.58.05 JOB00909  +GDDR199I GDDR Using program IEBGENER as IEBGENER utility
15.58.05 JOB00909  +GDDR718I GDDR Source member SITEDC1 copied into target member SRDFSTAR
15.58.05 JOB00909  +GDDR818I MSC State for SRDFMSC1: ENABLED NOTFOUND NOTFOUND ???? ???? ???? ???? ???? NNN
NOT_STARTED ???? ????
15.58.05 JOB00909  +GDDB045I GDDR Waiting for PARM_REFRESH to complete for group SRDFMSC1. Please Reply
(Yes/Cancel).
15.58.05 JOB00909 @65 GDDB045I GDDR Waiting for PARM_REFRESH to complete for group SRDFMSC1. Please Reply
(Yes/Cancel).
15.58.05 JOB00909  +GDDR670I GDDR Issuing command SC GLOBAL,PARM_REFRESH
15.58.29 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE DISABLED PRIMARY ???? STAR CONCURRENT VAL_IP SITEAV41
NNN NOT_STARTED
                        NNNYNNNN00 ????
15.58.48 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE DISABLED PRIMARY ???? STARFIRE  043
   043            CONCURRENT VAL_IP SITEAV41 NNN NOT_STARTED NNNNYNNN00 ????
15.59.07 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE INITIAL PRIMARY ???? STARFIRE  044
   044           CONCURRENT NOT_STARTED SITEAV41 NNN NOT_STARTED NNNNYNNN00 ????
15.59.32 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE ACTIVE PRIMARY B30D STARFIRE CONCURRENT ???? SITEAV41
YYY NOT_STARTED
                        YYNNYYYY00 ????
16.00.24 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE ACTIVE PRIMARY B30D STARFIRE CONCURRENT FC01 SITEAV41
YYY NOT_STARTED
                        YYNNYYYY00 ????
16.00.28 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE ACTIVE PRIMARY B30D STARFIRE CONCURRENT FC02 SITEAV41
YYY NOT_STARTED
                        YYNNYYYY80 ????
16.00.32 JOB00909  +GDDR818I MSC State for SRDFMSC1: SINGLE ACTIVE PRIMARY B30D STARFIRE CONCURRENT FC03 SITEAV41
YYY NOT_STARTED
                        YYYNYYYY80 ????
16.00.32 JOB00909  +GDDR639I GDDR Completed Issue MSC Command with rc 0
16.00.33 JOB00909  +GDDR639I GDDR Completed Activate SRDF/Star on Primary MSC Server with rc 0
```

The following GDDRSDDF joblog shows run 67 of 100 calling out full push risk of 32 devices, ending with RC=16. It shows typical error messages issued by the utility when run concurrently with MSC activation. All messages shown here are expected during MSC activation.

```
GDDRSDDF joblog

15.59.50 JOB01211  +GDDP440I GDDRSDDF Starting SDDF session verification, run 67 of 100
15.59.50 JOB01211  +GDDP448E Found 1 SDDF sessions  for device 01A6 in Symmetrix 0001957-00578 at DC2. Expected 2
15.59.50 JOB01211  +GDDP448E Full PUSH risk for device 01A6 in Symmetrix 0001957-00578 at DC2
15.59.50 JOB01211  +GDDP448E Found 1 SDDF sessions  for device 01A7 in Symmetrix 0001957-00578 at DC2. Expected 2
15.59.50 JOB01211  +GDDP448E Full PUSH risk for device 01A7 in Symmetrix 0001957-00578 at DC2

15.59.50 JOB01211  +GDDP448E Found 1 SDDF sessions  for device 01B6 in Symmetrix 0001957-00578 at DC2. Expected 2
15.59.50 JOB01211  +GDDP448E Full PUSH risk for device 01B6 in Symmetrix 0001957-00578 at DC2
15.59.50 JOB01211  GDDM031E Error on EMCSAI SDDFGETB, return code 24, EMCRC 001C, EMCRS 0018, EMCRCX 17001806
15.59.50 JOB01211  +GDDP449W Unchanged SDDF Session Time stamps for device 01B7 in Symmetrix 0001957-00578 at DC2
15.59.50 JOB01211  GDDM031E Error on EMCSAI SDDFGETB, return code 24, EMCRC 001C, EMCRS 0018, EMCRCX 17001806
15.59.50 JOB01211  +GDDP449W Unchanged SDDF Session Time stamps for device 01B8 in Symmetrix 0001957-00578 at DC2
15.59.50 JOB01211  GDDM031E Error on EMCSAI SDDFGETB, return code 24, EMCRC 001C, EMCRS 0018, EMCRCX 17001806
15.59.50 JOB01211  +GDDP449W Unchanged SDDF Session Time stamps for device 01B9 in Symmetrix 0001957-00578 at DC2

15.59.51 JOB01211  +GDDP449W Unchanged SDDF Session Time stamps for device 01D5 in Symmetrix 0001957-00578 at DC2
15.59.51 JOB01211  +GDDP448E Found 1 SDDF sessions  for device 0536 in Symmetrix 0001957-00578 at DC2. Expected 2
15.59.51 JOB01211  +GDDP448E Full PUSH risk for device 0536 in Symmetrix 0001957-00578 at DC2
15.59.51 JOB01211  +GDDP448E Found 1 SDDF sessions  for device 0537 in Symmetrix 0001957-00578 at DC2. Expected 2
15.59.51 JOB01211  +GDDP448E Full PUSH risk for device 0537 in Symmetrix 0001957-00578 at DC2

15.59.52 JOB01211  +GDDP453E GDDRSDDF considers 32 devices at risk for a Full PUSH
15.59.52 JOB01211  +GDDP441I GDDRSDDF Completed SDDF session verification with RC=16, run 67 of 100
```

The following GDDRSDDF joblog shows run 68 of 100 calling out full push risk of 128 devices, ending with RC=16. It shows typical error messages issued by the utility when run concurrently with MSC activation.

```
GDDRSDDF joblog

16.00.16 JOB01211  +GDDP440I GDDRSDDF Starting SDDF session verification, run 68 of 100
16.00.17 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01A6 in Symmetrix
0001957-00600 at DC3.
16.00.17 JOB01211  +         Expected 3339
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 01A6 in Symmetrix 0001957-00578 at DC2
16.00.17 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01A7 in Symmetrix
0001957-00600 at DC3.
16.00.17 JOB01211  +        Expected 3339
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 01A7 in Symmetrix 0001957-00578 at DC2
16.00.17 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01A8 in Symmetrix
0001957-00600 at DC3.
16.00.17 JOB01211  +        Expected 3339
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 01A8 in Symmetrix 0001957-00578 at DC2
16.00.17 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01A9 in Symmetrix
0001957-00600 at DC3.
16.00.17 JOB01211  +        Expected 3339
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 01A9 in Symmetrix 0001957-00578 at DC2

16.00.17 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01D5 in Symmetrix
0001957-00600 at DC3.
16.00.17 JOB01211  +        Expected 3339
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 01D5 in Symmetrix 0001957-00578 at DC2
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 0536 in Symmetrix 0001957-00578 at DC2
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 0537 in Symmetrix 0001957-00578 at DC2
16.00.17 JOB01211  +GDDP448E Full PUSH risk for device 0538 in Symmetrix 0001957-00578 at DC2
16.00.18 JOB01211  +GDDP453E GDDRSDDF considers 128 devices at risk for a Full PUSH
16.00.18 JOB01211  +GDDP441I GDDRSDDF Completed SDDF session verification with RC=16, run 68 of 100
```

The following GDDRSDDF joblog shows run 69 of 100 calling out full push risk of 128 devices, ending with RC=16 and run 70 of 100 with RC=0. It shows GDDRSDDF error messages during the final stages of MSC activation, as well as the first GDDRSDDF run after that which finds no errors.

```
GDDRSDDF joblog


16.00.42 JOB01211  +GDDP440I GDDRSDDF Starting SDDF session verification, run 69 of 100
16.00.42 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01A6 in Symmetrix
0001957-00600 at DC3.
16.00.42 JOB01211  +       Expected 3339
16.00.42 JOB01211  +GDDP448E Full PUSH risk for device 01A6 in Symmetrix 0001957-00578 at DC2
16.00.42 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 01A7 in Symmetrix
0001957-00600 at DC3.
16.00.42 JOB01211  +       Expected 3339
16.00.42 JOB01211  +GDDP448E Full PUSH risk for device 01A7 in Symmetrix 0001957-00578 at DC2
16.00.44 JOB01211  +GDDP448E Invalid cylinder count 1 for SDDF session B4FF for device 0585 in Symmetrix
0001957-00600 at DC3.
16.00.44 JOB01211  +       Expected 3339
16.00.44 JOB01211  +GDDP448E Full PUSH risk for device 0585 in Symmetrix 0001957-00578 at DC2

16.00.44 JOB01211  +GDDP453E GDDRSDDF considers 128 devices at risk for a Full PUSH
16.00.44 JOB01211  +GDDP441I GDDRSDDF Completed SDDF session verification with RC=16, run 69 of 100


16.01.08 JOB01211  +GDDP440I GDDRSDDF Starting SDDF session verification, run 70 of 100
16.01.10 JOB01211  +GDDP441I GDDRSDDF Completed SDDF session verification with RC=0, run 70 of 100
```

The following output shows partial GDDR SDDF Session Verification utility sysout from the SDDFLIST and the DIFFLIST DD statements. Many fields are shown with all "-" signs, indicating that expected session information was not found.

```
GDDRSDDF SDDFLIST DD card

GDDP444I GDDRSDDF SDDF Session information for GDDR managed devices (67/100)
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA31  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP446I  Device    CYLS     ID  Action  State     Time stamp     Remark     Device    CYLS     ID  Action  State     Time stamp
GDDP447I 000001A6 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A6 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001A7 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A7 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001A8 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A8 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001A9 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A9 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001AA 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001AA 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------

GDDP444I GDDRSDDF SDDF Session information for GDDR managed devices (68/100)
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA31  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP446I  Device    CYLS     ID  Action  State     Time stamp     Remark     Device    CYLS     ID  Action  State     Time stamp
GDDP447I 000001A6 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001A6 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001A7 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001A7 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001A8 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001A8 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001A9 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001A9 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001AA 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001AA 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001AB 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001AB 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
...
GDDP447I 000001D5 000000001 B4FF DEACTI FINISHED 20130812160003        CYL  000001D5 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA3F  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP446I  Device    CYLS     ID  Action  State     Time stamp     Remark     Device    CYLS     ID  Action  State     Time stamp
GDDP447I 00000536 ---------- ---- ------ -------- --------------  (2)   (1)  00000536 ---------- ---- ------ -------- --------------
GDDP447I          ---------- ---- ------ -------- --------------
GDDP447I 00000537 ---------- ---- ------ -------- --------------  (2)   (1)  00000537 ---------- ---- ------ -------- --------------
GDDP447I          ---------- ---- ------ -------- --------------
GDDP447I 00000538 ---------- ---- ------ -------- --------------  (2)   (1)  00000538 ---------- ---- ------ -------- --------------
GDDP447I          ---------- ---- ------ -------- --------------
```

The following output shows GDDR SDDF Session Verification utility sysout from the SDDFLIST DD statement. It shows typical output from the utility during MSC activation. Many fields are shown with all "-" signs, indicating that expected session information was not found.

```
GDDRSDDF SDDFLIST DD card

GDDP444I GDDRSDDF SDDF Session information for GDDR managed devices (67/100)
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA31  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP446I  Device    CYLS     ID  Action  State     Time stamp     Remark     Device    CYLS     ID  Action  State     Time stamp
GDDP447I 000001A6 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A6 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001A7 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A7 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001A8 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A8 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001A9 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001A9 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------
GDDP447I 000001AA 000003339 B4FE RESET  FINISHED 20130812153644  (2)        000001AA 000003339 B4FF RESET  FINISHED 20130812142514
GDDP447I          --------- ---- ------ -------- --------------

GDDP444I GDDRSDDF SDDF Session information for GDDR managed devices (68/100)
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA31  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP446I  Device    CYLS     ID  Action  State     Time stamp     Remark     Device    CYLS     ID  Action  State     Time stamp
GDDP447I 000001A6 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001A6 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001A7 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001A7 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001A8 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001A8 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001A9 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001A9 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001AA 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001AA 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP447I 000001AB 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001AB 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
...
GDDP447I 000001D5 000000001 B4FF DEACTI FINISHED 20130812160003       CYL  000001D5 000000001 B4FF DEACTI FINISHED 20130812160003
GDDP447I          000000001 B4FE DEACTI FINISHED 20130812160003
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA3F  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP446I  Device    CYLS     ID  Action  State     Time stamp     Remark     Device    CYLS     ID  Action  State     Time stamp
GDDP447I 00000536 ---------- ---- ------ -------- --------------  (2)   (1)  00000536 ---------- ---- ------ -------- --------------
GDDP447I          ---------- ---- ------ -------- --------------
GDDP447I 00000537 ---------- ---- ------ -------- --------------  (2)   (1)  00000537 ---------- ---- ------ -------- --------------
GDDP447I          ---------- ---- ------ -------- --------------
GDDP447I 00000538 ---------- ---- ------ -------- --------------  (2)   (1)  00000538 ---------- ---- ------ -------- --------------
GDDP447I          ---------- ---- ------ -------- --------------
```

The following output shows partial GDDR SDDF Session Verification utility sysout from the DIFFLIST DD statement. It shows typical output from the utility during MSC activation. Many fields are shown with all "-" signs, indicating that expected session information was not found.

```
GDDRSDDF DIFFLIST DD card continued

GDDP450I GDDRSDDF SDDF Session information for GDDR managed devices (67/100)
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA31  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP451I  Device    CYLS     ID    Tracks     Differential    Device    CYLS     ID    Tracks
GDDP452I 000001A6 000003339 B4FE ----------    0000050085    000001A6 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
GDDP452I 000001A7 000003339 B4FE ----------    0000050085    000001A7 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
GDDP452I 000001A8 000003339 B4FE ----------    0000050085    000001A8 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
GDDP452I 000001A9 000003339 B4FE ----------    0000050085    000001A9 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
GDDP452I 000001AA 000003339 B4FE ----------    0000050085    000001AA 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
...
GDDP452I 000001B7 000003339 34FF 0000000000    0000000000    000001B7 000003339 B4FF 0000000000
GDDP452I          000003339 B4FE 0000000000
GDDP452I 000001B8 000003339 34FF 0000000000    0000000000    000001B8 000003339 B4FF 0000000000
GDDP452I          000003339 B4FE 0000000000
GDDP452I 000001B9 000003339 34FF 0000000000    0000000000    000001B9 000003339 B4FF 0000000000
GDDP452I          000003339 B4FE 0000000000
GDDP452I 000001BA 000003339 34FF 0000000000    0000000000    000001BA 000003339 B4FF 0000000000
GDDP452I          000003339 B4FE 0000000000
GDDP445I Symmetrix 0001957-00578 (5876 - 7C0F - DC2 - SiteB) <--  RA3F  --> Symmetrix 0001957-00600 (5876 - 8B0F - DC3 - SiteC)
GDDP451I  Device    CYLS     ID    Tracks     Differential    Device    CYLS     ID    Tracks
GDDP452I 00000536 000003339 B4FE ----------    0000050085    00000536 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
GDDP452I 00000537 000003339 B4FE ----------    0000050085    00000537 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
GDDP452I 00000538 000003339 B4FE ----------    0000050085    00000538 000003339 B4FF ----------
GDDP452I          --------- ---- ----------
```

### Return codes

See message GDDP441I in the *GDDR Message Guide*.

# GDDR SRDF Device Status Check Utility (GDDRSC06)

The GDDR SRDF Device Status Check utility (GDDRSC06) queries the SRDF device status for selected sites and SRDF groups. This may be helpful in identifying issues in the SRDF replication groups impacting script execution. The utility can also verify that devices are in the required replication status.

### Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRSC06).

```
//GDDRSC06 JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//              CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRSC06 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRSC06 arguments
/*
```

### Arguments

The syntax is as follows:

```
GDDRRXST GDDRSC06 DCn function-code srdf-type
T(device-type) L(linked-site)
```

Where:

DC*n*

Specifies the site ID (DC1, DC2, DC3, or DC4).

*function-code*

Type one of the following function codes:

- 03—Verifies that devices are R1s on the specified SRDF leg
- 04—Verifies that devices are R2s on the specified SRDF leg
- 06—Verifies that R1 devices are not TNR on the specified SRDF leg
- 07—Verifies that R1 devices are TRN on the specified SRDF leg
- 08—Verifies that devices are R21s

*srdf-type*

Type one of the following SRDF type codes (available depending on the GDDR configuration):

- J0—SRDF/S
- JA—SRDF/A

L(*linked-site)*

Specifies the SRDF-paired site (DC1, DC2, DC3, or DC4) for which you want to query the devices. This is a required parameter.

T(*device-type*)

Specify one of the following device types:

- I for internal devices (consistency protected)
- E for external, ADCOPY-DISK devices
- A for all devices, internal and external

## Example

```
GDDRRXST GDDRSC06 DC1 03 J0
```

# GDDR TimeFinder Management utility (GDDRTF20)

The GDDR TimeFinder Management utility allows you to perform GDDR local replication management functions using TimeFinder/SnapVX, TimeFinder/Clone, and TimeFinder/Mirror. This includes creating a point-in-time copy of the data at a selected site and performing LINK and UNLINK operations for SnapVX target devices.

The GDDR TimeFinder Management utility dynamically determines which technology applies to each storage system at the target site based on the choices made in the Define GDDR Configuration features panel (M,P,C,F), as described in "Define configuration features (M,P,C,F)" on page 302. The utility supports the following local replication technologies:

- TimeFinder/SnapVX with the RE-ESTABLISH, SPLIT, SPLITINSTANT, LINK, and UNLINK operations
- TimeFinder/Clone and TimeFinder/Mirror with the RE-ESTABLISH, SPLIT, and SPLITINSTANT operations

The GDDR TimeFinder Management utility supports creation of consistent point-in-time copies across Dell EMC storage systems of different generations. This can be done in two ways:

- Use SnapVX on PowerMax and VMAX3 systems and TF/Clone on VMAX1/2 systems. In this case, the utility generates the required CREATE and SNAP VOLUME commands followed by a single ACTIVATE command in the same QCINPUT.
- Use TF/Mirror across all system generations

## Requirements

It is recommended to run GDDRTF20 at a site with channel connectivity to the storage systems being managed.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRTF20).

```
//GDDRTF20 JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRKF20 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRKF20 arguments
/*
```

## Arguments

The syntax is as follows:

```
GDDRRXST GDDRKF20 C(operation) A(DCn) L(DCm)
[T({I|E|A})] +
[S({GOLD|TEST})] +
[H({CKD|FBA|ALL})] +
```

```
[DLM({YES|NO|ONLY})] +
[LINK({Y|N})] +
[WAIT(N)] +
[METH(method)] +
[SIMULATE({Y|N})] +
[FREEUNLK({Y|N})] +
[WDF({Y|N})]
```

## Required arguments

A(DC*n*)

This required parameter specifies the site where the managed devices are located.

C(*operation*)

This required parameter specifies the action to perform. Depending on the local replication technology selected for the managed storage system, the utility performs TimeFinder commands based on the entered *action* as follows:

**Table 38** GDDRTF20 operations and TimeFinder commands

| *operation* | **Local replication technology** | **TimeFinder command** |
|---|---|---|
| RE-ESTABLISH | SnapVX | UNLINK(TARGET(SYMDV#(*symbcv#*)) ... READY(NO))<br>**Note:** No operation performed for unlinked devices |
| | TF/Clone | CONFIG(TARGET(SYMDV#(*symbcv#*)) ... READY(NO) ) |
| | TF/Mirror | RE-ESTABLISH<br>**Note:** No operation performed for ESTABLISHed devices |

**Table 38** GDDRTF20 operations and TimeFinder commands

| *operation* | **Local replication technology** | **TimeFinder command** |
|---|---|---|
| SPLIT and SPLITINSTANT<br><br>**Note:** The GDDRTF20 SPLIT action creates a consistent point-in-time image. The SPLITINSTANT action does not guarantee consistency. | TF/Mirror | SPLIT (with the CONSISTENT option if necessary)<br><br>**Note:** The devices must be ESTABLISHed. |
| | TF/Clone | SNAP VOLUME (SOURCE(SYMDV#(*symstd#*)) TARGET(SYMDV#(*symbcv#*)) ... READY(YES)) |
| | SnapVX | TERMINATE (NAME(*snapshotname*) SOURCE(SYMDV#(*symstd#*)) ... AUTO_UNLINK(YES) TERMINATE_ALL(NO))<br><br>**Note:** No operation performed if the snapshot is already terminated<br><br>CREATE (NAME(*snapshotname*) SOURCE(SYMDV#(*symstd#*)) ... ) |
| | TF/Clone and SnapVX | ACTIVATE |
| | SnapVX | SOURCE(SYMDV#(*symstd#*)) TARGET(SYMDV#(*symbcv#*)) ... READY(YES))<br><br>**Note:** LINK is not performed by default, only with the optional LINK(Y) argument specified. |
| LINK and UNLINK | SnapVX on VMAX3 and later systems | If target device linking is not done at the time of a SPLIT command (which is the default behavior), you can LINK the devices afterwards. This allows you to use the snapshots created earlier. When you no longer need the linked devices, you can unlink them with the UNLINK command.<br><br>**Note:** The LINK command requires an activated GDDR-managed snapshot on the GDDR-managed source devices. |
| TERMINATE | VMAX3 and later systems | If a GDDR script is abandoned, it may be necessary to issue a TERMINATE command to free up SRP resources consumed by snapshots created by the script. |

`FREEUNLK({Y|N})`

Determines whether to start (Y) or not (N) a background process to free any SRP resources consumed by previously linked target devices for the affected snapshot name.

The default value is N.

This argument applies to the UNLINK and TERMINATE commands of SnapVX, as well as to RE-ESTABLISH commands in SnapVX configurations.

**Note:** The target devices can be relinked only after the free process completes.

When FREEUNLK(N) is specified, the WDF parameter is forced to N.

L(DC*m*)

This required parameter specifies the site that has an SRDF device pairing with the managed devices.

**Note:** GDDR management of local replication is applicable to GDDR-managed SRDF devices.

## Optional arguments

D({YES|NO|ONLY})

Determines whether to verify DLm backend devices. The default value is YES.

H({CKD|FBA|ALL})

Determines the type of BCV devices to verify: CKD, FBA, or ALL (default).

LINK({Y|N})

Determines whether to link target devices at the end of the SPLIT or SPLITINSTANT operation. The default value is N.

This parameter applies only to SPLIT or SPLITINSTANT operations run against a PowerMax or VMAX3 storage system.

METH(*method*)

Specifies the local replication technology: SnapVX, TF/Clone, or TF/Mirror.

This parameter can be specified as SNVX, CLON, MIRR or any combination of these values (a blank or comma-separated list).

Tis parameter is intended for testing purposes. By default, all methods are used, as applicable in the defined configuration.

When performing LINK, UNLINK, or TERMINATE operations, either do not specify this parameter, or specify M(SNVX).

S({GOLD|TEST})

Determines whether to verify the GOLD set or the test set of BCVs. The default value is GOLD.

**Note:** "GOLD and TEST sets of BCVs" on page 75 describes the BCV sets.

SIMULATE({Y|N})

Runs the utility in validation mode, including syntax verification of arguments and device-state checking by querying the storage systems. The generated commands are echoed to SYSTSPRT in the job log. However, EMCTF is not invoked and EMCSNAP is invoked with GLOBAL TYPRUN(SCAN). The state of the TimeFinder devices is not changed.

T({I|E|A})

Determines whether to manage BCVs attached to internal SRDF devices, external SRDF devices, or both.

Valid values are:

- I— (**Default**) Manage BCVs for internal SRDF devices only
- E—Manage BCVs for external SRDF devices only
- A—Manage BCVs for both internal and external SRDF devices

`WFD({Y|N})`

The WFD (WAIT_FOR_DEFINITION) option determines whether SnapVX is waiting (Y) or not (N) for target volume tracks to be fully defined before proceeding with unlinking.

See the *TimeFinder SnapVX and zDP Product Guide* for detailed information about this option.

The default value is N.

This argument applies to the UNLINK and TERMINATE commands of SnapVX, as well as to RE-ESTABLISH commands in SnapVX configurations.

When FREEUNLK(N) is specified, the WDF parameter is forced to N.

## Examples

- In preparation to business testing at DC3, data will be refreshed towards the TF/Mirror target devices, TF/Clone target devices will be made NR, and SnapVX target devices will be unlinked and made NR. The TEST set of target devices will be affected, and both internal and external targets will be included. When using TF/Mirror, a RE-ESTABLISH is required before a SPLIT can be attempted. In this example, local replication management is limited to devices paired in SRDF with DC1. Depending on the configuration, a second run may be required specifying DC2 as the linked site.

  ```
  GDDRRXST GDDRKF20 C(RE-ESTABLISH) A(DC3) L(DC1) S(TEST) T(A)
  ```

- Anytime after the RE-ESTABLISH command above, a consistent point-in-time copy can be created. If SnapVX is used in this configuration, target devices will be linked.

  ```
  GDDRRXST GDDRKF20 C(SPLIT) A(DC3) L(DC1) S(TEST) T(A) LINK(Y)
  ```

- In a configuration where SnapVX is used, target devices will be linked to a GDDR-managed snapshots activated earlier. If the GDDR-managed snapshot name for TEST set devices is not found activated for the source devices, an error is produced.

  ```
  GDDRRXST GDDRKF20 C(LINK) A(DC3) L(DC1) S(TEST) T(A)
  ```

- This example can be used in lab environments to create a point-in-time copy (consistency not guaranteed) including only TF/Clone target devices in the TEST set.

  ```
  GDDRRXST GDDRKF20 C(SPLITINSTANT) A(DC3) L(DC1) S(TEST) M(CLON)
  ```

# GDDR BCV Reporting utility (GDDRTF00)

The GDDR BCV Reporting utility checks BCV status. Run the GDDR BCV Reporting utility to ensure that all BCVs are in one of the expected states.

The utility dynamically determines which local replication technology applies to each storage system at the queried site based on the choices made in the Define GDDR Configuration features panel (M,P,C,F), as described in "Define configuration features (M,P,C,F)" on page 302.

**Note:** "TimeFinder device states" on page 76 lists states of TimeFinder devices supported by GDDR.

## Requirements

It is recommended to run this utility at a site which has channel connectivity to the storage systems being queried.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRTF00).

```
//GDDRTF00 JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRTF00 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRTF00 arguments
/*
```

## Arguments

The syntax is as follows:

```
GDDRRXST GDDRTF00 B(DCn) Q(statelist) +
[R(DCm)] +
[T({I|E|A})] +
[S({GOLD|TEST})] +
[H({CKD|FBA|ALL})] +
[DLM({YES|NO|ONLY})] +
[PCHECK({YES|NO})] +
```

Where:

`B(DC`*n*`)` *

This required parameter indicates the site where the BCVs are located.

`D({YES|NO|ONLY})`

Determines whether to verify DLm backend devices. The default value is YES.

`H({CKD|FBA|ALL})`

Determines the type of BCV devices to verify: CKD, FBA, or ALL (default).

`PCHECK({YES|NO})`

Determines whether to verify that all included source devices are protected by a local replication target device or snapshot, and that the device found paired is also

`Q(statelist)`

This required parameter lists states that you consider acceptable, depending on the local replication technology and device type (source or target). Possible values are: ACTIVATE, CREATE, ESTABLISH, LINK, RESTORE, SPLIT, UNLINK.

See "TimeFinder device states" on page 76 for detailed explanation of these values and usage examples.

`R(DC`*m*`)` *

This parameter indicates the remote site which is SRDF-paired with the site specified in B(DC*n*).

In SRDF/Star configurations, SRDF/A devices paired DC1-DC3 or DC2-DC3 are possibly a subset of SRDF/S devices paired DC1-DC2. The R(DC*m*) parameter helps determine the devices to be verified. The default value is the SRDF/S partner site.

When verifying external BCV devices, R(DC*m*) is a required parameter if the GDDR configuration includes more than two sites.

`S({GOLD|TEST})`

Determines whether to verify the GOLD set or the test set of BCVs. The default value is GOLD.

**Note:** "GOLD and TEST sets of BCVs" on page 75 describes the BCV sets.

`T({I|E|A})`

Determines whether to verify BCVs attached to internal SRDF devices, external SRDF devices, or both.

Valid values are:

- I— (**Default**) Verify BCVs for internal SRDF devices only
- E—Verify BCVs for external SRDF devices only
- A—Verify BCVs for both internal and external SRDF devices

## Examples

- To verify whether the gold set of internal BCV devices at DC1, attached to SRDF devices paired with DC2, are currently established:

```
GDDRRXST GDDRTF00 B(DC1) Q(ESTABLISH) R(DC2)
```

- To verify whether the test set of external BCV devices at DC3, attached to SRDF devices paired with DC2, are currently established:

```
GDDRRXST GDDRTF00 B(DC3) Q(ESTABLISH) R(DC2) T(E) S(TEST)
```

# BCVGROUP Validation utility (GDDRBCVG)

The BCVGROUP Validation utility verifies the contents of the BCVGROUP dataset against the GDDR RDF.DEVICES and STDBCV parameters.

**Note:** The BCVGROUP dataset is used to implement the GDDR Multi-Tenancy for TimeFinder feature described in "GDDR multi-tenancy for TimeFinder" on page 79.

It is recommended to run this utility each time the BCVGROUP dataset is modified.

The GDDRBCVG utility is also called automatically during script generation of the Test IPL from BCVs script if you have indicated the intent to use the GDDR Multi-Tenancy for TimeFinder feature. During script generation, BCVGROUP dataset validation is limited to the systems selected for the test IPL and to devices at the target site for the script.

## Prerequisites

- Complete the steps described in"Configure GDDR Multi-Tenancy for TimeFinder" on page 158.
- GDDRMAIN must be up and running.
- A GDDR Parameter Wizard session must have successfully activated a GDDR configuration describing the SRDF devices and their local replication targets as managed by GDDR.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRBCVG).

Point the GDDRPROC JCLLIB card to the customized GDDRPROC. This is the dataset identified as PROCLIB in the Define GDDR Datasets panel (M,P,C,D).

```
//GDDRBCVG JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRBCVG EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRBCVG
/*
```

## Return codes

Table 39 lists the GDDRBCVG utility return codes.

**Table 39** GDDRBCVG return codes

| Return code | Description |
|---|---|
| 0 | Validation completed successfully/ |
| 12 | Validation error. |

Other additional return codes can be obtained from invoked services.

# GDDR Invalid Track Monitor utility (GDDRMIN0)

Run the GDDR Invalid Track Monitor (GDDRMIN0) to query GDDR-managed SRDF or TimeFinder devices for invalid tracks.

**Note:** The query capability for TimeFinder devices, only verifies TimeFinder target devices for local invalid tracks. It cannot be used to query the STD device to TGT device differential.

The GDDR Invalid Track Monitor is a query-only utility; it does not affect the state of the GDDR-managed configuration.

The GDDR Invalid Track Monitor can run on any site in the configuration, but it is recommended to run at the site where the queried devices are located. Since typically the GDDR Invalid Track Monitor is used during a massive resynchronization over SRDF links, running at the queried site avoids impact from the data traffic on the queries.

## Requirements

- SCF must be up and running with the SCF$*nnnn* connector as specified in your customized GDDRPROC JCL, which is pointed to by the JCLLIB statement.
- GDDRMAIN must be up and running, and connected with the same SCF instance.
- RDF.DEVICES parameters reflecting site pairs as expected for the defined type of configuration.
- BCV parameters reflecting the sites in the configuration are expected if TimeFinder devices are to be queried.
- Channel and SRDF group connectivity to the storage systems relevant to the queries.

## Sample JCL

The sample GDDRMIN0 JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRMIN0).

```
//GDDRMIN0 JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRMIN0 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRMIN0 S(DC1) R(DC3) T(I) W(30000,RAGROUP,3) I(15)
/*
```

## Arguments

The syntax is as follows:

```
{S(DCn) {L|R}(DCm) | B(DCn) [SET(GOLD|TEST)]}
[T({I|E|A})]
[{N(n)|W(inv-trck-count,objtype,stallmax)}]
[I(interval)]
```

## Required arguments

B(DC*n*) [SET(GOLD|TEST)]

Indicates that TimeFinder devices are to be queried at the DC*n* site.

**Note:** Specify either S(DC*n*) or B(DC*n*), but not both.

With B(DC*n*), only local invalid track queries are supported.

SET(GOLD|TEST)

This optional argument determines whether to query the GOLD (default) or TEST set of TimeFinder devices. This argument can only be used with B(DCn).

S(DC*n*)

Indicates that SRDF devices are to be queried at the DC*n* site.

**Note:** Specify either S(DC*n*) or B(DC*n*), but not both.

With S(DCn), the {L|R}(DC*m*) argument is required.

{L|R}(DCm)

This argument is required when S(DC*n*) is specified. It helps determine the type of devices and the site to be queried:

- Specify L to indicate the SRDF paired site when querying TimeFinder devices associated with external devices (local invalid track queries).
- Specify R to query devices that are local to the site specified with S(DC*n*) in their mirror positions reflecting the partner devices (remote invalid track queries).

  **Note:** R does not mean querying remote devices.

- DC*m* indicates the paired site.

## Optional arguments

I(*interval*)

This optional argument determines the number of minutes in between runs. The default number is 15.

This argument is ignored if N is set (or defaults) to 1.

N(*n*)|W(*inv-trck-count,objtype,stallmax*)

This optional argument determines how long the GDDR Invalid Track Monitor should run.

If N or W is not specified, the default value of N(1) is used.

N(*n*)

Indicates a fixed number of runs. *n* can be any integer. The default value is 1.

`W(inv-trck-count,objtype,stallmax)`

Sets a resynchronization goal for the GDDR Invalid Track Monitor to achieve.

**Note:** `W` is ignored if `N` is specified.

*inv-trck-count*

The number of invalid tracks which allows the GDDR Invalid Track Monitor to terminate with RC=0.

*objtype*

The type of object to which the *inv-trck-count* value applies. It can be specified as one of the following: `SITe`, `SYMmetrix`, `RAGroup`, `RANge`, `DEVice`.

*stallmax*

The number of consecutive runs during which there is no progress in total resynchronization. When this value is exceeded, the GDDR Invalid Track Monitor terminates.

If `W` is used, the GDDR Invalid Track Monitor terminates when one of the following conditions is met:

- The resynchronization goal reached.
- The *stallmax* value exceeded.
- More than 99999 runs are required for the resynchronization goal to be reached.

`T(I|E|A)`

This optional argument determines whether to query Internal (`I`), External (`E`), or All (`A`) devices. The default value is `I`.

## Examples

- To determine if there any invalid tracks on the ConGroup-protected devices at DC1, owed to DC2:

  ```
  GDDRRXST GDDRMIN0 S(DC1) R(DC2) T(I) N(1)
  ```

  The GDDR Invalid Track Monitor queries all GDDR-managed internal devices at DC1 and reports any devices with invalid tracks owed to DC2. RC=0 is set if there are no invalid tracks owed to DC2.

- SRDF/A from DC1 to DC3 has been out for several days. There are invalid tracks at DC1. Resynchronization was just started. How long is it going to take, when is it going to be complete or stopped?

  ```
  GDDRRXST GDDRMIN0 S(DC1) R(DC3) W(0,RAG,5)
  ```

  The GDDR Invalid Track Monitor runs every 15 minutes, querying the DC1 devices for invalid tracks owed to DC3, until one of the following conditions is met:

  - All GDDR-managed SRDF groups DC1-DC3 have 0 invalid tracks.

- During six consecutive runs, the number of invalid tracks owed from DC1 to DC3 does not decrease.
- More than 99999 times/15 minutes elapse.

During each run, any SRDF group with invalid tracks is reported.

As of the second run, the GDDR Invalid Track Monitor produces a report at the SRDF group level, showing the site, storage system, SRDF group, invalid track count, resynchronization speed (tracks per second) and time required to completion (ETA).

Any SRDF group for which there is no resynchronization progress during any run is called out.

**Note:** If more detail than the SRDF group level is required, use the RANGE level.

- During execution of the GDDRPA07 script, GDDR has just done the RESUMEPAIR from DC2 to DC3 with KEEPR2, and resumed SRDF. The operator then accidentally canceled the script. Run the GDDRMIN0 utility to monitor resynchronization progress as follows:

```
GDDRRXST GDDRMIN0 S(DC3) L(DC2) W(0,RAG,3) I(5)
```

The GDDR Invalid Track Monitor runs every 5 minutes, querying the DC3 devices paired with DC2 for local invalid tracks, until one of the following conditions is met:

- All GDDR-managed SRDF groups at DC3-DC2 have 0 invalid tracks.
- During four consecutive runs, the number of local invalid tracks at DC3 does not decrease.
- More than 99999 times/5 minutes elapse.

During each run, any SRDF group with invalid tracks is reported.

As of the second run, the GDDR Invalid Track Monitor produces a report at the SRDF group level, showing the site, storage system, SRDF group, invalid track count, resynchronization speed (tracks per second) and time required to completion (ETA).

Any SRDF group for which there is no resynchronization progress during any run is called out.

**Note:** If more detail than the SRDF group level is required, use the RANGE level.

- To determine if there local invalid tracks on the BCVs at DC4:

```
GDDRRXST GDDRMIN0 B(DC4) N(1)
```

The GDDR Invalid Track Monitor queries the BCV devices at DC4 for local invalid tracks, one time, and exits with RC=0 if there are none.

Any GDDR-managed BCV device at DC4 with invalid tracks is reported.

## Return codes

Table 44 lists the GDDRMIN0 utility return codes.

**Table 40** GDDRMIN0 return codes

| Return code | Description |
|---|---|
| 0 | No objects have more than expected invalid tracks. |
| 8 | Some objects have more than expected invalid tracks. |
| 12 | The maximum stall value exceeded while waiting for a specified resynchronization goal. |

**Note:** See message GDDP474I in the *GDDR Message Guide* for more information about return codes.

# GDDR DLm Serial Number utility (GDDRDLSN)

The GDDR DLm Serial Number utility is used to list FICON serial numbers of all DLM devices (and DLM slaves, if any) defined to GDDR. The serial numbers are obtained from the DLm.

Run this utility if the 'GDDR' license is the only active license.

## Requirements

- SCF must be running with the SCF$*nnnn* connector specified in your customized GDDRPROC JCL, which is pointed to by the JCLLIB statement.
- Appropriate DLm code is required for the utility to work.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB (GDDRDLSN).

```
//GDDRDLSN JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),REGION=64M
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//GDDRDLSN EXEC GDDRPROC
//SYSTSIN  DD  *
  EXECUTIL SEARCHDD(YES)
  GDDRDLSN
//*
//
```

# GDDR IPL Parameter Swap utility (GDDRISWP)

The GDDR IPL Parameter Swap utility performs swaps of normal and alternate IPL addresses and load parameters for managed systems by site or by system while propagating the changes to all C-systems.

The utility executes on the master C-system as a batch job and swaps the normal and alternate IPL address and load parameter values for the following pairs of global variables:

| Normal | swapped with | Alternate |
|---|---|---|
| PARMS.IPL | | PARMS.IPLALT |
| PARMS.IPLBCVS | | PARMS.IPLBCVA |
| PARMS.IPLBCTS | | PARMS.IPLBCTA |

**Note:** The GDDR IPL Parameter Swap utility does not process IPL address and load parameters for C-systems. Only the global variable values for managed systems are swapped.

## Requirements

The utility must run on the master C-system.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB (GDDRISWP).

```
//GDDRISWP JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRISWP EXEC GDDRPROC,
//             PARM='GDDRISWP SYSTEM(*) SITE(*)'
//GDDTRACE DD  SYSOUT=*
//*
```

## Required EXEC parameters

The following required EXEC parameters specify the system and/or site to which the updates are applied:

`SYSTEM(`*name*`|*)`

System name or asterisk (*) for all systems.

`SITE(DC`*n*`|*)`

Site name or asterisk (*) for all sites.

## Sample SYSTSPRT output

Messages appear in the job's SYSTSPRT DD and are issued in the following sections:

- The SYSTEM and SITE parameters for this execution of the utility
- The values of the GDDR global variables at the time of job initiation
- The C-system IPL-related global variables that are excluded from processing

- The results of swapping the managed system load-address and load-parameter values
- The results of broadcasting the updated global variables to the other C-systems

**Note:** The example illustrates an SRDF/Star configuration.

```
Swapping Managed System primary and alternate IPL Parameters for
 System = *, Site = *

 Initial Global Variable values:

   GLOBAL.GDDR.PARMS.IPL.BTMUL144.DC2          : 94F1,0670MGM1
   GLOBAL.GDDR.PARMS.IPL.BTMUL150.DC1          : 94F1,0670MGM1
   GLOBAL.GDDR.PARMS.IPL.BTMUL48.DC3           : 94FF,0F13KGM1
   GLOBAL.GDDR.PARMS.IPL.BTMUL51.DC1           : 01111,1111STM1
   GLOBAL.GDDR.PARMS.IPLALT.BTMUL51.DC1        : 1234,1111STM1
   GLOBAL.GDDR.PARMS.IPLBCVS.BTMUL51.DC1       : 01111,1111BGM1
   GLOBAL.GDDR.PARMS.IPLBCVA.BTMUL51.DC1       : 03333,333333M3
   GLOBAL.GDDR.PARMS.IPLBCTS.BTMUL51.DC1       : 04444,444444M4

 C-System Global Variables excluded from processing:

   GLOBAL.GDDR.PARMS.IPL.BTMUL144.DC2          : 94F1,0670MGM1
   GLOBAL.GDDR.PARMS.IPL.BTMUL150.DC1          : 94F1,0670MGM1
   GLOBAL.GDDR.PARMS.IPL.BTMUL48.DC3           : 94FF,0F13KGM1

 Global Variable update results:

   GLOBAL.GDDR.PARMS.IPLALT.BTMUL51.DC1         set to 01111,1111STM1
   GLOBAL.GDDR.PARMS.IPL.BTMUL51.DC1            set to 1234,1111STM1
   GLOBAL.GDDR.PARMS.IPLBCVA.BTMUL51.DC1        set to 01111,1111BGM1
   GLOBAL.GDDR.PARMS.IPLBCVS.BTMUL51.DC1        set to 03333,333333M3

   Variable not found: GLOBAL.GDDR.PARMS.IPLBCTA.BTMUL51.DC1
   No changes made to: GLOBAL.GDDR.PARMS.IPLBCTS.BTMUL51.DC1
   Return code set to: 4

 Broadcasting updated Global Variables to DC1

GDDR373I GDDR Broadcasting PARMS.IPLALT.BTMUL51.DC1 = 01111,1111STM1
GDDR739I GDDR -> Set PARMS.IPLALT.BTMUL51.DC1 = 01111,1111STM1 at DC1
GDDR739I GDDR -> Set PARMS.IPLALT.BTMUL51.DC1 = 01111,1111STM1 at DC3
   Send of GLOBAL.GDDR.PARMS.IPLALT.BTMUL51.DC1         Completed
GDDR373I GDDR Broadcasting PARMS.IPL.BTMUL51.DC1 = 1234,1111STM1
GDDR739I GDDR -> Set PARMS.IPL.BTMUL51.DC1 = 1234,1111STM1 at DC1
GDDR739I GDDR -> Set PARMS.IPL.BTMUL51.DC1 = 1234,1111STM1 at DC3
   Send of GLOBAL.GDDR.PARMS.IPL.BTMUL51.DC1            Completed
GDDR373I GDDR Broadcasting PARMS.IPLBCVA.BTMUL51.DC1 = 01111,1111BGM1
GDDR739I GDDR -> Set PARMS.IPLBCVA.BTMUL51.DC1 = 01111,1111BGM1 at DC1
GDDR739I GDDR -> Set PARMS.IPLBCVA.BTMUL51.DC1 = 01111,1111BGM1 at DC3
   Send of GLOBAL.GDDR.PARMS.IPLBCVA.BTMUL51.DC1        Completed
GDDR373I GDDR Broadcasting PARMS.IPLBCVS.BTMUL51.DC1 = 03333,333333M3
GDDR739I GDDR -> Set PARMS.IPLBCVS.BTMUL51.DC1 = 03333,333333M3 at DC1
GDDR739I GDDR -> Set PARMS.IPLBCVS.BTMUL51.DC1 = 03333,333333M3 at DC3
   Send of GLOBAL.GDDR.PARMS.IPLBCVS.BTMUL51.DC1        Completed

 Broadcasting updated Global Variables to DC3

GDDR373I GDDR Broadcasting PARMS.IPLALT.BTMUL51.DC1 = 01111,1111STM1
GDDR739I GDDR -> Set PARMS.IPLALT.BTMUL51.DC1 = 01111,1111STM1 at DC1
GDDR739I GDDR -> Set PARMS.IPLALT.BTMUL51.DC1 = 01111,1111STM1 at DC3
   Send of GLOBAL.GDDR.PARMS.IPLALT.BTMUL51.DC1         Completed
GDDR373I GDDR Broadcasting PARMS.IPL.BTMUL51.DC1 = 1234,1111STM1
GDDR739I GDDR -> Set PARMS.IPL.BTMUL51.DC1 = 1234,1111STM1 at DC1
GDDR739I GDDR -> Set PARMS.IPL.BTMUL51.DC1 = 1234,1111STM1 at DC3
   Send of GLOBAL.GDDR.PARMS.IPL.BTMUL51.DC1            Completed
GDDR373I GDDR Broadcasting PARMS.IPLBCVA.BTMUL51.DC1 = 01111,1111BGM1
GDDR739I GDDR -> Set PARMS.IPLBCVA.BTMUL51.DC1 = 01111,1111BGM1 at DC1
GDDR739I GDDR -> Set PARMS.IPLBCVA.BTMUL51.DC1 = 01111,1111BGM1 at DC3
   Send of GLOBAL.GDDR.PARMS.IPLBCVA.BTMUL51.DC1        Completed
GDDR373I GDDR Broadcasting PARMS.IPLBCVS.BTMUL51.DC1 = 03333,333333M3
GDDR739I GDDR -> Set PARMS.IPLBCVS.BTMUL51.DC1 = 03333,333333M3 at DC1
GDDR739I GDDR -> Set PARMS.IPLBCVS.BTMUL51.DC1 = 03333,333333M3 at DC3
   Send of GLOBAL.GDDR.PARMS.IPLBCVS.BTMUL51.DC1        Completed

  Maximum return code: 4
READY
END
```

## Return codes

Table 44 lists the GDDRISWP utility return codes.

**Table 41** GDDRISWP return codes

| Return code | Description |
|---|---|
| 0 | Parameter updates complete and broadcast successfully to the other C-systems. |
| 4 | No corresponding global variable found (for example, PARMS.IPL exists but PARMS.IPLALT does not exist or has a null value). |
| 8 | The update of the global value failed on the master C-system.<br>-or-<br>The update of the global variable was successful, but the broadcast to one or more other C-systems failed. |
| 12 | The job was not executed on the master C-system.<br>-or-<br>The SYSTEM or SITE parameter value is missing or has an invalid value.<br>-or-<br>No IPL-related global variables were found for the SYSTEM/SITE specified. |

# GDDR IPL Assist Monitor utility (GDDRGIAM)

The GDDR IPL Assist Monitor (GDDRGIAM) utility monitors systems being IPL'd for a hardcoded list of messages and WTORs and ensures that certain error conditions or other prompts do not cause unnecessary delay in system restart processing. This goal is achieved either by performing hardcoded replies to selected WTORs, or by echoing selected messages to create visibility.

**⚠CAUTION**

**Use of GDDRGIAM is not without risk. It is recommended to use this utility in RDR-type situations, only after a thorough study of the functionality. Before enabling GDDRGIAM, review the replies the utility gives to sysplex-related WTORs.**

The GDDR IPL Assist Monitor intercepts a hardcoded list of messages and WTORs occurring on GDDR-managed LPARs.

**Note:** "Intercepted messages" on page 497 and "Intercepted WTORs" on page 497 list the messages and WTORs.

Each intercepted WTOR or message is echoed locally on the system where the GDDR IPL Assist Monitor runs, using message GDDHMC1I. The GDDR message rule for GDDHMC1I is disabled by default. To use the GDDR IPL Assist Monitor, set GDDHMC1I message interception to be enabled automatically when GDDRMAIN starts, using the MSG GDDRPARM parameter.

When the GDDHMC1I message rule is enabled, the message is intercepted by standard GDDR message interception and propagated to the master C-system. The goal of this messaging is to provide visibility on issues during IPL of a managed system, on the C-system where the script runs. GDDR is allowed to interact with HMC and to perform LOAD CLEAR actions in scripts as specified in the default or script submission call overrides.

## Requirements

- The GDDR IPL Assist Monitor must run a on C-system.
- GDDRMAIN must be up and running on all BCPii-capable systems.

## Starting and stopping GDDR IPL Assist Monitor

When the GDDR IPL Assist Monitor is enabled, script GDDRPA27 will start the GDDR IPL Assist Monitor, and script GDDRPA28 will stop it.

**Note:** There could be significant time between the execution of scripts GDDRPA27 and GDDRPA28.

You can also stop the GDDR IPL Assist Monitor at any time, either by command or using *hlq*.GDDR*vrm*.SAMPLIB(GDDRGIAM).

- To determine if the GDDR IPL Assist Monitor is currently active on the local system, issue:

  ```
  F GDDRMAIN,BCPII,STATUS
  ```

- To turn the GDDR IPL Assist Monitor on or off for all LPARs on all CPCs, for all LPARs on a CPC, or for a specific LPAR, issue:

  ```
  F GDDRMAIN,GIAM,ON|OFF,cpc,lpar
  ```

  The action is taken on the system on which the command is issued and has no effect on the GDDR IPL Assist Monitor running on other systems.

- To turn debug on or off for all LPARs on all CPCs, for all LPARs on a CPC, or for a specific LPAR, issue:

  ```
  F GDDRMAIN,BCPII,DBGON|DBGOFF,cpc,lpar
  ```

  The action is taken on the system on which the command is issued and has no effect on the GDDR IPL Assist Monitor running on other systems.

## Sample JCL

The sample JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDRGIAM).

```
//GDDRGIAM JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRGF0S EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRGF0S STOP
/*
```

## Arguments

The syntax is as follows:

```
GDDRRXST GDDRGIAM {STOP|NONE}
```

Where:

STOP

Stops the GDDR IPL Assist Monitor. This is the default value.

NONE

Enables or disables BCPii debugging, based on the DIAGNOSTICS.BCPII.TRACE setting made in the Set Output Message Levels By Program panel (M,D).

## Examples

To stop the GDDR IPL Assist Monitor:

```
GDDRRXST GDDRGIAM STOP
```

## Intercepted messages

GDDRGIAM intercepts the following messages and uses message GDDHMC1I to echo them to the C-system where GDDRGIAM is running:

**GDDRBMSG MSGID=IEA371I**

Explanation: IEA371I *data-set* ON DEVICE *nnnn* SELECTED FOR IPL PARAMETERS

During system initialization, the system selected a dataset containing the IPL parameters. This is an informational message for visibility back to the C-system where the script runs.

**GDDRBMSG MSGID=IEE311I**

Explanation: IEE311I *cm* PARAMETER MISSING

While processing a command, the system detected that a required parameter is missing. This message is included for visibility back to the C-system where the script is running, as it might signal something is wrong with GDDRGIAM, or with the IPL process being monitored.

**GDDRBMSG MSGID=IEE389I**

Explanation: IEE389I MVS COMMAND PROCESSING AVAILABLE

The system issues this message during IPL processing when commands are expected to work successfully. This message is included for visibility back to the C-system where the script is running.

## Intercepted WTORs

GDDRGIAM intercepts the following WTORs and replies as indicated.

**GDDRBMSG MSGID=$HASP420,ACTION=((REPLY,Y))**

Explanation: $HASP420 REPLY 'Y' IF [*memname* IS | ALL MEMBERS ARE] DOWN (IPL REQUIRED), 'N' IF NOT

JES2 has issued either message $HASP419 reporting the members considered dormant, or message $HASP405 if it cannot determine if any other member in the multi-access spool configuration is active. JES2 requires confirmation of the status of each of the displayed members.

Reply **Y** to inform JES2 that the listed JES2 member is dormant. After a reply Y to all $HASP420 WTORs, JES2 will perform a cold start or a warm start, depending on the initialization option you previously specified.

**GDDRBMSG MSGID=IGGN505A,ACTION=((REPLY,CANCEL))**

Explanation: IGGN505A SPECIFY UNIT FOR *dsname* ON *volser* OR CANCEL

During nucleus initialization program (NIP) processing, the system determined that the volume must be mounted to access dataset *dsname*. Select an available device of the type required and respond **R *xx*,*dev***, where *xx* is the reply number and *dev* is the device number of the selected device. You may also respond by signaling EOB (pressing Enter on the console). This action indicates that the volume is not available and is not to be used for this IPL.

**GDDRBMSG MSGID=IXC247D,ACTION=((REPLY,U))**

Explanation: IXC247D REPLY U to ACCEPT USE OR D TO DENY USE OF THE COUPLE DATA SET FOR *typename*.

This system attempted to initialize a couple dataset for the specified type and determined that the dataset might be in use by another sysplex. Message IXC248E, which precedes this message, indicates the name of the dataset that is possibly in use by another sysplex.

Reply **U** to continue initialization of the couple dataset. Allowing initialization to continue on a couple dataset that is in use by another sysplex causes the other sysplex to lose access to the dataset, which might cause the systems in that sysplex to enter a wait state.

**GDDRBMSG MSGID=IXC269D,ACTION=((REPLY,U))**

Explanation: IXC269D REPLY U TO USE RESOLVED DATA SETS, C TO USE COUPLE DATA SETS SPECIFIED IN COUPLE*xx*, OR R TO RESPECIFY COUPLEXX

This system detected an inconsistency in the couple datasets specified in COUPLE*xx*. This system has resolved the inconsistency and has found a consistent primary and alternate couple dataset. However, it does not appear that any of the systems using the datasets are active. The couple datasets specified by COUPLE*xx* have been displayed via message IXC275I as well as those that this system has determined to be consistent.

Reply **U** to continue initialization with the resolved couple datasets that XCF has determined to be the better choice to use.

**GDDRBMSG MSGID=IXC289D,ACTION=((REPLY,U))**

Explanation: IXC289D REPLY U TO USE THE DATA SETS LAST USED FOR *typename* OR C TO USE THE COUPLE DATA SETS SPECIFIED IN COUPLE*xx*

XCF has detected an inconsistency in the couple datasets specified in COUPLE*xx*. The couple datasets specified in COUPLE*xx* are not the same couple datasets last used by the sysplex for this data type. The system lists, via message IXC288I, those datasets last used by the sysplex and also those specified in COUPLE*xx*. This message reply indicates the correct set of couple datasets to be used for further processing.

Reply **U** to continue initialization with the primary and alternate sysplex datasets that were last used by the sysplex, which are not the same couple datasets specified in COUPLE*xx*. This could have been caused by the removal of a primary or an alternate couple dataset, or as the result of the addition of a new alternate couple dataset after the sysplex was IPLed. A normal re-IPL should choose this option.

**GDDRBMSG MSGID=IXC405D,ACTION=((REPLY,I))**

Explanation: IXC405D REPLY I TO INITIALIZE THE SYSPLEX, J TO JOIN SYSPLEX *sysplex-name*, OR R TO REINITIALIZE XCF

This system is trying to initialize or join a sysplex, but XCF found one or more systems already active in sysplex *sysplex-name*. This message prompts the operator to indicate whether the systems displayed in message IXC404I are active in the sysplex and whether initialization should continue. See the explanation of IXC404I for additional information.

Reply **I** to request that sysplex initialization continue because none of the systems identified in message IXC404I are participating in an operating sysplex; that is, they are all residual systems. This system will perform cleanup of old sysplex data, initialize the couple dataset, and start a new sysplex. If any of the systems identified in message IXC404I are currently active in the sysplex, they will be placed into a disabled wait state.

**GDDRBMSG MSGID=IXC501A,ACTION=((REPLY,Y))**

Explanation: IXC501A REPLY Y TO USE COUPLING FACILITY NAMED *cfname* OR N TO NOT USE COUPLING FACILITY

This is the prompt associated with message IXC500I:

IXC500I CONFIRM REQUEST TO USE COUPLING FACILITY *type.mfg.plant.sequence* PARTITION: *partition side* CPCID: *cpcid* NAMED *cfname* AUTHORITY DATA: *plexname mm/dd/yyyy hh:mm:ss*

Message IXC500I is issued when CFRM determines that another sysplex currently owns the named coupling facility.

Verify that the CFRM active policy correctly specifies the coupling facility that is to be used by this sysplex and ensure that the sysplex identified in message IXC500I is stopped from using the coupling facility before responding to this message.

If **Y** is specified, this system issues message IXC559I to ensure that the operator is made aware of the need to prevent another sysplex from using the coupling facility, and then message IXC560A to prompt for confirmation before allowing this system to gain ownership of the coupling facility. Messages IXC500I and IXC559I identify the coupling facility and the sysplex that currently owns it.

---

**Note:** A reply of Y to this prompt and a reply of Y to message IXC560A may cause severe errors if the coupling facility is still being used by the sysplex identified in messages IXC500I and IXC559I.

---

If **N** is specified, the coupling facility will not be used by this system.

**GDDRBMSG MSGID=IXC560A,ACTION=((REPLY,Y))**

Explanation: IXC560A REPLY Y TO CONFIRM THAT COUPLING FACILITY NAME *cfname* SHOULD BE USED BY *plexname1* OR N TO DENY THE USE.

This message appears in conjunction with message IXC559I when the reply to message IXC501A is Y. Message IXC559I is issued to warn the operator that severe errors may occur if the coupling facility is still being used by the sysplex that currently owns it. The operator is asked to ensure that the coupling facility is not being used by the currently owning sysplex and then asked to either confirm or deny the use of the coupling facility.

Verify that the CFRM active policy correctly identifies that the coupling facility should be used by this sysplex and ensure that the sysplex identified in messages IXC500I and IXC559I is stopped from using the coupling facility before responding to this message.

If **Y** is specified, this system will gain ownership of the coupling facility for the sysplex and coupling facility cleanup will occur. Messages IXC500I and IXC559I identify the coupling facility and the sysplex that currently owns it.

---

**Note:** A reply of Y may cause severe errors if the coupling facility is still being used by the sysplex identified in these messages.

---

If **N** is specified, the coupling facility will not be used by this system.

# GDDR Workload Assist Monitor utility (GDDRHCMD)

The GDDR Workload Assist Monitor (WLAM) is a general-purpose facility that can automate actions against GDDR-managed LPARs independently of any software that is running on the target LPAR. WLAM enables you to issue z/OS operator commands and (optionally) wait for a specific SYSLOG message to confirm command completion, subject to a timeout specification.

The GDDR Workload Assist Monitor (WLAM) is called by GDDR scripts through the user exits GDDRUX01 (Start workload) and GDDRUX02 (Stop workload).

## Requirements

- The GDDR Workload Assist Monitor must run a on C-system.
- GDDRMAIN must be up and running on all BCPii-capable systems.

## Restrictions

- The returned message text is limited to 32 lines.
- For multi-line messages that do not use CART (Command And Response Token), only the first message in the series is returned.

## Sample REXX

The sample REXX code for the GDDRUX01 and GDDRUX02 user exits is in *hlq*.GDDR*vrm*.SAMPLIB.

## Arguments

The command specification syntax is as follows:

`CMD(`*`command`*`) [MSG(`*`message`*`)] [WAIT(`*`minutes`*`)]`

Where:

`CMD(`*`command`*`)`

The z/OS command to be issued through the HMC.

`MSG(`*`message`*`)`

The 1-12 character message ID that confirms successful completion of the z/OS command execution.

If the MSG parameter is specified and the return code is 0 or 4, the message text is returned on the REXX External Data Queue (EDQ) and is shown in SYSTSPRT.

`WAIT(`*`minutes`*`)`

The maximum time (in minutes) to wait for the message ID to appear in the SYSLOG.

The default value is 5 minutes (also used when the MSG parameter is specified without the wait parameter).

## Examples

```
CMD(D T) MSG(IEE136I) WAIT(2)
CMD(D IPLINFO) MSG(IEE254I) WAIT(2)
CMD(D A,L) MSG(IEE114I) WAIT(2)
```

## Return codes

Table 44 lists the GDDRHCMD utility return codes.

**Table 42** GDDRHCMD return codes

| Return code | Description |
|---|---|
| 0 | Command issued and message returned successfully. |
| 4 | Command issued, message was truncated due to excessive length. |
| 8 | Command issued, wait time exceeded waiting for the indicated message. |
| 12 | Missing or inconsistent parameters - see messages in SYSTSPRT. |
| 16 | HMC posted a non-zero return code. The HMC RC and error message appear in the REXX External Data Queue and in SYSTSPRT in format:<br>HMC RC=*nnn* - *hmc-error-message-text* |

# GDDR HMC Actions utility (GDDRHMCA)

The GDDR HMC Actions utility (GDDRHMCA) enables GDDR to perform supported HMC actions. Use the GDDR HMC Actions utility to prepare for DR testing by performing LPAR actions before executing a test script.

## Requirements

The GDDRHMCA job must be run on a C-system.

## Sample JCL

The sample JCL can be found in *hlq*.GDDR*vrm*.SAMPLIB (GDDRHMCA).

```
//GDDRHMCA JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRHMC2 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
  EXECUTIL SEARCHDD(YES)
  GDDRRXST GDDRHMC2 +
  ACTION(HMC_ACTION_COMMAND) +
  SYSTEM(SSSSSSSS) +
  SITE(DCX) +
  CPC(NETID.NAU)
  LPAR(LPARNAME) +
  LOADA(LCUU) +
  LOADP(LOADPARM)
/*
```

## Arguments

ACTION(*hmc-action*)

Specifies the HMC action to perform. Supported actions include:

- CBU_ACTIVATE
- CBU_UNDO
- SINGLE_LOAD
- SINGLE_LOAD_BCVS
- SINGLE_LOAD_RECOVERY
- SINGLE_LPAR_ACTIVATE
- SINGLE_LPAR_DEACTIVATE
- SINGLE_LPAR_START
- SINGLE_LPAR_STOP
- SINGLE_RESET_CLEAR

SYSTEM(*ssssssss*)

Specifies the target system name.

SITE(DC*x*)

Specifies the GDDR site ID of the target CPC.

CPC(*cpc-name*)

Specifies the name of the target CPC. The name must be specified in *netid.nau* format.

`LPAR(lpar-name)`

Specifies the name of the target LPAR.

`LOADA(lcuu)`

Specifies the load address (CUU) of the SYSRES volume.

`LOADP(loadparm)`

Specifies the load parameters.

`REALCBU(REAL|TEST)`

Used with ACTION(CBU_ACTIVATE) or ACTION(CBU_UNDO) to determine whether to execute the action specified:

`REAL`

Executes the action.

`TEST`

Validates but does not execute the action.

## Example

```
GDDRRXST GDDRHMC2 +
ACTION(SINGLE_LPAR_STOP) +
SYSTEM(Q31A) +
SITE(DC2) +
CPC(IBM390PS.Q3) +
LPAR(ZOSEQ313) +
LOADA(7000) +
LOADP(7000Q3M1)
```

## Return codes

Table 43 lists the GDDRHMCA utility return codes.

**Table 43** GDDRHMCA return codes

| Return code | Description |
|---|---|
| 0 | Command executed successfully. |
| 12 | ARGUMENT ERROR. An invalid SYSTSIN statement was provided. See message GDDR998E for a description of the error.<br>GDDRHMC returned a non-zero RC. See additional messages in the joblog or the GDDRWORK (GDDWXH*nn*) address space for more information about the problem. |

# GDDR BCPii Connectivity Test utility (GDDRBCPI)

The GDDR BCPii Connectivity Test utility (GDDRBCPI) generates a list of CPC and images that can be connected to from the system on which this job is run. Run this job on all B-systems to ensure that BCPii connectivity is set properly.

**Note:** Run GDDRBPCI after all hardware requirements in the *MVS Programming Callable Services for High Level Languages* are met and all RACF updates creating the specific Facility Classes and setting the required Permits are made.

The utility, from each of the B-systems, verifies connectivity to all CPCs visible through the HMC grouping. Note that this does not mean each of the GDDR B-systems will be able to connect to all CPCs visible via CPC grouping. The purpose of this job is to ensure those CPCs defined to GDDR can be interrogated for information. Further, for each CPC that BCPii can connect to, a list of Images on each is obtained and a BCPii connection attempted to each of those images to ensure all GDDR systems under BCPii management are ready for GDDR.

For systems not eligible for BCPii connectivity from a B-system, error messages are generated.

## Sample JCL

The sample JCL can be found in *hlq*.GDDR*vrm*.SAMPLIB(GDDRBCPI).

```
//GDDRBCPI JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRBCPI EXEC GDDRPROC,PARM='GDDRBCPI'
//*
//
```

## Sample output

As this output shows, the local B-system running on the Q3 CPC has visibility to three other CPCs (C, O, and X1) but has authority to only one other CPC (X1). The error message section at the start is used to diagnose any problems or to highlight issues that might be ignored (for instance, non-authorized CPCs). The BCPii Connection Report shows a list of the CPCs that were visible during processing and also a list of all Images on the CPCs to which the job found authorization. BCPii authority to a CPC is separate from its authority to an image on that CPC (that is, not all images on a CPC may be available for BCPii communication). The JES job log also contains further diagnostic information that Dell EMC support personnel can use to diagnose at a deeper level.

```
BCPii Connection Test beginning...                4 Feb 2011  08:00:56

 These messages may be ignored:

  Error 746: EVALBLOCK too small, retrying for larger
          Default area for data returned from BCPii call is not
          large enough.  A larger area is automatically obtained

  Error 539: BCPII Connection Error - rc 00000F02 rsn FFFFFFFF
          F02 indicates a lack of SAF Authority to the object of
           call to BCPii.  This may require investigation
```

```
  rc XXXXXXXX can be found in the error section in the BCPii section
             of MVS Programming Callable Services for High Level
             Languages for each of the BCPii Call types (CONNECT,
             DISCONNECT, LIST, QUERY, etc).  Each call type has a
             set of specific errors as shown in the above error msg
             showing F02 as a SAF AUTH error during CONNECT attempt.

BCPii Checking CPC/LPAR Connectivity...           4 Feb 2011  08:00:56

BCPii Processing messages follow.  See BCPii Connection Report
below for results of BCPii Connectivity testing

GDDR992E Error 539 in GDDBCPCO: BCPII Connection Error - rc 00000101 rsn FFFFFFFF
GDDR992E Error 539 in GDDBCPCO: BCPII Connection Error - rc 00000101 rsn FFFFFFFF
GDDR992E Error 746 in GDDBCPLS: EVALBLOCK too small, retrying for larger

BCPii Connection Report for 4 identified CPCs  4 Feb 2011  08:01:32

CPC 001 is IBM390PS.C     Connection Not Made - Check RACF Facility Classes
CPC 002 is IBM390PS.O     Connection Not Made - Check RACF Facility Classes
CPC 003 is IBM390PS.Q3    Connection Complete - Will Check Image Connection <-- Local CPC
CPC 004 is IBM390PS.X1    Connection Complete - Will Check Image Connection

BCPii LPAR Connectivity Report                    4 Feb 2011  08:01:32

 CPC IBM390PS.Q3         Image Q31D             Connection Established
 CPC IBM390PS.Q3         Image Q319             Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ31A         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ31B         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ31C         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ31E         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ31F         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ311         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ312         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ313         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ314         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ315         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ316         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ317         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ318         Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ84          Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ85          Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ86          Connection Established
 CPC IBM390PS.Q3         Image ZOSEQ87          Connection Established
 CPC IBM390PS.X1         Image X10A             Connection Established
 CPC IBM390PS.X1         Image X10B             Connection Established
 CPC IBM390PS.X1         Image X10C             Connection Established
 CPC IBM390PS.X1         Image X10D             Connection Established
 CPC IBM390PS.X1         Image X10E             Connection Established
 CPC IBM390PS.X1         Image X10F             Connection Established
 CPC IBM390PS.X1         Image X101             Connection Established
 CPC IBM390PS.X1         Image X102             Connection Established
 CPC IBM390PS.X1         Image X103             Connection Established
 CPC IBM390PS.X1         Image X104             Connection Established
 CPC IBM390PS.X1         Image X105             Connection Established
 CPC IBM390PS.X1         Image X106             Connection Established
 CPC IBM390PS.X1         Image X107             Connection Established
 CPC IBM390PS.X1         Image X108             Connection Established
 CPC IBM390PS.X1         Image X109             Connection Established
 CPC IBM390PS.X1         Image X11A             Connection Established
 CPC IBM390PS.X1         Image X11B             Connection Established
 CPC IBM390PS.X1         Image X11C             Connection Established
 CPC IBM390PS.X1         Image X11D             Connection Established
 CPC IBM390PS.X1         Image X11E             Connection Established
 CPC IBM390PS.X1         Image X11F             Connection Established
```

```
 CPC IBM390PS.X1         Image X111             Connection Established
 CPC IBM390PS.X1         Image X112             Connection Established
 CPC IBM390PS.X1         Image X113             Connection Established
 CPC IBM390PS.X1         Image X118             Connection Established
 CPC IBM390PS.X1         Image X119             Connection Established

BCPii Connection Report Completed

READY
END
```

# GDDR Load Profile Management utility (GDDRLPRF)

The GDDR Load Profile Management utility (GDDRLPRF) is a batch interface between GDDR and the HMC-stored load activation profiles, applicable to systems running at the current primary DASD site.

The utility provides two options:

- Update the HMC load activation profile of a managed system based on GDDR IPL values stored for that system
- Validate the GDDR global variable-stored IPL parameters for a managed system against the managed system information and the information stored in the associated LOAD profile member

## Requirements

- The utility must run on the C-system of any site.
- GDDRMAIN must be up on the C-system from which the utility is run.
- GDDRMAIN must be up on any system from which the information is requested.

## Sample JCL

The sample JCL can be found in *hlq*.GDDR*vrm*.SAMPLIB(GDDRLPRF).

```
//GDDRLPRF JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRLPRF EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRLPRF VALIDATE
/*
```

## Arguments

The syntax is as follows:

```
GDDRRXST GDDRLPRF VALIDATE|UPDATE
```

Where:

VALIDATE

Examines the IPL parameters for the managed systems running at the current primary DASD site. For each system that specifies a LOAD profile stored on the HMC, the utility verifies the IPL load address and load parameters stored within GDDR against those stored in the load profile and, as an additional check, verifies the current load address and load parameters for the managed system(s) in question.

If there is no stored load profile and GDDR cannot communicate with the managed system, no validation is performed.

UPDATE

Updates the HMC load activation profile specified for a managed system running on the current primary DASD site with the GDDR IPL values (residence volume address and the IPL Load parameter) stored for that system, if a GDDR global variable containing the load activation profile name is present. If no global variable, no update takes place.

## Example

```
GDDRRXST GDDRLPRF VALIDATE
```

# GDDR ECGUTIL Driver utility (GDDRECG0)

Use the GDDRRECG0 utility to change device status.

The GDDRRECG0 utility only affects GDDR-managed devices.

## Requirements

Run GDDRRECG0 at the site with channel access to the DASD at the selected site.

## Sample JCL

The sample JCL can be found in *hlq*.GDDR*vrm*.SAMPLIB(GDDRECG0).

```
//GDDRECG0 JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDRECG0 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDRECG0 arguments
/*
```

## Arguments

The syntax is as follows:

DC*x function* ALL | FBA

Where:

ALL | FBA

Runs against all devices or FBA devices. This argument is optional.

DC*x*

Determines the site where the devices reside (DC1, DC2, DC3, or DC4).

*function*

One of the following values:

- UNR—Makes devices USRNRDY.
- RDY—Makes devices USRRDY.
- CLR—Clears the CG and ECA device flags.

## Examples

- To clear the CG and ECA device flags at DC1:

  ```
  GDDRRXST GDDRECG0 DC1 CLR
  ```

- To make FBA devices at DC3 USRNRDY:

  ```
  GDDRRXST GDDRECG0 DC3 UNR FBA
  ```

# GDDR Expected Events utility (GDDREE00)

Run GDDREE00 to either view the list of expected events or to clear any outstanding expected events. This might be useful in cases when a GDDR script has not completed but set some expected events on.

**Note:** "Expected events" on page 69 discusses expected events.

## Requirements

Run GDDREE00 on the master C-system.

## Sample JCL

The sample GDDREE00 JCL is in *hlq*.GDDR*vrm*.SAMPLIB(GDDREE00).

```
//GDDREE00 JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//GDDRPROC JCLLIB ORDER=(...)  <--- Lib containing customized GDDRPROC
//*
//GDDREE00 EXEC GDDRPROC
//GDDR.SYSTSIN DD *
 EXECUTIL SEARCHDD(YES)
 GDDRRXST GDDREE00 arguments
/*
```

## Arguments

The syntax is as follows:

`GDDRRXST GDDREE00 A(action) [E(events)]`

Where:

`A(action)`

This required parameter specifies one of the following actions:

`CLEAN`

Clears expected events.

`PRINT`

Displays a list of expected events.

`REMOVE`

Removes specific events from the list of expected events.

The REMOVE action requires you to specify the event(s) to be removed using the E(*events*) parameter.

`E(events)`

Lists one or more event names to be removed. This parameter is required when A(RMOVE) is specified.

**Note:** Table 3 on page 70 lists possible expected events.

You can separate the entries with a blank space or a comma, for example:

```
E(SRA,MSC)
E(SRA MSC)
```

## Examples

- To clear expected events:

  ```
  GDDRRXST GDDREE00 A(CLEAN)
  ```

- To view expected events:

  ```
  GDDRRXST GDDREE00 A(PRINT)
  ```

- To remove 'SRA' and 'MSC' events from the list of expected events:

  ```
  GDDRRXST GDDREE00 A(REMOVE) E(SRA MSC)
  ```

## Return codes

Table 44 lists the GDDREE00 utility return codes.

**Table 44** GDDREE00 return codes

| Return code | Description |
|---|---|
| 0 | Successful operation. |
| 4 | Warning: request to display/clean an empty list of expected events or to remove an event which is not an expected event. |
| 8 | Argument error: no or wrong action specified or required parameter missing. |
| 16 | Severe error: the GDDREE00 utility invoked not on the master C-system. |

# GDDR DIV Management utility (GDDRGVX)

The GDDRGVX utility provides global variable analysis and management functions for the DIV dataset.

## Sample JCL

The GDDRGVX utility is available in *hlq*.GDDR*vrm*.SAMPLIB(GDDRGVX).

```
//GDDRGVX  JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//GDDRGVX  EXEC PGM=GDDRGVX,PARM='DSPLIST'
//STEPLIB  DD  DISP=SHR,DSN=DS-PREFIX.LINKLIB  <--- Your GDDR LINKLIB
//GDDRGVDS DD  DISP=SHR,DSN=DS-PREFIX.CSYS.DIV <--- Your DIV dataset
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//GDD$GDDR DD  DUMMY  <--- Your GDDR subsystem name ('GDDR' is default)
//*
//
```

Modify STEPLIB to specify the GDDR LINKLIB used by your GDDR instance, and modify GDDRGVDS to specify your global variable DIV dataset name. Also, modify GDDR subsystem name below to match that of your GDDR instance.

## Parameters

`DSPLIST`

The DSPLIST parameter produces a formatted list of the contents of the global variable data space.

DSPLIST can be used to show information about locks. It shows the time when the exclusive lock was set, as well as the job which set it. It does not show the jobs which set the shared lock, but DSPLIST indicates the last time the shared lock was changed.

`DIVLIST`

The DIVLIST parameter produces a formatted list of the contents of the DIV.

This parameter may only be used when GDDRMAIN is not active. The output format is identical to DSPLIST. The GDDRGVX utility JCL requires a GDDRGVDS DD pointing to the DIV dataset.

`DSPSAVE`

The DSPSAVE parameter copies the contents of the global variable data space to a sequential dataset.

The dataset is pointed to by a GVSAVE DD statement. The output dataset can be used by IDCAMS REPRO to populate a new DIV dataset.

`RELOAD`

The RELOAD parameter copies global variables from an old DIV to a new DIV.

The reason to use RELOAD instead of IDCAMS REPRO is that RELOAD does not copy deleted variables. Also, any “gas” in the DIV will be removed.

To use the RELOAD parameter, GDDRMAIN must be running.

The old DIV is pointed to by a GDDRGVDS DD statement in the GDDRGVX utility JCL, while the GDDRGVDS DD statement in GDDRMAIN must be pointing to another (empty) DIV.

# GDDR Command Queue utility (GDDRXCMD)

The GDDRXCMD utility allows you to print or clear the GDDR command queue.

Customize the following examples according to your site's requirements.

## Print current queue

To print the current queue:

```
//QPRINT EXEC PGM=GDDRXCMD,PARM=PRINT
//SYSPRINT DD SYSOUT=*
```

The result will be a hexadecimal + character printout of the current records in the queue.

## Clear current queue

To clear the current queue:

```
//QCLEAR EXEC PGM=GDDRXCMD,PARM=BOTH
//SYSPRINT DD SYSOUT=*
```

The job must end with RC=0. Run another print job to verify that the queue has indeed been cleared.

IMPORTANT

Only clear the queue if you have been advised to do so by Dell EMC Customer Support.

# GDDRMAIN Trace Print utility (GDDRTRCP)

Run the GDDRTRCP utility to troubleshoot suspected GDDR software issues as instructed by GDDR Solution Support.

## Sample JCL

```
//GDDRTRCP JOB (EMC),'GDDR',SYSAFF=*,NOTIFY=&SYSUID,
//             CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//GDDRTRCP EXEC PGM=GDDRTRCP
//STEPLIB  DD  DISP=SHR,DSN=DS-PREFIX.LINKLIB  <--- Your GDDR LINKLIB
//SYSPRINT DD  SYSOUT=*
//SYSABEND DD  SYSOUT=*
//GDD$GDDR DD  DUMMY  <--- Your GDDR subsystem name ('GDDR' is default)
//*
```

Modify STEPLIB to specify the GDDR LINKLIB used by your GDDR instance. Also, modify the GDDR subsystem name below to match that of your GDDR instance.

## Optional EXEC parameters

`DEBUG`

Dump entire trace buffer.

`END=`*`YYYYMMDDHHMMSS`*

Filter by end time (can be truncated to a lower resolution).

`JOB=`*`jobname`*

Filter by job name (supports wildcards).

`JOBID=`*`job_id`*

Filter by job ID (supports wildecards).

`MODULE=`*`module`*

Filter by module (supports wildcards).

`START=`*`YYYYMMDDHHMMSS`*

Filter by start time (can be truncated to a lower resolution).

# CHAPTER 8
# Running GDDR Scripts

This chapter includes the following topics:


- Overview ..... 518
- Running scripts via GDDR ISPF interface ..... 524
- Planned scripts ..... 530
- Test scripts ..... 538
- Unplanned scripts ..... 542
- Resumption scripts ..... 547
- RDR scripts ..... 552
- Special scripts ..... 557

# Overview

## What is a GDDR script

A GDDR script is a predetermined sequence of function calls. Generally one function call corresponds to one action type performed on all applicable target objects. A GDDR script is started by calling GDDR-provided routines, either from a batch job or as a result of specific messages being issued.

GDDR scripts can only run on the master C-system. A single GDDR-plex can have one in-progress script at any time.

Scripts can be initiated as follows:

- From the GDDR ISPF interface, as described in "Run GDDR scripts (S)" on page 375
- Using the GDDRMAIN,SCRIPT command described in "SCRIPT" on page 225

GDDR provides scripts to perform actions of the following types:

- Planned event management

  Scripts to verify your configuration, perform site swaps, switchover and switchback.

- Testing management

  Scripts to test your configuration with a snapshot copy.

- Unplanned event management

  The messages or events that trigger an unplanned or takeover process can originate on any system, either a C-system or a production system. They only take place on the current master C-system.

  They are invoked automatically after any of the following types of failure or loss are detected:

  - Sites
  - DASD
  - Systems
  - Loss of SRDF link
  - Loss of host channels

- Resumption of replication after SRDF link outages
- Regional Disaster Recovery (RDR) operations
- Special actions

## Site and region designations

Regions participating in the SRDF/SQAR configuration are labeled primary (RG1) or secondary (RG2), depending on the location of the primary site workload. When the region is shown as <n> in script titles, this indicates the role of the region at the start of the script.

Sites participating in the SRDF/SQAR configuration are labeled Site A or primary (*DC1*), Site B or secondary (*DC2*), Site C or tertiary (*DC3*), and Site D or quaternary (*DC4*). When in script titles, this indicates the role of the site at the start of the script. *Italics* means that the values are interchangeable.

## Scripts by category

GDDR allows you to run the following types of scripts:

- Planned scripts

  Operations personnel can handle planned event management scenarios by running the following scripts:

  - Automated Configuration Check - DASD (GDDRPCCD)
  - Abandon Secondary/Tertiary/Quaternary Site (DC2/DC3/DC4) (GDDRPA60)
  - Abandon Region RG1 (site swap) (GDD2P17A)
  - Restart production at DC3/DC4 after site swap (GDD2P18A)
  - Swap production from DC1 to DC2 (GDDRPA42)
  - Swap DASD from DC1 to DC2 (GDDRPA21)

- Test scripts
  - Perform test IPL from BCVs at DC2/DC3/DC4 (GDD2P01A)
  - Resume after test IPL from BCVs at DC2/DC3/DC4 (GDD2P02A)
  - Perform test IPL from R2s at DC2 (GDD2P03A)
  - Perform test IPL from R2s at DC3/DC4 (GDDRPA27)
  - Resume after test IPL at DC3/DC4 (GDDRPA28)
  - Resume after test IPL from R2s at DC2 (GDD2P16A)

- Unplanned scripts

  Operations personnel can manage unplanned events as follows:

  The GDDR Event Monitor prompts the operator for management confirmation of trigger events which indicate a site or DASD outage. The operator replies affirmative to the prompt and the GDDR recovery script starts.

  The operator may start the appropriate unplanned script and respond to prompts. The script initiates and validates that the state of the current host and storage environments matches the script prerequisites before proceeding.

  - Recover after unplanned swap (GDDRUP31)
  - Recover after loss of DC1 (LDR) (GDD2U13A)

- Resume replication after unplanned swap (GDDRPA0A)
- Resume replication after loss of DC1 (GDDRPA0A)

- Resumption scripts

  Operations personnel can resume operations after planned or unplanned outages by running any of the following scripts:

  - Resume SRDF/S replication after ConGroup trip (GDDRPA23)
  - Resume SRDF/A in MSC mode to DC3/DC4 (GDDRPM29)
  - Resume SRDF/A in SQAR mode (GDDRPF29)
  - Reclaim Secondary/Tertiary/Quaternary site (DC2/DC3/DC4) (GDDRPA65)

- RDR scripts

  Operations personnel can initiate secondary region operations by running any of the following scripts:

  - Recover at DC3/DC4 after RDR in primary region/Recover at DC3/DC4 with SRDF/S to DC4/DC3 (GDDRPA05)
  - Restart production at DC3/DC4 with cascaded/concurrent SRDF (GDDRPA06)
  - Recover at DC3 with cascaded SRDF DC3->DC4, DC4->DC2/Recover at DC4 with concurrent SRDF DC4->DC3, DC4->DC2 (GDDRPA07)

- Special scripts

  - Planned STAR-HA Takeover (GDDRPA36)
  - Unplanned STAR-HA Takeover (GDDRUP35)
  - Transfer AutoSwap Owner to DCn (GDDRPXAS)
  - Restart primary MSC server at DCn (GDDRMMSC)
  - Restart secondary MSC server at DCn (GDDRPA36)
  - Transfer Master C System to DCn (GDDRPXMC)
  - Global variable backup (GDDRPGVB)
  - Move systems to alternate CPC (GDDRMCPC)

## Pre-script environment checks

Before you initiate any planned swap scenarios, it is recommended to complete the following checks:

- Verify GDDR-managed storage matches GDDR configuration definition
- Perform GDDR health check
- Verify SRDF/A is active and cycling
- Verify ConGroup is active and enabled
- Validate active AutoSwap group
- Verify SDDF sessions

## Verify GDDR-managed storage matches GDDR configuration definition

Run the GDDRPCCD script described in "Automated Configuration Check - DASD (GDDRPCCD)" on page 530. The script is listed in the Planned script section of the Select Script to Run panel (S).

Review the GDDRPCCD script joblog for GDDP400E and GDDP420E error messages. Resolve discrepancies by revising the RDF.DEVICES, DLM.DEVICES and STDBCV parameters before proceeding.

## Perform GDDR health check

Use the Perform Health Check panel (C) to verify your configuration. Verify that there are no active events, and no Degraded mode.

## Verify SRDF/A is active and cycling

The GDDR Event Monitor automatically verifies that SRDF/A is running, and sets SRA and MSC events based on its findings. This monitoring is done on a cyclic basis, under control of the EVM SRDF/A Check Interval option in the Specify GDDR Tuning Values panel (M,P,O,T).

To verify manually, enter the following SRDF Host Component command from the GDDR master C-system:

`#SQ SRDFA,`*`cuu`*

Where:

`#`

Represents the SRDF Host Component command prefix.

*`cuu`*

Represents the gatekeeper UCB address for any storage system at the primary DASD site.

---

**Note:** The #SQ SRDFA command displays data about the entire SRDF/A configuration in the storage system. Use #SQ SRDFA,LCL or #SQ SRDFA,RMT syntax to display data about a single SRDF group. The *SRDF Host Component for z/OS Product Guide* describes syntax and usage of the #SQ SRDFA command.

---

To ensure that SRDF/A is active and cycling, the following #SQ SRDFA output fields should all be set to Y:

- `SECONDARY CONSISTENT`
- `MSC ACTIVE`
- `GLOBAL CONSISTENCY`
- `STAR RECOVERY AVAILABLE`

The TIME SINCE LAST CYCLE SWITCH value should generally be around the number of seconds specified in the SRDF Host Component initialization parameter, MSC_CYCLE_TARGET. A large number would indicate that SRDF/A has stopped cycling. Subsequent displays should also show the CAPTURE TAG and TRANSMIT TAG values incrementing.

Take the following actions if SRDF/A is not active and cycling:

- Report the problem to your z/OS systems programming support group.
- Do *not* start the swap.

## Verify ConGroup is active and enabled

Enter the following command from the GDDR master C-system:

```
F emccgrp,D C
```

Where:

*emccgrp*

Is the name of the Consistency Groups for z/OS started task.

**Note:** The *Consistency Groups for z/OS Product Guide* describes syntax, usage, and output of the DISPLAY CONGROUP command.

The display will show the consistency group as Enabled and Active and each device in the consistency group *must* show as ARMeD and Enabled.

There is a problem if the group is not in the expected state or one or more devices are not in the expected state.

Take the following actions if ConGroup is not active and enabled:

- Report the problem to your z/OS systems programming support group.
- Do *not* start the swap.

## Validate active AutoSwap group

Enter the following command from the GDDR master C-system:

```
F emccgrp,DAS VAL GRP *
```

**Note:** There is a space between the GRP and the *.

Where:

*emccgrp*

Is the name of the ConGroup started task.

**Note:** The *AutoSwap for z/OS Product Guide* describes syntax, usage, and output of the VALIDATE GROUP command.

Verify that:

- Your Lost Owner Policy is set to OPERATOR.
- The AutoSwap Owner is at the secondary DASD site.

To verify that the AutoSwap group or swap group is valid, the display must show *all* of the following:

- The Total Devices and Valid counts *must* be equal.

- The Invalid count *must* be zero.
- All systems participating in the swap *must* be listed and report devices valid.

**Note:** The C-system at DC3 (site C) will always report AutoSwap as not active. AutoSwap never runs at DC3 (site C).

Take the following actions if validation of the AutoSwap group fails:

- Report the problem to the z/OS systems programming support group.
- Do *not* start the swap.

### Verify SDDF sessions

Run the GDDRSDDF utility described in "GDDR SDDF Session Verification utility (GDDRSDDF)" on page 469 to verify SDDF sessions on the recovery leg of the GDDR configuration.

Run the GDDRSDDF utility for a minimum of 2 intervals, with an interval time that is appropriate for your configuration.

## Rerunning scripts

The return codes from the function calls that make up a GDDR script are saved in GDDR global variables. For functions that issue SRDF Host Component commands, the return code of the commands is also saved. If multiple commands are issued from one function, the return codes from each command are saved in GDDR global variables.

After the cause of the original failure has been identified and resolved, the GDDR script can be rerun. GDDR uses the saved return codes to establish the point of restart; that is, the point of the previous failure. This ensures that no modifications to the supplied GDDR script jobs are required in order to rerun after a failure.

# Running scripts via GDDR ISPF interface

IMPORTANT

Do not run GDDR scripts during DLm configuration changes.

Take the following steps to run a script:

1. Navigate to the Select Script to Run panel (S) described in "Run GDDR scripts (S)" on page 375.
2. Select the script by typing **S** next to the script and press **Enter**.

   **Note:** If another script is currently in progress, a message similar to the following is displayed on the Select Script to Run panel (S):
   `Script GDDRPA42 is in progress. Only the script in progress may be submitted at this time.`
3. If no script is in progress, the Specify Parameters for Initial Script Run panel is displayed:

```
--------------- GDDR - Specify Parameters For Initial Script Run ---------------
 Command ===>

   Selected script:  Swap production from DC2 to DC1

   Enter or change the job statement below:

      //*
      //*
      //*
      //*

   Specify call override changes for this script?    ===> N   Enter Y or N

   Press <Enter> when ready to submit script job
   Press <F3> to return to menu without submitting job
```

**Figure 130** Specify Parameters for Initial Script Run panel

4. Confirm or clear any of the call overrides that apply to this script by entering **Y** in "Specify call override changes for this script?"

The Specify Call Overrides for Script panel is displayed, similar to the following:.

```
----------------- GDD3 - Specify Call Overrides for Script ---------------------

 Script:  Global Variable Backup..................... Current Master: LB01M3A
 For each row, you may change the call indicator:       Primary Site:   DC1
   Enter Y to call the function                         Primary DASD:   DC1
   Enter N to not call the function                     Automation:     ON
 When ready, press <F3> to continue the script submit process
 Enter CANCEL and press <Enter> to terminate the script submit process

 Call?  Program   Function
 -----  --------  ------------------------------------------------
   N    GDDRXDRV  Manage Distributed Workload
   Y    GDDRRDF0  Call DYNAPI Interface
   Y    GDDRRDF0  DYNAPI - SRDF/S Commands
   Y    GDDRRDF0  DYNAPI - DeletePair and Half-DeletePair Commands
   Y    GDDRRDF0  DYNAPI - CreatePair and ResumePair Commands
   Y    GDDRRDF0  DYNAPI - SRDF/A Commands
   Y    GDDRRDF0  DYNAPI - Swap and Half-Swap Commands
   N    GDDRGF08  Use ConGroup Shutdown/Startup instead of Refresh
   N    GDDRKF0C  Trigger Production System Shutdown
   N    GDDRKF0I  Trigger Production System Startup
   N    GDDRKF20  (Scan Mode) - Manage BCVs
   Y    GDDRGFFM  Transfer Master Function Ownership
   Y    GDDRKF20  (DC1) - Manage BCVs
...
```

**Figure 131** Specify Call Overrides for Script panel

The functions listed in the Specify Call Overrides for Script panel vary based on the current configuration and script selected.

The Specify Call Overrides for Script panel provides help information (accessed by pressing **PF1**) that describes the function of each program. Use **PF3** to return to the Specify Call Overrides panel.

Table 5, “GDDR call overrides,” on page 85 lists the possible call overrides and their meanings. All call overrides do not apply to all scripts. For any script you run, you normally see a subset of the possible call overrides.

You can overtype the default “Y” or “N” call override values for this script.

You can specify call overrides only once for each execution of a script. If a script terminates before completion, you can rerun the script (as discussed in “Recovering from script errors” on page 527), but any changes you make to the call overrides will be ignored. However, when you execute the same script again after it successfully completes, you can specify call overrides for it.

5. When you are finished, press **F3** to return to the Specify Parameters for Initial Script Run panel. Then press **Enter** to continue.

**Result:** A pop-up similar to the following is displayed and you are prompted *twice* to confirm the job submission.

```
+----------- Please Confirm Job Submission -----------+
| Command ===>                                        |
|                                                     |
| Swap DASD from DC2 to DC1                           |
|                                                     |
| Please confirm submit of this script                |
|                                                     |
| Submit script? ===> N Reply Y or N                  |
|                                                     |
+-----------------------------------------------------+
```

**Figure 132** Job submission confirmation panel

6. At each prompt, reply **Yes** to confirm or **No** to terminate job submission.

   **Result:** After GDDR receives both confirmations, it submits the job to start the selected planned script. On starting, you are prompted (through WTOR) to confirm that you want to run the script.

7. Reply **Yes** to allow the script to continue or **No** to terminate the script.

## Selecting managed systems for test scripts

If GDDR has been configured to manage P-systems, you may be prompted to select managed systems for the script to run against after you select a test script in the Scripts for TEST Action section of the Select Script to Run panel (S).

- For test scripts that perform a test IPL, the Select System for DC*n* Restart panel is displayed similar to the following:

```
---------------------- GDDR - Select System For DC2 Restart --- Row 1 to 2 of 2
Command ===>
Enter S next to system(s) to be restarted or           Current Master: GDDRDEVL
 *ALL* (all systems)                                     Primary Site: DC1
Press <F3> to return to script selection                 Primary DASD: DC1
      System     Usual   DC2 Load   DC2 Load     DC2 Target
Sel   Name       Site    Address    Parameter    LPAR
---   --------   ----    --------   ---------    -----------
_     *ALL*      DC2
_     CLAYB034   DC2     1111       111101M1     IBM390PS.M2964,M34
****************************** Bottom of data ********************************
```

**Figure 133** Select System for DC*n* Restart panel

  - In the Select System for DC*n* Restart panel, type **S** for the system you want to include in the test IPL script and press **Enter** to proceed. You can select one, multiple, or *ALL* systems.

- For test scripts that resume operations after a test IPL, the Select System for DC*n* Reset panel is displayed similar to the following:

```
---------------------- GDDR - Select System For DC2 Reset --- Row 1 to 3 of 3
Command ===>
Enter S next to system(s) to be reset     or           Current Master: GDDRDEVL
 *ALL* (all systems) or *IPLed* (previously IPLed)       Primary Site: DC1
Press <F3> to return to script selection                 Primary DASD: DC1
      System     Usual   DC2 Load   DC2 Load        DC2 Target
Sel   Name       Site    Address    Parameter       LPAR
---   --------   ----    --------   ---------       -----------
 _    *ALL*      DC2
 _    *IPLed*    DC2
 _    CLAYB034   DC2                                IBM390PS.M2964,M34
****************************** Bottom of data ******************************
```

**Figure 134** Select System for DC*n* Reset panel

  - In the Select System for DC*n* Reset panel, type **S** for the system you want to include in the script and press **Enter** to proceed. You can select one, multiple, or *ALL* systems.

  *IPLed* indicates that the script is to shut down the same systems as those that were selected for the test IPL script.

## Recovering from script errors

The Resiliency Expert utility automates recovery from GDDR-detected script errors. Prior to GDDR V4.2, if an SRDF Host Component command failed during a GDDR script, the script stopped and manual intervention was required. The Resiliency Expert attempts to correct devices which have failed to successfully complete an SRDF Host Component command.

The Resiliency Expert is only called from module GDDRRDF0 (this module controls GDDR usage of SC VOL commands during GDDR scripts). It is controlled via global variables which permit GDDR to either:

- Identify the failing commands and devices without attempting fixes.
- Identify them and attempt correction.

Table 45 lists the global variables that determine the operating mode:

**Table 45** Resiliency Expert global variables

| Variable | Description |
|---|---|
| GLOBAL.GDDR.PARMS.RDF0.FIX | If the value is 0, the Resiliency Expert is disabled.<br>If the value is greater than 0, the Resiliency Expert is enabled.<br>The default value is 1. |
| GLOBAL.GDDR.PARMS.RDF0.FIX.LIMIT | Sets the maximum number of iterations allowed to fix a device.<br>If the value is 0, fix attempts are blocked, but failing devices are identified.<br>If the value is greater than 0, fixes are attempted, but are limited by the value of the global. For example, if GLOBAL.GDDR.PARMS.RDF0.FIX.LIMIT = 10, the fix process terminates after 10 consecutive failures to fix devices (10 separate devices, not 10 attempts at the same device). The failures must be consecutive; a success resets the counter to 0.<br>The default value is 10. |

**Table 45** Resiliency Expert global variables

| Variable | Description |
|---|---|
| GLOBAL.GDDR.PARMS.RDF0.FIX.LIMIT.SYMM | Sets the maximum number of iterations allowed to fix devices on a single storage system. This limit applies to each storage system separately.<br>If the value is 0, fix attempts are blocked, but failing devices are identified.<br>If the value is greater than 0, fixes are attempted, but are limited by the value of the global. The failures must be consecutive; a success resets the counter to 0.<br>The default value is 10. |
| GLOBAL.GDDR.PARMS.RDF0.FIX.LIMIT.RDF | Sets the maximum number of iterations allowed to fix devices in an SRDF group. This limit applies to each group separately.<br>If the value is 0, fix attempts are blocked, but failing devices are identified.<br>If the value is greater than 0, fixes are attempted, but are limited by the value of the global. The failures must be consecutive; a success resets the counter to 0.<br>The default value is 10. |

You can enable or disable the Resiliency Expert via the FIXPERT command as follows:

| | |
|---|---|
| `F GDDRMAIN,FIXPERT,ON` | Toggles GLOBAL.GDDR.PARMS.RDF0.FIX ON |
| `F GDDRMAIN,FIXPERT,OFF` | Toggles GLOBAL.GDDR.PARMS.RDF0.FIX OFF |
| `F GDDRMAIN,FIXPERT` | Displays the current state of the Resiliency Expert |

The Resiliency Expert supports the following SRDF Host Component commands:

| | | |
|---|---|---|
| ADCOPY | HSWAP | RDF-SUSP |
| ADCOPY-DISK | ITA | RDF-WR-ENABLE |
| CASRSUM | MOVEPAIR | RDY |
| CASSUSP | NADCOPY | R/O |
| CASSWAP | NDOMINO | R/W |
| CREATEPAIR | NITA | SWAP |
| DELETEPAIR | NRDY | USR_NRDY |
| DOMINO | RDF-NRDY | USR_RDY |
| HDELETEPAIR | RDF-RDY | |
| HMOVEPAIR | RDF-RSUM | |

The Resiliency Expert handles only one device at a time. If it does not have a specific action to take for the failing command, it reissues the failing command. Commands are typically issued for all devices or a range of devices, so if the reissue is effective in fixing the problem, it may fix it for a number of devices. The original command is only reissued one time.

## Rerunning a script

If you are not running the Resiliency Expert or any step of a script completes with a non-zero return code, the script terminates. The failure reason must be investigated and resolved. You can find a description of the GDDR639I message you receive and the return codes that accompany it in the *GDDR Message Guide*.

After the issue has been resolved, submit the script again. No JCL changes are required. GDDR determines the correct step from which to resume script processing.

Depending on how the issue was resolved, it may be necessary to skip the failing step. The GDDR Solution Support team will assist you in making this decision and taking appropriate action.

## WTOR messages

During execution of a script, GDDR displays WTOR messages for each of the steps required to execute a site swap.

No operator reply is required for GDDB-prefixed WTOR messages, as these will be replied to automatically by GDDR message interception rules or by GDDR internal logic.

An operator reply is always required for GDDO-prefixed WTOR messages.

An operator reply is optional for GDDR-prefixed WTOR messages. GDDR will automatically DOM these messages when the condition it is waiting for is reached. To stop the GDDR wait, the operator can reply causing the script to stop, or stop waiting and continue.

# Planned scripts

Choose the planned script you want to run from the Scripts for PLANNED Actions list in the Select Script to Run panel (S).

**Note:** See "Site and region designations" on page 519 for information about site designations.

## Automated Configuration Check - DASD (GDDRPCCD)

Use this script as part of the pre-script checkup before any GDDR script is run.

The GDDRPCCD script runs the GDDRACDD utility in validation mode. This utility then validates that the GDDR-managed storage matches the SRDF device population defined storage configuration at any time.

If the validation fails, the GDDRPCCD script raises the CFG event and blocks all scripts running against the storage concerned.

Review the GDDRPCCD script joblog for GDDP400E and GDDP420E error messages.

Resolve discrepancies by revising the RDF.DEVICES parameters before proceeding.

See "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 for more information about validations performed by GDDRACDD.

Note the following:

- The GDDRPCCD script should be run before any script from the Planned, Test, or Resumption category is run.
- If the GDDRPCCD script discovers any discrepancies between the discovered storage and the defined storage it will raise the CFG event. As long as this event is true, no scripts touching the storage configuration are allowed. Investigate the source of the discrepancy. Either correct the storage configuration, or start a GDDR Parameter Wizard session and run GDDRACDD in Discovery mode to update the defined storage configuration.

### Steps

This script performs the following actions:

- Discovers PowerMax/VMAX devices in a set of defined storage systems and SRDF groups as well as TimeFinder devices, and validates that discovered devices are defined to GDDR in RDF.DEVICES, DLM.DEVICES and STDBCV parameters.
- Validates existing RDF.DEVICES and DLM.DEVICES parameters, as well as PARMS.STDBCV parameters and other configuration global variables against the discovered DASD configuration and against GDDRPARM information.
- If discrepancies are found between the discovered configuration and the defined configuration, the CFG event is set to true. This prevents all scripts from running, except the GDDRPCCD and GDDRPGVB scripts. To be able to run other scripts, either the DASD configuration must be changed to match the defined configuration, or new parameters must be activated using the GDDR Parameter Wizard to match the existing DASD configuration.

# Abandon Secondary/Tertiary/Quaternary Site (*DC2/DC3/DC4*) (GDDRPA60)

The GDDRPA60 script allows operations personnel to take down any site in the configuration for maintenance purposes. This script is used to shut down the single site workload at the designated site in preparation for site maintenance.

## Steps

The script performs the following actions:

- To abandon the secondary site:
  - Verify SQAR mode
  - Check for invalid tracks in SRDF/S
  - Perform a ConGroup trip
  - Verify status after the ConGroup trip
  - Realign coupled datasets
  - Rebuild CF structures at DC1
  - Run the SRDF RDF-SUSP command against external devices replicating from DC2 to DC4
  - Stop application workloads at DC2
  - Reset LPARs at DC2
- To abandon the tertiary site:
  - Verify SQAR mode
  - Check for invalid tracks in SRDF/S
  - Run the MSC,PENDDROP command to stop SRDF/A replication from DC1 to DC3
  - Run ADCOPY from DC1 to DC3
  - Run the SRDF RDF-SUSP command against external devices replicating from DC1 to DC3
- To abandon the quaternary site:
  - Verify SQAR mode
  - Run the MSC,PENDDROP command to stop SRDF/A replication from DC2 to DC4
  - Run the SRDF RDF-SUSP command against external devices replicating from DC2 to DC4

## Restrictions

Before starting, the script verifies that the script is running from the master C-system at the primary site. If this check fails, the script terminates with a GDDR926E script generation error. For details, see the *GDDR Message Guide*.

For configurations with AutoSwap, note the following considerations when using GDDRPA60 to abandon the secondary DASD site:

- The master C-system and the AutoSwap owner are co-located on the secondary DASD site. If the configuration has MSC, then the primary MSC server is also located on the secondary DASD site.
- For the Abandon Secondary Site script GDDRPA60, you must take the required actions in advance to move all these roles to the primary DASD site.
- Use the GDDR scripts (GDDRPXMC, GDDRPXAS, GDDRMMSC, GDDRPA36) to perform these moves.

## Abandon Region *RG1* (site swap) (GDD2P17A)

This script is used to shut down the single site workload at the primary region in preparation for the restart of processing at the secondary region.

### Steps

The script performs the following actions:

- Wait for SRDF/A session consistency
- Suspend DLm heartbeat
- Vary the DLm tape drives offline at DC1
- Run the following DLm commands:
  - DEVSTATUS,VTE,QUIESCE at DC1
  - VTECFGSWAP,VTE,DUMMY at DC1
  - VG8HALT,VG8-1 (*primary-site* EQ DC1) or VG8RESET,VG8-1 (*primary-site* NE DC1)
  - VG8RESET,VG8-1 at DC2
- Stop production workload at DC1 and DC2
- Reset production LPARs at DC1 and DC2
- Handle ConGroup groups
- Force a ConGroup trip
- Verify status of R1 devices
- Run the MSC,PENDDROP command against MSC group A
- Run the MSC,PENDDROP command against MSC group B
- Disable ConGroup
- Suspend SRDF/A devices
- DREFR MSC for MSC groups A and B
- Broadcast GDDR global variables
- Force master C-system transfer to DC4

### Restrictions

Before starting, the script verifies that the script is running from the master C-system at the primary site. If this check fails, the script terminates with GDDR926E script generation error. For details, see the *GDDR Message Guide*.

## Restart production at *DC3/DC4* after site swap (GDD2P18A)

This script is used to restart the single site workload after the 'Abandon Region <n> (Out-of-Region site swap)' script has completed successfully.

### Single-Side Restart

Single-Side Restart is an enhancement to the SQAR recovery process which reduces the mean time to recovery for production workloads by allowing the operator to select Single Side Restart as a call override to the GDD2P18A script.

When Single Side Restart is selected, replication is not resumed on the A-B and B-D legs of the SQAR configuration, which retains the R21 and R22 devices at sites B and D as a Gold Copy of the data. Replication is resumed in MSC mode on the A-C leg of the SQAR configuration, at which time production workloads may be restarted.

The full SQAR configuration may be restored by running script GDDRPA23 any time after script GDD2P18A has completed successfully.

---

**Note:** When using the GDDRMAIN,SCRIPT command, this call override is specified as 1SIDE=Y.

---

### Steps

The script performs the following actions:

- Suspend DLm heartbeat
- Clean ConGroup devices
- Run the SRDF HSWAP command to half-swap the devices
- SRDF SPLIT BCV devices
- Run the SRDF RESUMEPAIR command against SRDF/S devices with check/fixup
- Run the M6 MSC Cleanup utility
- Run the SRDF RDF-RSUM command against SRDF/A devices with check/fixup
- Wait for invalid tracks to synchronize
- Resume replication (RSUM) for recovery devices
- START/REFRESH ConGroup
- Perform a ConGroup takeover
- Run the SRDF HSWAP command to half-swap the external devices
- IPL production LPARs
- Run the following DLm commands at DC3 or DC4:
  - VG8ACT,VG8-1

    - VTECFGSWAP,VTE,PROD
- Vary DLm tape devices online at DC2 or DC4
- Resume the DLm heartbeat issued by GDDR
- Make DLm R2 devices read-only
- Restart production workload
- Re-establish/terminate BCVs
- Activate SRDF/A
- REFRESH/RESTART MSC
- SRDF ADCD/RSUM external devices
- Vary the tape devices online at DC3 or DC4
- Resume DLm heartbeat
- Run the following DLm commands at DC1 (if primary-site NE DC1):
    - VG8CLEANUP,VG8-1,SHUTDOWN
    - VG8CLEANUP,VG8-1,CLEANUP
    - VG8CLEANUP,VG8-1,REBOOTCS
    - VG8CLEANUP,VG8-1,CHECK
- Broadcast GDDR global variables
- Force master C-system transfer to DC3

### Restrictions

The script can only start after the master C-system has transferred from the original primary region to the secondary (new primary) region. The master C-system function is automatically transferred when the 'Abandon Region <n> (Out-of-Region site swap)' script completes successfully. If this check fails, the script terminates with a nonzero return code and error message GDDR639I. For details, see the *GDDR Message Guide*.

## Swap production from *DC1* to *DC2* (GDDRPA42)

This script is used to swap the DASD and the single-site workload from the primary DASD site (DC1) to the secondary DASD site (DC2) with SRDF/S preference.

In environments with geographically dispersed sysplexes supported by cross-site host-DASD channels, coupled dataset and coupling facility structure management is performed as a result of a site move, either planned or unplanned. These actions are controlled by the Realign Couple Datasets and Rebuild Couple Facility Structure parameters. See "Define script sysplex options (M,P,O,S)" on page 341.

### Steps

This script performs the following actions:

- Stop the business workload at the primary DASD site

- Swap the DASD to the secondary DASD site (AutoSwap followed by SRDF/S personality swap)
- Swap the primary and secondary MSC server roles when SRDF/SQAR High Availability is implemented.
- Resume SRDF/S
- Restart the business workload

The detailed steps are as follows:

- Verify SRDF/A status
- Stop workload
- Suspend the DLm heartbeat
- Vary the DLm tape devices offline at DC1
- Run the following DLm commands at DC1:
  - DEVSTATUS,VTE,QUIESCE
  - VTECFGSWAP,VTE,DUMMY
  - VG8HALT,VG8-1
  - VG8RESET,VG8-1
- Check for SRDF/S invalid tracks
- Shut down production LPARs at DC1
- Run the MSC,PENDDROP command
- Swap to R2 devices
- Clear ConGroup flags
- Run the SRDF HSWAP command to half-swap R1 and R2 devices
- Check status of swapped devices
- Make DLm R2 devices read-only at DC1
- Run the following DLm commands at DC2:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape devices online at DC2
- Make DLm R2 devices read-only
- Run the SRDF RDF-RSUM command with check/fixup
- Wait for invalid tracks to synchronize
- Refresh ConGroup
- Start production LPARs at DC2
- Resume SRDF/A devices
- Run the SRDF RDF-RSUM command with check/fixup
- Resume SRDF/A
- Switch primary/secondary MSC servers
- Perform ConGroup takeover
- Complete MSC server swap
- Wait for SRDF/A session consistency
- Realign coupled datasets
- Manage CF structures
- Run the following DLm commands at DC1 (if *primary-site* NE DC1):
  - VG8CLEANUP,VG8-1,SHUTDOWN
  - VG8CLEANUP,VG8-1,CLEANUP
  - VG8CLEANUP,VG8-1,REBOOTCS
  - VG8CLEANUP,VG8-1,CHECK
- Resume DLm heartbeat

- Broadcast GDDR global variables
- Transfer master C-system

### Restrictions

The script can only be run from the current master C-system.

## Swap DASD from *DC1* to *DC2* (GDDRPA21)

This script is used to swap only the DASD from the primary DASD site (DC1) to the secondary DASD site (DC2) with SRDF/S preference.

### Steps

This script performs the following actions:

- Stop the business workload at the primary DASD site
- Swap the DASD to the secondary DASD site (AutoSwap followed by SRDF/S personality swap)
- Swap the primary and secondary MSC server roles when SRDF/SQAR High Availability is implemented.
- Resume SRDF/S
- Restart the business workload

The detailed steps are as follows:

- Verify SRDF/A status
- Stop workload
- Suspend the DLm heartbeat
- Vary the DLm tape devices offline at DC1
- Run the following DLm commands at DC1:
  - DEVSTATUS,VTE,QUIESCE
  - VTECFGSWAP,VTE,DUMMY
  - VG8HALT,VG8-1
  - VG8RESET,VG8-1
- Check for SRDF/S invalid tracks
- Run the MSC,PENDDROP command
- Swap to R2 devices
- Clear ConGroup flags
- Run the SRDF HSWAP command to half-swap R1 and R2 devices
- Check status of swapped devices
- Make DLm R2 devices read-only at DC1
- Run the following DLm commands at DC2:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape devices online at DC2
- Make DC3 DLm R2 devices read-only
- Run the SRDF RDF-RSUM command with check/fixup.
- Wait for invalid tracks to synchronize
- Refresh ConGroup
- Start production LPARs at DC2
- Resume SRDF/A devices

- Run the SRDF RDF-RSUM command with check/fixup
- Resume SRDF/A
- Switch primary/secondary MSC servers
- Perform ConGroup takeover
- Complete MSC server swap
- Wait for SRDF/A session consistency
- Realign coupled datasets
- Run the following DLm commands at DC1 (if *primary-site* NE DC1):
  - VG8CLEANUP,VG8-1,SHUTDOWN
  - VG8CLEANUP,VG8-1,CLEANUP
  - VG8CLEANUP,VG8-1,REBOOTCS
  - VG8CLEANUP,VG8-1,CHECK
- Resume DLm heartbeat
- Broadcast GDDR global variables
- Transfer master C-system

## Restrictions

The script can only be run from the current master C-system. If you are employing the z/OS Enhanced Catalog Sharing (ECS) feature, see "GDDR AutoSwap and restart of Enhanced Catalog Sharing" on page 563 for information about ECS restart.

# Test scripts

Choose the test script you want to run from the Scripts for TEST Actions list in the Select Script to Run panel (S). Follow the instructions in "Selecting managed systems for test scripts" on page 526 to select managed systems to be included in a script run.

**Note:** See "Site and region designations" on page 519 for information about site designations.

## Perform test IPL from BCVs at *DC2/DC3/DC4* (GDD2P01A)

This script is used to IPL contingency systems at DC2/DC3/DC4 using BCV devices.

### Steps

The script performs the following actions:

- Wait for SRDF/A session consistency
- SRDF REESTABLISH/SPLIT/SPLITINSTANT the BCVs
- Run the following DLm commands at DC2, DC3, or DC4:
  - VG8BCV,VG8-2,MOUNT
  - VTECFGSWAP,VTE,DRTEST
- Vary the tape drives online at DC2, DC3, or DC4
- Activate and IPL LPARs from BCVs

### Restrictions

The script can only be run from the current master C-system.

## Resume after test IPL from BCVs at *DC2/DC3/DC4* (GDD2P02A)

This script is used to reset clear contingency systems after a test at Dc2, DC3, or DC4.

### Steps

The script performs the following actions:

- Stop test business workload, if applicable
- Vary the tape DLm tape drives offline at DC2, DC3, or DC4
- Run the following DLm commands at DC2, DC3, or DC4:
  - DEVSTATUS,VTE,QUIESCE
  - VTECFGSWAP,VTE,DUMMY
  - VG8BCV,VG8-2,UNMOUNT
- Reset the LPARs used during the test IPL
- Reestablish/terminate the BCVs

### Restrictions

The script can only be run from the current master C-system.

The following prerequisite must also be met before running the script:

- All DC3 or DC4 testing is complete.

## Perform test IPL from R2s at *DC2* (GDD2P03A)

This script is used to IPL contingency systems at DC1 or DC2, depending on which site is currently the secondary DASD site.

### Steps

The script performs the following actions:

- Confirm that SRDF/A has been stopped normally via an SRDF/A PENDDROP
- Activate LPARs and IPL test z/OS systems using R2 volumes
- IPL hosts at DC2 for test purposes and start test business workload, if applicable

The detailed script steps are as follows:

- Wait for SRDF/A consistency
- Reestablish/terminate BCVs
- Stop workload at DC2
- Reset LPARs at DC2
- Clean ConGroup devices
- Force a ConGroup trip
- Check status of R2 devices at DC2
- Run the MSC,PENDDROP command
- Set the R2 devices to RDY/RW/RDFRDY
- SPLIT BCVs at DC2
- IPL the test LPARs at DC2
- Run the following DLm commands at DC2:
  - VG8ACT,VG8-1,SKIP_SITEA_SHUTDOWN
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape drives online at DC2
- Start test application workload at DC2

### Restrictions

The script can only be run from the current master C-system.

## Perform test IPL from R2s at *DC3/DC4* (GDDRPA27)

This script is used to IPL contingency systems at DC3 using R2 devices.

### Steps

The script performs the following actions:

- Confirm that SRDF/A has been stopped normally via an SRDF/A PENDDROP
- Activate LPARs and IPL test z/OS systems using R2 volumes
- Start test business workload, if applicable

### Restrictions

The script can only be run from the current master C-system.

## Resume after test IPL at *DC3/DC4* (GDDRPA28)

This script is used to reset clear contingency systems after a test at DC3 or DC4. This script restores the SRDF/A link to DC3 or DC4 (either from DC1 or DC2 depending upon where the production workload is currently running) after a test on DC3 or DC4.

### Steps

This script performs the following tasks:

- Reset clear all systems IPLed during the test
- Deactivate all LPARs previously activated for the test
- Restart SRDF/SQAR with AutoSwap

The detailed script steps are as follows:

- Stop the test application workload at DC3 or DC4
- Vary the tape drives offline at DC3 or DC4
- Run the following DLm commands at DC3 or DC4:
  - DEVSTATUS,VTE,QUIESCE
  - VTECFGSWAP,VTE,DUMMY
  - VG8REST,VG8-1,SEC,SKIP_SITEA_SHUTDOWN
- GDDR resets the test LPARs at DC3 or DC4
- SRDF sets R2s to RO/REFR/RFRSUM
- Activate SRDF/A
- REFRESH MSC
- Wait for SRDF/A consistency

### Restrictions

The script can only be run from the current master C-system.

The following prerequisite must also be met before running the script:

- All DC3 or DC4 testing is complete.

## Resume after test IPL from R2s at *DC2* (GDD2P16A)

This script is used to reset clear contingency systems after a test at DC2.

### Steps

The script performs the following actions:

- Stop test business workload, if applicable
- Reset clear test system LPARs
- Restart SRDF/S to DC2
- Restart SRDF/S with AutoSwap to DC2

The detailed script steps are as follows:

- Stop test application workload at DC2
- Reset test LPARs at DC2
- SRDF REESTABLISH the BCVs at DC2-DC1
- Set SRDF SYNCDIR TOR2
- SRDF REFR/RFRSUM DC2-DC1
- RESUME ConGroup
- SRDF REESTABLISH the BCVs at DC4-DC2
- Resume SRDF replication (RSUM) at DC2-DC4
- Run the SRDF RDF-RSUM command at DC2-DC4
- Activate SRDF/A at DC2
- REFRESH MSC
- Wait for SRDF/A consistency
- IPL production LPARs
- Start production workload
- REESTABLISH the BCVs at all sites
- SRDF makes the DLm R2 devices read-only (RO)
- Run the following DLm commands at DC2 (if *secondary-site* NE DC1):
  - VG8CLEANUP,VG8-1,SHUTDOWN
  - VG8CLEANUP,VG8-1,CLEANUP
  - VG8CLEANUP,VG8-1,REBOOTCS
  - VG8CLEANUP,VG8-1,CHECK

### Restrictions

The script can only be run from the current master C-system.

The following prerequisite must also be met before running the script:

- All DC2 testing is complete.

# Unplanned scripts

GDDR unplanned script processes are invoked by one of two methods:

- The GDDR Event Monitor prompts the operator to request management confirmation of a trigger event or events which may indicate an outage before the script is permitted to start.
- The operator submits a script from the Scripts for UNPLANNED Actions list in the Select Script to Run panel (S).

GDDR validates the status of the current host and storage environment against script rules before the script is allowed to begin processing. The steps you need to take to run these scripts are described in "Running scripts via GDDR ISPF interface" on page 524.

**Note:** See "Site and region designations" on page 519 for information about site designations.

## Recover after unplanned swap (GDDRUP31)

After an unplanned swap, the GDDR Event Monitor submits this script. After the loss of the primary site, this script is used to restart processing at the secondary site.

### Steps

The script steps are as follows:

- Stop SRDF/A
- Clear ConGroup flags
- Suspend DLm heartbeat
- Vary the DLm tape drives offline at DC1
- Run the following DLm commands at DC1:
  - VTECFSWAP,VTE,DUMMY
  - VG8RESET,VG8-1
- Run the SRDF HSWAP command to half-swap the devices
- Check status of devices
- Make DLm R2 devices read-only
- Activate and IPL LPARs at DC2
- Run the following DLm commands at DC2:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape drives online at DC2
- Resume DLm heartbeat
- Stop the MCS message scanner

This script performs the following restart actions after an unplanned swap has completed successfully:

- With SRDF/S preference (default):
  - GDDR determines whether this recovery type is possible. If not, then the SRDF/A preference recovery is performed.
  - Perform the required HSWAP operations for an intra-region site swap
  - Split BCVs at DC1
  - Resume synchronous replication from DC2 to DC1
  - Start ConGroup
  - Start workload at DC2 (if applicable)
  - Split BCVs at DC3
  - Resume SRDF/A from DC1 to DC3
  - Switch MSC definitions, preserving SQAR mode of operation
  - Move AutoSwap owner to DC1
  - Manage coupled datasets and CF Structures (if applicable)
  - Re-establish BCVs at DC1 and DC3
  - Transfer master C-system function to DC1
- With SRDF/A preference:
  - This version requires you to run script GDDRPA0A at a later time, to resume synchronous replication from DC2 to DC1.
  - Perform HSWAP operations at DC2, DC3, and DC4
  - Start workload at DC2 (if applicable)
  - Manage coupled datasets and CF structures (if applicable)

### Restrictions

The GDDR Event Monitor and GDDR Heartbeat Monitor must be active on each C-system. The Automation flag on the GDDR ISPF dashboard (described in "GDDR dashboard" on page 282) must be ON. See "Changing GDDR automation state" on page 286 for details.

## Recover after loss of *DC1* (LDR) (GDD2U13A)

This script restarts the single site workload after the GDDR Event Monitor has detected a local disaster (LDR) at the primary site. The following events occurring concurrently are considered a local disaster: loss of SRDF/A, loss of SRDF/S, or a ConGroup trip, or the primary site GDDR Heartbeat Monitor has timed out.

Loss of DC1 can occur through a sequence of events which may occur in different order. From a data protection perspective, there are two important sequences:

- Type 1: loss of channel access occurs first, then replication is lost
- Type 2: remote replication is lost first, then channel access is lost

GDD2U13A covers both scenarios. GDDR blocks this script if DC3 has more recent data.

## Steps

The script performs the following actions:

- Warn user about potential data loss
- Stop workload at DC1 (if applicable)
- Perform HSWAP operations at DC2, DC3 and DC4
- Split BCVs at DC4 (Type 2 only)
- Resume replication DC2 to DC4 (Type 2 only)
- Start SRDF/A in MSC mode, DC2 to DC4
- Reestablish/terminate BCVs at DC4 (Type 2 only)
- Start workload at DC2 (if applicable)
- Manage coupled datasets and CF structures (if applicable)

The detailed script steps are as follows:

- Stop production workload at DC1
- Vary the DLm tape drives offline at DC1
- Run the following DLm commands at DC1:
  - DEVSTATUS,VTE,QUIESCE
  - VTECFGSWAP,VTE,DUMMY
  - VG8HALT,VG8-1
- Reset production LPARs at DC1
- Run the M9 utility to stop SRDF/A
- Clean ConGroup devices
- Clear ECA sessions
- Run the SRDF HSWAP command to half-swap SRDF devices
- Make the DLm R2 devices read-only
- IPL LPARs at DC2
- Run the following DLm commands at DC2:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape devices online at DC2
- Start production workload at DC2
- DREFR/RESREF MSC
- Restart ConGroup at DC2 in MSC mode

### Restrictions

The GDDR Event Monitor and GDDR Heartbeat Monitor must be active on each C-system. The Automation flag on the GDDR ISPF dashboard (described in "GDDR dashboard" on page 282) must be ON. See "Changing GDDR automation state" on page 286 for details.

## Resume replication after unplanned swap (GDDRPA0A)

This script is used to resume the SRDF/S link to the secondary DASD site after unplanned swap (due to the loss of the primary DASD).

### Steps

This script performs the following actions:

- Clean ConGroup devices
- Vary the DLm tape drives offline at DC1
- Run the following DLm commands at DC1:
  - DEVSTATUS,VTE,QUIESCE
  - VTECFGSWAP,VTE,DUMMY
  - VG8RESET,VG8-1
- Run the SRDF HSWAP command to half-swap R2 devices
- Process ConGroup groups
- Run the RDF-RSUM command with check/fixup
- Wait for invalid tracks to synchronize
- REFRESH ConGroup
- SRDF makes R2 devices read-only
- Run SRDF SUSPEND/RESUME commands against R1 devices
- Run the RDF-RSUM command with check/fixup
- Activate SRDF/A
- Switch primary/secondary MSC servers
- Perform a ConGroup takeover
- Complete MSC server swap
- Wait for SRDF/A session consistency
- Run the following DLm commands at DC1 (if *primary-site* NE DC1):
  - VG8CLEANUP,VG8-1,SHUTDOWN
  - VG8CLEANUP,VG8-1,CLEANUP
  - VG8CLEANUP,VG8-1,REBOOTCS
  - VG8CLEANUP,VG8-1,CHECK
- Update/broadcast GDDR global variables

### Restrictions

The script is displayed in the Select Script to Run on GDDRPLEX panel (S) of the current master C-system after completion of the 'Recover after loss of DC1 (LDR)' script.

If the 'Recover after loss of DC1 (LDR)' script failed to complete the transfer of the master C-system to the new primary site, complete the transfer using Setup and Maintenance Menu panel option T, Transfer Master C-System, before starting this script.

## Resume replication after loss of *DC1* (GDDRPA0A)

This script is used to resume the SRDF/S link to the secondary site after a primary site disaster. This script also restarts SRDF/A in SRDF/SQAR with AutoSwap mode.

### Steps

This script performs the following actions:

- HSWAP at DC1
- Split BCVs at DC1
- Resume SRDF/S from DC2 to DC1
- Start ConGroup
- Split BCVs at DC3
- Resume replication DC1 to DC3
- Activate SRDF/A DC1 to DC3
- Swap SQAR mode definitions
- Transfer AutoSwap owner to DC1
- Reestablish/terminate BCVs at DC1 and DC3
- Transfer master C-system function to DC1

### Restrictions

The script is displayed in the Select Script to Run on GDDRPLEX panel (S) of the current master C-system after completion of the 'Recover after loss of DC1 (LDR)' script.

If the 'Recover after loss of DC1 (LDR)' script failed to complete the transfer of the master C-system to the new primary site, complete the transfer using Setup and Maintenance Menu panel option T, Transfer Master C-System, before starting this script.

Following completion of the 'Recover after loss of DC1 (LDR)' script, the C-systems at Site A and Site B may still have the UCBs (Unit Control Blocks) for the MSC Star gatekeeper devices at the old primary site ONLINE.

Perform the following actions on both C-systems before starting the 'Resume replication after loss of DC1' script:

- Vary OFFLINE for the UCBs representing the MSC Star gatekeeper devices at the old primary site.
- Vary ONLINE for the UCBs representing the MSC Star gatekeeper devices at the new primary site.

No action is needed on the GDDR-managed systems, as the MSC SQAR gatekeeper devices are typically OFFLINE to the production systems.

# Resumption scripts

Choose the resumption script you want to run from the Scripts for RESUME Actions list in the Select Script to Run panel (S).

**Note:** See "Site and region designations" on page 519 for information about site designations.

## Resume SRDF/S replication after ConGroup trip (GDDRPA23)

This script resumes SRDF/S replication and reestablishes/terminates the BCVs at the secondary DASD site if applicable.

This script is used to:

- Resume SRDF/S replication
- Reestablish/terminate the BCVs at the secondary DASD site, if applicable

### Steps

This script performs the following actions:

- Run the M9 utility to stop SRDF/A
- Split the BCV devices at DC4
- Run the SRDF SUSP/ADCD/RSUM commands against devices replicating from DC2 to DC3
- Run the RDF-RSUM command against devices replicating from DC2 to DC4 with check/fixup
- Split the BCV devices at DC2
- Set SRDF NADC mode on devices at DC1
- Resume ConGroup
- Activate SRDF/A from DC2 to DC4
- REFRESH/RESTART MSC
- Activate ConGroup in SQAR mode
- Re-establish/terminate BCVs at DC2 and DC4

- Activate ConGroup in SQAR HA mode

### Restrictions

This script must only be run after resolving the problem that resulted in the consistency group trip.

---

**Note:** The confirmation panels display the current secondary DASD site. Depending on the current value, the secondary DASD site is displayed as either DC1, DC2, DC3, or DC4.

---

## Resume SRDF/A in MSC mode to DC3/DC4 (GDDRPM29)

This script restores the SRDF/A link to DC3 or DC4 in MSC mode (from either DC1 or DC2 depending upon where the production workload is currently running) after a planned or unplanned swap (a test at DC3 or DC4).

The confirmation panel displays as either DC1 or DC2 (depending on the value of the current primary DASD site) as the "from" site.

### Steps

The script performs the following actions:

- Run the M9 utility to stop SRDF/A
- Make the R2 devices read-only at DC3
- Split the BCV devices at DC3
- Run the SRDF ADCD/SUSP/RSUM commands against devices replicating from DC1 to DC3
- Run the SRDF RSUM command against devices replicating from DC1 to DC3 with check/fixup
- Run the M6 utility to destroy old SQAR leg DC1 to DC3
- Activate SRDF/A from DC1 to DC3
- Start MSC from DC1 to DC3 on the primary server
- Wait for MSC consistency
- Start MSC from DC1 to DC3 on the secondary server
- Make R2 external devices read-only at DC3
- Run the SRDF ADCD/SUSP/RSUM commands against external devices replicating from DC1 to DC3
- Re-establish/terminate BCVs at DC3

### Restrictions

The script can only be run from the current master C-system.

The following prerequisites must also be met before running the script:

- SRDF/A must be down for the leg you want to run GDDRPM29 against.

- There must not be an SRDF event for the site pair for which you want to resume SRDF/A.
- In the primary region, one or more of the following conditions must be true:
  - There must be an RDF event for the current tertiary DASD site and current quaternary DASD site pair or
  - There must be an RDF event for the SRDF/A site pair opposite of the one GDDRPM29 will resume or
  - There must be an LDR event for the SRDF/A R1 site opposite of the one GDDRPM29 will resume.
- In the secondary region, GDDRPM29 is allowed on the SRDF/A leg that was activated earlier by a GDDRPA06 or GDDRPA07 script.

## Resume SRDF/A in SQAR mode (GDDRPF29)

This script restores the SRDF/A links to the secondary region sites in SQAR mode after a planned or unplanned stop of SRDF/A (a test at the secondary region).

**Note:** The confirmation panel displays as either DC1 or DC2 (depending on the value of the current primary DASD site) as the “from” site.

### Steps

The script performs the following actions:

- Run the M9 utility to stop SRDF/A
- Make the R2 devices read-only at DC3
- Split the BCV devices at DC3
- Run the SRDF ADCD/SUSP/RSUM commands against devices replicating from DC1 to DC3
- Run the SRDF RSUM command with check/fixup against devices replicating from DC1 to DC3
- Activate SRDF/A from DC1 to DC3
- Deactivate SQAR on the secondary MSC server
- Restart SQAR on the primary MSC server
- Wait for SRDF/A session consistency
- Restart SQAR on the secondary MSC server
- Make R2 external devices read-only at DC4
- Split the BCVs devices at DC3
- Run the SRDF ADCD/SUSP/RSUM commands against external devices replicating from DC1 to DC3
- Re-establish/terminate BCVs at DC3

### Restrictions

The script can only be run from the current master C-system.

The following prerequisites must also be met before running the script:

- All DC3 or DC4 testing is complete.
- All contingency systems running in DC3 or DC4 are closed down.

## Reclaim Secondary/Tertiary/Quaternary site (*DC2/DC3/DC4*) (GDDRPA65)

This script allows operations personnel to restore normal operations after a site has been abandoned for maintenance.

### Steps

The script performs the following actions:

- To reclaim the secondary site:
  - Make the R2 devices read-only at DC4
  - Split the BCV devices at DC2
  - Split BCV devices at DC4
  - Resume ConGroup
  - Run the RDF-RSUM command against devices replicating from DC2 to DC4
  - Run the RDF-RSUM command with check/fixup against devices replicating from DC3 to DC4
  - Activate SRDF/A
  - Refresh MSC on the secondary server
  - Refresh MSC on the primary server
  - Start ConGroup in MSC mode
  - Refresh MSC on the primary server
  - Restart MSC on the primary server
  - Verify that SQAR Recovery is available
  - Restart MSC on the secondary server
  - Make external devices read-only at DC4
  - Run the SRDF ADCD/SUSP/RSUM commands against external devices replicating from DC2 to DC4
  - Re-establish BCVs at DC2
  - IPL LPARs at DC2
  - Start workload at DC2
  - Re-establish/terminate BCVs at DC4
- To reclaim the tertiary site:

- Make the R2 devices read-only at DC3
- Split the BCV devices at DC3
- Run the SRDF ADCD command against devices replicating from DC1 to DC3
- Run the RDF_RSUM command against devices replicating from DC1 to DC3
- Run the RDF_RSUM command with check/fixup against devices replicating from DC1 to DC3
- Activate SRDF/A
- Restart MSC on the primary server
- Verify that SQAR Recovery is available
- Restart MSC on the secondary server
- Make external devices read-only at DC3
- Run the SRDF ADCD/SUSP/RSUM commands against external devices replicating from DC1 to DC3
- Re-establish/terminate BCVs at DC3

◆ To reclaim the quaternary site:

- Make the R2 devices read-only at DC4
- Split the BCV devices at DC4
- Run the RDF_RSUM command against devices replicating from DC2 to DC4
- Run the RDF_RSUM command with check/fixup against devices replicating from DC2 to DC4
- Activate SRDF/A
- Restart MSC on the primary server
- Verify that SQAR Recovery is available
- Restart MSC on the secondary server
- Make external devices read-only at DC4
- Run the SRDF ADCD/SUSP/RSUM commands against external devices replicating from DC2 to DC4
- Re-establish/terminate BCVs at DC4

## Restrictions

For configurations with AutoSwap, note the following considerations when using GDDRPA65 to reclaim the secondary DASD site:

◆ Before running this script, the master C-system must be moved to the secondary DASD site.

◆ After the script completes, the AutoSwap owner must be moved to the secondary DASD site. If the configuration has MSC, then the primary MSC server must also be moved to the secondary DASD site.

- GDDR offers scripts (GDDRPXMC, GDDRPXAS, GDDRMMSC) to perform these moves.

# RDR scripts

Choose the RDR script you want to run from the Scripts for RDR Actions list in the Select Script to Run panel (S). These scripts are used to perform secondary region operations.

**Note:** See "Site and region designations" on page 519 for information about site designations.

## Options for recovery after loss of primary region

GDDR offers the following planned site swaps:

- Intra-region (GDDRPA21, and GDDRPA42)
- Inter-region (GDD2P17A + GDD2P18A)

The inter-region option supports the following alternatives:

- GDD2P17A + GDD2P18A: This option completes a full inter-region site swap, and requires availability of all 4 sites, primary region becomes secondary region and vice-versa.
- GDD2P17A + GDDRPA06: This option abandons the primary region, restarts in the secondary region, with SRDF/A back to a surviving primary region site
- GDD2P17A + GDDRPA05: This option abandons the primary region, restarts in the secondary region, with no access to either of the primary region sites, and therefore provides no out of region recovery.

A shorter version of the inter-region alternatives is as follows:

- GDD2P17A can be followed by GDD2P18A to complete a site swap,
- GDD2P17A can be followed by GDDRPA06 if one of the primary sites is not available for use when processing restarts in the other region,
- GDD2P17A can be followed by GDDRPA05 if both primary sites are unavailable for use when processing restarts in the other region

## Recover at DC3/DC4 after RDR in primary region/Recover at DC3/DC4 with SRDF/S to DC4/DC3 (GDDRPA05)

These scripts are used to restart the production systems at the tertiary site or quaternary site.

Choose one of these scripts to run in the event of a major failure that prevents the production workload from being run from either DC1 or DC2. The script warns the user if DC2 has more recent data than the secondary region.

The GDDRPA05 script should only be run if there is a regional disaster which causes outages at both DC1 and DC2 or if DC1 and DC2 have been abandoned using the GDD2P17A script. It activates LPARs and restarts the production z/OS systems at the

chosen site. The script can be run using either DC3 or DC4 for single site recovery, or using DC3 or DC4 as SRDF/S source site, using DC4 or DC3 as SRDF/S target site. If SRDF/S is desired, you are presented with an option to drain data from the site with the most recent data, limited to DC3 and DC4.

### Steps

This script performs the following actions:

- Activate all needed LPARs
- Activate CBU (if required)
- Create a consistency point at DC3 or DC4
- Prepare SRDF environment
- IPL all needed production systems

The detailed script steps are as follows:

- Run the M9 utility to stop SRDF/A
- SRDF SPLIT BCVs at DC3 or DC4
- Set SRDF devices at DC3 or DC4 RDY/RW/RDFRDY
- Clean ConGroup devices
- IPL LPARs at DC3 or DC4
- Make DLm R2 devices read-only
- Run the following DLm command at DC3 or DC4:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape drives online at DC3 or DC4
- Start production workload at DC3 or DC4

### Restrictions

The script must be run from the C-system at DC3 or DC4. If it is not run from the appropriate C-system at the secondary region, the script terminates with a GDDR926E script generation error. For details, see the *GDDR Message Guide*.

## Restart production at DC3/DC4 with cascaded/concurrent SRDF (GDDRPA06)

These scripts are designed to be run after GDD2P17A has run. They are used to restart the production systems at the tertiary site or quaternary site, with SRDF/A back to either DC1 or DC2 depending on which site is available as an SRDF/A target site.

This script should only be run if there is a major failure that prevents the production workload from being run from either DC1 or DC2. It restarts the production z/OS systems at the chosen site and reestablishes SRDF/A to the primary region.

GDDRPA06 is available in four variations in SRDF/SQAR configurations:

- DC3 source site, with cascaded replication DC3 -> DC4 and DC4 -> DC2
- DC3 source site, with concurrent replication, DC3 -> DC4 and DC3 -> DC1
- DC4 source site, with cascaded replication DC4 -> DC3 and DC3 -> DC1
- DC4 source site, with concurrent replication DC4 -> DC3 and DC4 -> DC2

In concurrent or 'inactive cascaded' configurations, SRDF/A is started back to the primary DASD site, while in active cascaded configurations, SRDF/A is started back to the secondary DASD site.

## Steps

This script performs the following actions:

- Activate all needed LPARs
- Activate CBU (if required)
- Create a consistency point at DC3 or DC4
- Prepare SRDF environment
- IPL all needed production systems
- Start SRDF/A in MSC mode

The detailed steps are as follows:

- Clear ECA sessions
- Split BCVs at DC3 or DC4
- Run the SRDF HSWAP command against SRDF devices
- Run the SRDF RDF-RSUM command with check/fixup
- Wait for invalid tracks to synchronize
- Clean ConGroup devices
- START/REFRESH ConGroup
- Perform a ConGroup takeover
- Run the SRDF HSWAP command against SRDF devices
- Make DLm R2 devices read-only
- IPL LPARs at DC3 or DC4
- Run the following DLm command at DC3 or DC4:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape drives online at DC3 or DC4
- Start production workload at DC3 or DC4
- Run the SRDF RDF-RSUM command with check/fixup
- Run the M6 MSC Cleanup utility
- Activate SRDF/A
- REFRESH/RESTART MSC
- Wait for SRDF/A device consistency
- Start the secondary MSC server
- Broadcast GDDR global variables

## Restrictions

The script must be run from a secondary region C-system. If it is not run from a secondary region C-system, the script terminates with a GDDR926E script generation error. For details, see the *GDDR Message Guide*.

IMPORTANT

Running this script risks data loss if GDD2P17A was not run successfully first. There is a potential for being two SRDF/A cycles behind in asynchronous operations from the old primary DASD site to the secondary region. Obtain management approval before executing this script.

## Recover at DC3 with cascaded SRDF DC3->DC4, DC4->DC2/Recover at DC4 with concurrent SRDF DC4->DC3, DC4->DC2 (GDDRPA07)

Choose one of these scripts to run if there is a local disaster that prevents the production workload from being run from DC1. These scripts restart the production LPARs at the chosen site and reestablishes concurrent or cascaded SRDF (SRDF/A) to to a surviving primary region site (DC2). Note that GDDRPA07 with cascaded replication is not allowed unless DC3 (site C) has the most recent data.

This script will only be run in the event of a local disaster (LDR) that prevents the production workload from being run at DC1. The Recover at DC3 with Cascaded SRDF version of this script is blocked if Site B or Site D have more recent data. The considerations for restarting at DC3 the secondary region are:

- There was a ConGroup trip before DC1 was lost. In that case the secondary region will have more recent data than DC2.
- There are signs that DC2 may be in danger as well as DC1.

The distinction between this script and the GDDRPA06 'Restart production at DC3/DC4 with concurrent/cascaded SRDF' script is that GDDRPA06 is designed to run after GDD2P17A, as a planned operation. GDDRPA07 is designed to run after an unplanned loss of the primary DASD site.

### Steps

These scripts perform the following actions:

- Activate all needed LPARs including CFs
- Activate CBU (if required)
- Create a consistency point
- Prepare SRDF environment
- IPL all needed production systems
- Start SRDF/A in MSC mode

The detailed steps are as follows:

- Run the M9 utility to stop SRDF/A
- Clear the ECA sessions
- Clean ConGroup devices
- Run the SRDF HSWAP command to half-swap devices
- Check status of SRDF devices
- SPLIT BCVs at DC3 or DC4
- Run the RDF_RSUM command with check/fixup
- Wait for invalid tracks to synchronize

- Clean ConGroup devices
- Start/refresh ConGroup
- Perform a ConGroup takeover
- Run the SRDF HSWAP command to half-swap devices
- Make DLm R2 devices read-only
- IPL LPARs at DC3 or DC4
- Run the following DLm commands at DC3 or DC4:
  - VG8ACT,VG8-1
  - VTECFGSWAP,VTE,PROD
- Vary the DLm tape drives online at DC3 or DC4
- Start production workload at DC3 or DC4
- SPLIT BCVs on the SQAR recovery leg
- Run the M9 MSC Cleanup utility
- Activate SRDF/A on the SRDF groups
- REFRESH/RESTART MSC
- Wait for SRDF/A session consistency
- Start the secondary MSC server

## Restrictions

The script must be run from a secondary region C-system. If it is not run from a secondary region C-system, the script terminates with a GDDR926E script generation error. For details, see the *GDDR Message Guide*.

# Special scripts

Choose the special script you want to run from the Scripts for SPECIAL Actions list in the Select Script to Run panel (S).

**Note:** See "Site and region designations" on page 519 for information about site designations.

## Planned STAR-HA Takeover (GDDRPA36)

This script transfers the MSC owner.

The script performs the following actions:

- Deactivate MSC groups A and B on the primary MSC server
- Perform a ConGroup takeover
- Perform an MSC takeover
- Make the old primary the secondary MSC server for MSC groups A and B
- Wait for SQAR HA

**Note:** This script swaps primary and secondary MSC server roles without interruption of replication or MSC function.

## Unplanned STAR-HA Takeover (GDDRUP35)

This script transfers the MSC owner.

The script performs the following actions:

- Perform a ConGroup takeover
- Perform an MSC takeover
- Wait for SQAR HA

## Transfer AutoSwap Owner to DC*n* (GDDRPXAS)

This script performs an AutoSwap takeover.

The site shown, DC*n*, is the opposite of the site which is currently the AutoSwap owner.

## Restart primary MSC server at DC*n* (GDDRMMSC)

The script restarts the primary MSC server at the indicated site.

The script performs the following actions:

- Create a BCV copy at both Site C and Site D
- Perform MSC,REFRESH, copy RDFPARM, and perform PARM_REFRESH for MSC groups A and B

- Wait for SRDF/SQAR Recovery to be available
- Re-establish/terminate BCVs at Site C and Site D

The site shown, DC*n*, is variable. In general it is the site which has the primary MSC server role. Typically this is the current secondary DASD site, but either site in the primary region can be shown.

**Note:** This script causes a temporary interruption of MSC functionality.

### Moving primary MSC server to a different site

When used to move the primary MSC server to a different site, the GDDRMMSC script performs the following actions:

- Deactivate MSC groups SQARA and SQARB on the primary MSC server
- Perform a ConGroup takeover
- Perform an MSC takeover
- Wait for SQAR Recovery to be available

## Restart secondary MSC server at DC*n* (GDDRPA36)

This script transfers the MSC owner.

The script performs the following actions:

- Add the old primary server as the secondary MSC server for MSC groups A and B
- Wait for SQAR HA

The site shown, DC*n*, is variable. In general it is the site which has the secondary MSC server role. Typically this is the current primary DASD site, but either DC1 or DC2 in the primary region can be shown.

## Transfer Master C System to DC*n* (GDDRPXMC)

The script transfers the master C-system to the indicated site.

The script performs the following actions:

- Broadcast global variables
- Transfer the master C-system function to the site specified

One GDDRPXMC script is shown for each site in the configuration which is not currently the master C-system site.

## Global variable backup (GDDRPGVB)

This script performs complete backup of GDDR global variables from the current master C-system to a PDS member.

The PDS is defined to GDDR in the GDDR Parameter Wizard. The member name is generated at run-time and reflects the time down to the second.

# Move systems to alternate CPC (GDDRMCPC)

This script performs a planned CPC swap.

This script moves all systems that have both a "regular" LPAR slot and a Recovery LPAR slot defined to their Recovery slot or back to their regular slot if they are currently running in their Recovery slot.

This script is available for use as Option W, CPC Swap on the Perform HMC LPAR Actions panel (A,L).

## Steps

The script performs the following actions:

- Find LPAR Recovery-protected systems on the selected CPC
- Perform a RESET CLEAR on any systems running in their designated Recovery slot
- Perform a RESET CLEAR on the LPAR Recovery protected systems in their current slot
- Perform a LOAD CLEAR for the LPAR Recovery protected systems in their target slot

## Restrictions

A planned CPC swap is much more restrictive than unplanned CPC recovery, and if the following conditions are not met, the swap will not occur:

1. There must be at least one recoverable system on the input CPC.
2. The recoverable systems on the input CPC must all have recovery defined to the same CPC (the target CPC). See the Recovery Site, CPC, and LPAR parameter descriptions on page 326.
3. The recoverable systems on the input CPC must all be either at home or must all be away; no mixture.
4. The recoverable systems on the target CPC (if there are any), must all have recovery defined to the input CPC.
5. The recoverable systems on the target CPC must all either be at home or away; no mixture.

In addition, workload is not stopped prior to the swap and is not restarted once the swap is complete. This is a user responsibility.

# CHAPTER 9
# Handling Unplanned Events

This chapter includes the following topics:


- Overview ..... 562
- Consistency group trip ..... 562
- Unplanned swap ..... 562
- Local and regional disaster ..... 564
- Loss of AutoSwap owner ..... 566
- System failure ..... 567
- GDDR master function transfer ..... 570

# Overview

This chapter describes some unplanned events you may need to deal with and the steps you need to take to deal with them. The events include:

- Consistency group trip
- Unplanned swap
- Local and regional disaster
- Loss of AutoSwap owner
- System failure
- GDDR master function transfer

# Consistency group trip

A consistency group trip is usually seen as a single independent unplanned event. However, in some disaster scenarios, a consistency group trip may indicate the start of a rolling disaster. In this case, the consistency group trip may not be seen on all systems or sites. For example, a consistency group trip is always seen by the primary DASD site, but may not be seen at the secondary site if the C-system observing the ConGroup trip is lost before it can propagate the event to the other C-systems.

# Unplanned swap

An unplanned swap occurs when one or more systems lose access to one or more primary DASD devices (source R1s) because of a *no paths* or *intervention required* condition.

AutoSwap automatically detects this condition and swaps all devices to their respective secondary DASD devices (target R2s). This swap is transparent to the z/OS application systems running from the primary DASD.

---

**Note:** All FBA disks (R1s) in the Enterprise consistency group are also made *Not Ready* at the same time. This eventually causes timeouts on all attached open systems servers.

---

GDDR detects the *no paths* or *intervention required* AutoSwap condition and displays the following on the system consoles and in the SYSLOG:

```
GDDR CAX Warning ===>
************************************************************
*********     GDDR Detected CAX Event in Progress  *********
*********          ...waiting for Unplanned SWAP.  *********
************************************************************
```

Upon detecting the successful swap to the secondary DASD devices, GDDR prompts you to confirm the run of the "Recover after an unplanned swap" script to reconfigure SRDF/S and SRDF/A to reflect the unplanned event change.

```
GDDR Confirmation ===>
|-------------------------------------------------------|
|               Please Confirm Run of ALLSITE n…        |
|                  *** Un-Planned CAX Swap ***          |
|                                                       |
|                       from Source Site : DC1          |
|                         to Target Site : DC2          |
|-------------------------------------------------------|
*nn Please Confirm Run of ALLSITEn (Yes/No):


GDDO095I GDDR Unplanned Event ===>  600
Autoswap detected
>
>    Identify cause of Autoswap and
>    take appropriate action
>
600 GDDO096A 49 GDDO096A Reply to unplanned AutoSwap warning
R 49,Y
```

Reply **Yes** to allow GDDR to reconfigure your SRDF/S and SRDF/A replication following the swap event. Reply **No** to terminate the script.

## GDDR AutoSwap and restart of Enhanced Catalog Sharing

Following an AutoSwap, if you are using the z/OS Enhanced Catalog Sharing (ECS) feature, you may experience multiple occurrences of message IEC378I issued for each ECS catalog on each system, indicating that the catalog has been removed from ECS.

You must restart ECS for the systems in each sysplex. Potentially the easiest way to restart ECS is to enable systems automation based on message ID IEC378I and perform the restart on each system. You are strongly encouraged to automate the restart of ECS after AutoSwap events (planned and unplanned). You can employ the system command ROUTE to send the restart command to all systems in a sysplex. Note also the potential for redundant messages from each system and each catalog using ECS.

```
RO *ALL,F CATALOG, ECSHR(ENABLEALL)
```

# Local and regional disaster

GDDR declares a local disaster (LDR) when the *no paths or intervention required* AutoSwap (CAX) condition occurs concurrently with a heartbeat timeout (HBM).

A local disaster may occur as:

- The instantaneous loss of DC1 (primary site).
- A rolling disaster that eventually leads to the loss of DC1.

In either case, a local disaster is detected and reported by the C-system running at the secondary site.

A regional disaster (RDR) can be either:

- The simultaneous loss of both the primary and secondary sites.
- A rolling disaster that eventually leads to the loss of both the primary and secondary sites.

In either case, a regional disaster is detected and reported by the C-system running at DC3 or DC4.

## Confirm loss of primary region

When an C-system at the secondary region detects the loss of both DC1 and DC2, the following panel is displayed. You are requested to confirm the loss of the primary region.

IMPORTANT

Reply only *after* you have management approval.

```
        * * * R D R  Detected  * * *
        * * * R D R  Detected  * * *

     Please Confirm Loss of the primary region

     Seek Management Approval Before Replying
     Seek Management Approval Before Replying

*nn Please Confirm Management Approval (Yes/No):
```

A reply of **No** terminates the dialog. A reply of **Yes** displays a confirmation message on the system console (and in the SYSLOG). After you have confirmed the regional disaster (RDR), GDDR takes the following steps:

- Make the C-system at DC3 the master C-system.
- Attempt to RESET_CLEAR all z/OS systems at both DC1 and DC2.

## Confirm ready for recovery at secondary region

Once GDDR has finished these housekeeping activities, the following operator prompt is displayed:

```
        * * * R D R  Recovery * * *
        * * * R D R  Recovery * * *

     GDDR Now Ready to Begin Recovery @ DC3

     Seek Management Approval Before Replying
     Seek Management Approval Before Replying

*nn Please Confirm Management Approval (Yes/No):
```

At this point, management must decide which regional disaster recovery option is to be used. You must reply only *after* you have gained management approval.

A reply of **No** terminates the dialog. A reply of **Yes** displays a confirmation message on the system console (and in the SYSLOG).

You must now select **Option S: Select Script to Run** in the Primary Options Menu panel and initiate the chosen DC3 recovery option:

- Restart production at DC3/DC4 after site swap(GDD2P18A)
- Recover at DC4 after RDR in primary region (GDDRPA05)
- Recover at DC3 with SRDF/S to DC4 (GDDRPA05)
- Recover at DC4 with SRDF/S to DC3 (GDDRPA05)
- Restart production at DC3 with cascaded SRDF (GDDRPA06)
- Restart production at DC4 with cascaded SRDF (GDDRPA06)
- Restart production at DC3 with concurrent SRDF (GDDRPA06)
- Restart production at DC4 with concurrent SRDF (GDDRPA06)

**Note:** "RDR scripts" on page 552 provides more information about these scripts.

-

# Loss of AutoSwap owner

One C-system is designated as the owner of the consistency group/AutoSwap group. In AutoSwap protected configurations, this should always be the C-system at the secondary DASD site.

For example, if the primary DASD site is DC1, then the consistency group/AutoSwap group owner is the C-system at DC2.

**Note:** "Validate active AutoSwap group" on page 522 provides more information.

## Lost owner policy

If, during a planned or unplanned AutoSwap event, the consistency group/AutoSwap group owning system fails or hangs, the AutoSwap running on the remaining systems instigates the *lost owner policy*. The lost owner policy is a specification of the actions that must be taken when the owning system is no longer communicating.

## Lost owner policy operations

When a lost owner condition is detected, the first indication you see is the following message. This message is issued on all remaining live systems:

```
*CGRS483S (00008) AutoSwap group SITEA has lost access to owner host
 ssss (0123412340123456). Lost owner policy instigated.
```

In the message, *ssss* is the z/OS system name of the owner that has failed or hung.

If the lost owner policy AutoSwap parameter is set to "Operator," the following message is then issued on all remaining live systems:

```
*nn CGRS485A (00008) Reply HOLDIO, BACKOUT, SYSRESET, TAKEOVERasowner.
```

GDDR on the remaining C-systems will reply **TAKEOVER** to this message.

After the reply has been accepted, the remaining outstanding instances of this message are removed from the console through a Delete Operator Message (DOM) operation. The swap continues and is controlled by the new owner.

IMPORTANT

*Never* reply to this message without the permission of your z/OS systems programming support group.

Both CGRS483S and CGRS485A are ConGroup messages. You can find more information about these messages in the *Mainframe Enablers Message Guide*.

# System failure

This section details how GDDR handles individual system failures and how you should respond to the GDDR prompts:

- C-system failure
- Managed system failure

## C-system failure

When GDDR detects that a C-system has failed at a particular site, the following WTOR message is displayed on the system console:

```
GDDR Unplanned Event ===>
 >    GDDR Detects C-System SYS1 has Failed at Site DC1,
 >    You have the following choices... >
 >    Reply IPL:SYS1 - GDDR to restart SYS5 at Current Location DC1
 >              Ignore  - GDDR to Do Nothing.
 >
*nn Enter desired Option...IPL:SYS1 or I:
```

This condition could be because of some type of disaster at DC1, but more likely, is caused by some type of network problem that has caused a C-system heartbeat timeout which has resulted in GDDR declaring the C-system at DC1 "dead". Therefore, before replying, you must confirm the status of the C-system that has been reported as failed.

- GDDR to restart *ssss* at Current Location DC*n*

  To have GDDR restart the failed C-system (at its present location), reply:

  ```
  R nn,IPL:ssss
  ```

  Where:

  - *nn* is the WTOR ID number.
  - *ssss* is the z/OS system name of the failed C-system.

  GDDR performs a LOAD CLEAR on the failed system. You are prompted to confirm the load address and load parameters.

- GDDR to Do Nothing

  To have GDDR take no further action for the failed C-system, reply:

  ```
  R nn,I
  ```

  Where *nn* is the WTOR ID number.

## Managed system failure

GDDR declares a managed system failure when one of the messages listed below occurs on any managed system and both of the following conditions are true for the managed system:

- GDDRMAIN is found not active on the GDDR-managed system.

- An HMC query either is unable to determine system status or returns a status different from "Operating".
- SCF0646W CSC (*ccccccc-ccccc*) HOST *system-name* (*hhhhhhhhhhhhhhhh*) REMOVED, MISSING HEART BEAT FOR *ssssssss* SECONDS
- SCF0696W CSC (*ccccccc-ccccc*) HOST *system-name* (*hhhhhhhhhhhhhhhh*) HAS BEEN UNREGISTERED BY HOST *system-name* (*hhhhhhhhhhhhhhhh*)

The following WTOR message is displayed on the system console:

```
GDDR Unplanned Event ===>
 >
 >    GDDR Detects Production System SYS2 has Failed at Site DC1,
 >    You have the following choices...
 >
 >    Reply IPL:SYS2 - IPL the system in the LPAR in which it was running.
 >          SYSRESET — RESET CLEAR the system and leave it down.
 >          SYSRECOV — Recover the system on its recovery LPAR.
 >          SYSSITEn — IPL the contingency system for the failed system.
 >                 I — Ignore the event and take no action.
 >
*nn Enter desired Option...IPL:SYS2, SYSRESET,SYSRECOV, SYSSITEn or I:
```

IMPORTANT

Before making any reply, confirm the status of the managed system that has been reported as failed.
z/OS SFM sysplex timer failures have the potential for expressing themselves to GDDR as system failures. You should review all z/OS systems consoles including the HMC system consoles for STP failure WTORs.

- `IPL the system in the LPAR in which it was running`

  To have GDDR restart the failed system (at its present location), reply:

  `R nn,IPL:ssss`

  Where:

  - *nn* is the WTOR ID number.
  - *ssss* is the z/OS system name of the failed system.

  GDDR performs a LOAD CLEAR on the failed system. You are prompted to confirm the load address and load parameters.

- `RESET CLEAR the system and leave it down`

  To have GDDR do a system reset for the failed managed system, reply:

  `R nn,SYSRESET`

  Where *nn* is the WTOR ID number.

  GDDR performs a RESET_CLEAR of the failed system. It performs no other actions.

- `IPL the contingency system for the failed system`

  To have GDDR restart the business applications at the opposite site to the failed production system, reply:

  `R nn,SYSSITEn`

  Where *nn* is the WTOR ID number.

  GDDR performs a RESET_CLEAR of the failed managed system, then triggers the restart of the business applications on the system that is the contingency partner for the failed managed system.

- `Recover the system on its recovery LPAR`

  To have GDDR recover the failed system in it's recovery LPAR, reply:

  `R nn,SYSRECOV`

  Where *nn* is the WTOR ID number.

- `Ignore the event and take no action`

  To have GDDR take no further action for the failed managed system, reply:

  `R nn,I`

  Where *nn* is the WTOR ID number.

# GDDR master function transfer

If the GDDR Heartbeat Monitor detects that the C-system that owns the master function is no longer updating its heartbeat and has been declared dead, GDDR transfers master function ownership to another C-system.

GDDR issues the following message to the system console requesting confirmation of GDDR master function ownership transfer:

```
GDDR Confirmation ===>

      Please Confirm C-System Master Transfer

                From System : SYS3
                  to System : SYS5

                   Reason: CSYSDEAD



*nn Please Confirm Management Approval (Yes/No):
```

IMPORTANT

Before replying, check with your z/OS systems programming support group to confirm the correct reply.

- GDDR to transfer master function ownership

  To have GDDR complete the transfer of master function ownership, reply:

  `R nn,yes`

  Where *nn* is the WTOR ID number.

- GDDR to take no action

  To prevent GDDR from transferring the master function ownership, reply:

  `R nn,no`

  Where *nn* is the WTOR ID number.

# CHAPTER 10
# Maintaining GDDR Environment

This chapter includes the following topics:


- Updating GDDR licenses ..... 572
- Updating GDDRPARM file statements ..... 573
- Setting up a new C-system ..... 575
- Renaming an existing C-system ..... 576
- Changing C-system or managed system IP address ..... 577
- Changing C-system or managed system IP port ..... 578
- Adding a new system or sysplex to GDDR ..... 579
- Changing the consistency group name ..... 580
- Adding new SRDF groups to GDDR ..... 581
- Adding new devices to GDDR ..... 583
- Removing an SRDF group from GDDR control ..... 585
- Removing devices from GDDR control ..... 586
- Removing a system or a sysplex from GDDR ..... 587
- Changing global variable DIV dataset or WORKER parameters ..... 588
- Increasing GDDR DIV size ..... 589
- Restoring normal operations after MSC THAW ..... 590
- Handling special types of datasets ..... 591

# Updating GDDR licenses

To update GDDR license information:

1. Update the License Feature Codes (LFCs) in the SCF initialization (SCFINI) file.

   **Note:** The *ResourcePak Base for z/OS Product Guide* discusses the SCFINI file.

2. Issue the F *emcscf*,INI REFRESH command.

   **Note:** The *ResourcePak Base for z/OS Product Guide* describes the INI REFRESH command.

3. Recycle GDDRMAIN, or issue the PARM_REFRESH command described in "PARM_REFRESH" on page 218.

   **Note:** It is recommended to recycle GDDRMAIN for any license changes.

# Updating GDDRPARM file statements

After you edit the statements in the GDDRPARM file, take appropriate action as described in Table 46 on page 573.

**Note:** "Customize GDDRMAIN parameters" on page 145 provides information about the GDDRPARM file. "GDDRPARM statements" on page 256 lists GDDRPARM statements.

**Table 46** Actions to refresh GDDRPARM settings

| Updated statements | Action |
|---|---|
| MSCGROUP<br>MULTI<br>SITE | 1. Issue the PARM_REFRESH command described in "PARM_REFRESH" on page 218.<br>2. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |
| CMDQMAXT<br>SYMM<br>WORKER | 1. Issue the F GDDRMAIN,RESTART WORKMGR command to restart the work manager (WORKMGR) subtask.<br>2. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |
| COMM<br>DRTCOMM | 1. Issue the F GDDRMAIN,RESTART COMM command to restart the communication (COMM) subtask.<br>2. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |
| CPC<br>VCPC | 1. Issue the F GDDRMAIN,RESTART MISC command to restart the miscellaneous (MISC) subtask.<br>2. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |

**Table 46** Actions to refresh GDDRPARM settings

| Updated statements | Action |
|---|---|
| CSYSSITE | 1. Issue the F GDDRMAIN,RESTART COMM command to restart the communication (COMM) subtask.<br>2. Issue the F GDDRMAIN,RESTART MISC command to restart the miscellaneous (MISC) subtask.<br>3. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |
| GVDIVDSN | 1. Issue the F GDDRMAIN,RESTART GVT command to restart the global variable manager (GVT) subtask.<br>2. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |
| MSG | 1. Issue the F GDDRMAIN,RESTART MCSOPER command to restart the MCSOPER console subtask.<br>2. Issue the MPARM,CHECK command described in "MPARM" on page 216 in order to force resynchronization of GDDRPARM checksums across all systems in the GDDR-plex and prevent a possible degraded mode condition.<br>Alternatively, you can recycle GDDRMAIN. |

## Changing SYMM parameters

When adding or removing SRDF groups or SRDF devices from GDDR, it is possible that changes may be required to the SYMM parameters in the GDDRPARM file. This will be the case if you add or remove storage systems, or if the changed configuration requires less or more gatekeeper devices. These changes, if necessary, must be done before performing the procedures described in Chapter 10, "Maintaining GDDR Environment."

To change SYMM parameters:

1. Update the relevant SYMM parameters locally, and perform a RESTART WORKMGR. If no error messages occur, propagate the updated GDDRPARM file to all systems in the GDDR-plex.
2. Restart the COMM subtask of GDDRMAIN on all GDDR systems, even though the COMM parameters have not changed. If you do not do this, you will receive message GDDM144W informing you that the dataset and the in-use values are possibly different. You can accomplish this by using the BC command of GDDRMAIN if you wish.
3. Perform a RESTART WORKMGR on all systems in the GDDRPLEX. You can use the BC command for this purpose.

# Setting up a new C-system

Complete the following steps to set up a new C-system:

1. Ensure that correct system software is installed.
   a. Ensure that the system meets the requirements for running GDDR described in "Hardware and software requirements" on page 96.
   b. Ensure that ResourcePak Base, Consistency Groups, SRDF Host Component, and (optionally) TimeFinder are installed at required release and maintenance levels.
2. Follow the procedures relevant to C-systems in "Introduction" on page 110, and particularly those described in "Customize GDDRMAIN parameters" on page 145. Confirm that the new C-system is communicating with the other systems in the GDDR-plex using the GDDRMAIN MPARM command described in "MPARM" on page 216.
3. Update GDDRPARM C-system CPC parameters described in "CPC" on page 261.
4. Verify that parameters describing the new C-system are populated correctly in the Define C-Systems panel (M,P,C,C).
5. Update SCF (Symmetrix Control Facility) and Consistency Groups started task names for the specified C-system name using the Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E).
6. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency : RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Renaming an existing C-system

Complete the following steps to rename an existing C-system:

1. Replace the old C-system system name with the new C-system system name using the procedures described in "Customize GDDRMAIN parameters" on page 145.
2. Update the GDDRPARM CPC parameters described in "CPC" on page 261.
3. Confirm that the new C-system is communicating with the other systems in the GDDR-plex using the GDDRMAIN MPARM command described in "MPARM" on page 216.
4. Confirm that the SMFID, IPL Parameters, CPC and LPAR name values are populated as needed using the Define C-Systems panel (M,P,C,C).
5. Replace the existing C-system system name for the SCF (Symmetrix Control Facility) and Consistency Groups started task names using the Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E).
6. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency : RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Changing C-system or managed system IP address

To change the IP address of one or more systems on which GDDRMAIN runs, update the GDDRPARM file in synchronization with the IP address change. Complete the following steps:

1. Stop the COMM subtask of GDDRMAIN on all systems which are having an IP address change.
2. Edit the GDDRPARM file to reflect the new IP addresses.
3. When the new IP addresses are in use, start the COMM subtask on one of the changed systems.

   **Note:** This will result in message GDDM103W being issued from systems on which the COMM subtask was not stopped; ignore these messages.

4. Verify that the new GDDRPARM file does not cause any parameter initialization errors. If it does, stop the COMM subtask, correct the problem, and repeat step 3.
5. Propagate the new GDDRPARM file to all C-systems and managed systems and ensure it is consistent on all systems, as discussed in "Install GDDRPARM file" on page 145 and "Verify GDDRPARM file consistency" on page 148.
6. Start the COMM subtask on systems where it is not running; restart the COMM subtask on systems where it was left running. Verify that there are no parameter error messages on any system. If there are, correct them and go back to step 3.
7. Verify connectivity and consistency for all systems as described in "Install GDDRPARM file" on page 145 and "Verify GDDRPARM file consistency" on page 148.

# Changing C-system or managed system IP port

This procedure is similar to an IP address change, except that the change will affect all copies of GDDRMAIN on all GDDR systems because the port number must be the same for all systems. Complete the following steps:

1. Edit the GDDRPARM file, changing the port number on all COMM statements.
2. Propagate the modified GDDRPARM file to all GDDR systems.
3. Issue the GDDRMAIN MPARM command described in "MPARM" on page 216 on any GDDR system and verify that the value shown for the dataset is the same for every system. If it is not, propagate the new GDDRPARM file to all systems where a difference is shown.
4. Restart the COMM subtask of GDDRMAIN on all GDDR systems.

   When COMM restarts, GDDRMAIN waits one minute before broadcasting its in-use GDDRPARM data.

   - If you restart the COMM subtask on all GDDRMAIN copies within a minute, the process should complete successfully.
   - If more than one minute elapses between the first COMM restart and the last, you may receive GDDM103W messages (which can be ignored) and GDDM141E messages with resultant setting of Degraded mode. This should resolve itself when all COMM subtasks have been restarted. Alternatively, you can stop COMM on each system and then start COMM on each system after all are down. You can restart COMM on all systems by using the BC command of GDDRMAIN.
5. Verify connectivity and consistency for all systems as described in "Install GDDRPARM file" on page 145 and "Verify GDDRPARM file consistency" on page 148.

# Adding a new system or sysplex to GDDR

Use the following procedure when adding a new system or sysplex to the Enterprise consistency group, and thereby placing them under the management of GDDR.

Adding a new system to the Enterprise consistency group might involve the addition of new DASD. If this is the case, after this step is complete, continue with the procedure described in "Adding new SRDF groups to GDDR" on page 581.

1. Ensure that ResourcePak Base isand ConGroup areset up using common parameter definitions already used by the other systems in the Enterprise consistency group.
2. Edit the GDDRPARM file accessible by the new system by adding a COMM statement for the new system.
3. Update the GDDRPARM CPC parameters.
4. Start GDDRMAIN on the new system and verify that no parameter initialization error messages occur. Note that GDDM103W messages will occur on other systems; these can be ignored.
5. Once the new GDDRPARM file is satisfactory, propagate it to all GDDR systems.
6. Issue the GDDRMAIN MPARM command described in "MPARM" on page 216 on any GDDR system and verify that the value shown for the dataset is the same for every system. If it is not, propagate the new GDDRPARM file to all systems where a difference is shown.
7. Restart the COMM subtask on all GDDR systems except the new system.

   When COMM restarts, GDDRMAIN waits one minute before broadcasting its in-use GDDRPARM data.

   - If you restart the COMM subtask on all GDDRMAIN copies within a minute, the process should complete successfully.
   - If more than one minute elapses between the first COMM restart and the last, you will probably receive GDDM103W messages (which can be ignored) and GDDM141E messages with resultant setting of Degraded mode. This should resolve itself when all COMM subtasks have been restarted. Alternatively, you can stop COMM on each system and then start COMM on each system after all are down. You can restart COMM on all systems by using the BC command of GDDRMAIN.
8. Verify connectivity and consistency for all systems as described in "Verify GDDRPARM file consistency" on page 148.
9. Verify that the parameters describing the new system are populated correctly in the Define Host Objects panel (M,P,H).
10. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

    Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency : RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Changing the consistency group name

To change the name of the consistency group, perform the following steps:

1. Replace the old consistency group name in the Define Site roles and groups panel (M,P,C,R).
2. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

   ```
   Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
   Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
   Propagate to Other C-systems :  YES  (YES/NO/TRY)
   Clear the GDDR Command Queue ?  YES  (YES/NO)
   Enforce consistency : RETRY=5 (YES/NO/RETRY(1-5)
   Ignore Backup Failure :  NO  (YES/NO)
   ```

3. Load the updated parameters, following the procedure described in "Update GDDR ISPF profile (P)" on page 284.
4. Update parameters on all systems to reflect the new name.
5. Update RDF Manager parameter members listed in "Create SRDF parameter members (SITxxxxx)" on page 119 to reflect the new name.
6. Update the RDFPARM member associated with the RDF entry for the affected C-systems in the Define Dell EMC Mainframe Enabler STCs panel (M,P,H,E) with the new name.

# Adding new SRDF groups to GDDR

This section describes how to add new DASD to an existing environment by adding one or more new SRDF groups to GDDR and including the new devices in the new SRDF groups.

When adding new SRDF groups to GDDR, it is possible that changes are required to the SYMM parameters in the GDDRPARM file. This will be the case if you add new storage systems, or if the expanded configuration requires additional gatekeeper devices. These changes must be done before performing the procedure described below. See "Restoring normal operations after MSC THAW" on page 590 for more information on how to change SYMM parameters in the GDDRPARM file.

1. Ensure the following:
   - The SRDF groups have been defined.

   **Note:** The procedure for defining dynamic SRDF groups and creating device pairs is provided in the *SRDF Host Component for z/OS Product Guide*.

2. Create new GNS groups by specifying the new SRDF group with the EMCGROUP utility, or add the new SRDF group to an existing enterprise GNS group.

   **Note:** See *ResourcePak Base for z/OS Product Guide* for the procedure.

   The GNS group(s) defined can be named according to the following convention:

   GDDR*n*_*ddd*_*xxxxx*_RGRP_*nn*_*xx*

   where:

   | | |
   |---|---|
   | *n* | has the value 1 or 2, used to identify which group to use depending upon the location of the primary DASD. |
   | *ddd* | has the value CKD or FBA, depending upon what type of DASD is defined in the GNS group. |
   | *xxxxx* | is the last five (5) digits of the PowerMax/VMAX system serial number. |
   | *nn* | is the new SRDF group. |

   For each new SRDF group, two GNS groups must be defined (per DC).

3. Update GDDR parameters:
   a. If any of the standard devices in the SRDF groups being added are to have an associated BCV, add the necessary device ranges using the Define TimeFinder Device Ranges panel (M,P,D,T).
   b. Run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 to discover the updated SRDF device configuration and automatically create RDF.DEVICES parameters.
4. Distribute the changed Dell EMC parameter members to all systems participating in the environment.

5. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency :  RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Adding new devices to GDDR

This section describes how to add new DASD to an existing GDDR environment by adding the devices to one or more existing SRDF groups.

When adding new devices to GDDR, it is possible that changes are required to the SYMM parameters in the GDDRPARM file. This will be the case if the expanded configuration requires additional gatekeeper devices. These changes must be done before performing the procedure described below. See "Restoring normal operations after MSC THAW" on page 590 for more information on how to change SYMM parameters in the GDDRPARM file.

1. Stop the GDDR Event Monitor and GDDR Heartbeat Monitor.
2. Stop SRDF/SQAR with AutoSwap replication by entering the MSC,REFRESH command.

   **Note:** The *ResourcePak Base for z/OS Product Guide* describes the MSC,REFRESH command.

3. Create SRDF/S and SRDF/A device pairs.

   Add the new devices to one or more existing SRDF groups using the #SC VOL CREATEPAIR command described in the *SRDF Host Component for z/OS Product Guide*.

4. Update GDDR parameters:
   - If any of the standard devices being added are to have an associated BCV, create appropriate BCV device ranges using the Define TimeFinder Device Ranges panel (M,P,D,T).
   - Run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 to discover the updated SRDF device configuration and automatically create RDF.DEVICES parameters.
5. Add the devices to the existing GDDR-protected GNS groups.
6. Remove or update any SCF.DEV.EXCLUDE.LIST SCF initialization parameters which would exclude the devices you wish to add.

   **Note:** The *ResourcePak Base for z/OS Product Guide* describes the SCF.DEV.EXCLUDE.LIST parameter.

7. Issue an SCF,GNS,REFRESH command.

   **Note:** The *ResourcePak Base for z/OS Product Guide* describes the GNS,REFRESH command.

8. Issue a ConGroup REFRESH,FORCE command.

   **Note:** The *Consistency Groups for z/OS Product Guide* describes the REFRESH command.

9. Restart SRDF/A and SRDF/SQAR with AutoSwap.

10. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

    Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
```

# Removing an SRDF group from GDDR control

When removing SRDF groups from GDDR, it is possible that changes are required to the SYMM parameters in the GDDRPARM file. This will be the case if you remove storage systems, or if the reduced configuration requires less gatekeeper devices. These changes must be done before performing the procedure described below. See "Restoring normal operations after MSC THAW" on page 590 for more information on how to change SYMM parameters in the GDDRPARM file.

Complete the following steps to remove an SRDF group from GDDR control:

1. Verify these prerequisites:
   - Ensure that the SRDF groups to be removed from the control of GDDR have been removed from any relevant MSC GNS group.
   - If the SRDF groups are for SRDF/S devices, ensure that the SCFG(...) parameters referencing the SRDF groups being removed from GDDR control have been removed from the CGRP*xxx* Consistency Group parameter member on all production systems and C-systems.
   - Ensure that references to the SRDF groups being removed from the MSC group definitions in the MSC parameter members have been removed.
2. Update GDDR parameters:
   - Delete all BCV devices associated with the standard devices being removed using the Define TimeFinder Device Ranges panel (M,P,D,T).
   - Run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 to discover the updated SRDF device configuration and automatically create RDF.DEVICES parameters.
3. Refresh the SRDF/SQAR environment.
4. Restart SRDF/SQAR protection.

   Once successfully restarted, check to ensure that the SRDF group(s) being deleted are no longer part of the groups.
5. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency :  RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Removing devices from GDDR control

When removing devices from GDDR, it is possible that changes are required to the SYMM parameters in the GDDRPARM file. This will be the case if the reduced configuration requires less gatekeeper devices. These changes must be done before performing the procedure described below. See "Restoring normal operations after MSC THAW" on page 590 for more information on how to change SYMM parameters in the GDDRPARM file.

**Note:** Ensure that the devices to be removed from the control of GDDR are not gatekeeper devices.

1. Stop SRDF/SQAR by entering the MSC,PENDDROP command.

   **Note:** The *ResourcePak Base for z/OS Product Guide* describes the MSC,PENDDROP command.

2. Delete the SRDF/A relationships for the devices being removed.
3. Update your GNS group definitions so they no longer include the devices being removed from GDDR control.
4. Issue a GNS,REFRESH command.

   **Note:** The *ResourcePak Base for z/OS Product Guide* describes the GNS,REFRESH command.

5. Issue a ConGroup REFRESH,FORCE command.

   **Note:** The *Consistency Groups for z/OS Product Guide* describes the REFRESH command.

6. Restart SRDF/SQAR.
7. Update GDDR parameters:
   - If any of the standard devices being removed have an associated BCV, remove the associated BCV devices or device ranges using the Define TimeFinder Device Ranges panel (M,P,D,T).
   - Run the GDDRACDD utility described in "GDDR Automated Configuration Discovery for DASD (GDDRACDD)" on page 412 to discover the updated SRDF device configuration and automatically create RDF.DEVICES parameters.
8. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency :  RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Removing a system or a sysplex from GDDR

**Note:** On the system or systems being removed, ensure that the ResourcePak Basestarted procedures have been stopped and will no longer be used.

1. Edit the GDDRPARM file, which is accessible by any remaining system, and remove the COMM statement for the system being removed.
2. Update the GDDRPARM CPC parameters described in "CPC" on page 261.
3. Start GDDRMAIN on any remaining system and verify that no parameter initialization error messages occur. Note that GDDM103W messages will occur on other systems; these can be ignored.
4. When the new GDDRPARM file is satisfactory, propagate it to all C-systems and managed systems.
5. Issue the GDDRMAIN MPARM command described in "MPARM" on page 216 on any GDDR system and verify that the value shown for dataset is the same for every system. If it is not, propagate the new GDDRPARM file to all systems where a difference is shown.
6. Restart the COMM subtask on all managed systems.
   - If you restart the COMM subtask on all GDDRMAIN copies within a minute, the process should complete successfully.
   - If more than one minute elapses between the first COMM restart and the last, you may receive GDDM103W messages (which can be ignored) and GDDM141E messages with resultant setting of Degraded mode. This should resolve itself when all COMM subtasks have been restarted. Alternatively, you can stop COMM on each system and then start COMM on each system after all are down. You can restart the COMM subtask on all systems by using the GDDRMAIN BC command described in "BC and BR" on page 179.
7. Verify connectivity and consistency for all systems as described in steps 6-8 of the "Install GDDRPARM file" on page 145 procedure.
8. Delete the remaining parameters describing the system(s) being removed using the Parameter Management Options Menu panel (M,P).
9. During an appropriate maintenance window, validate and activate the parameter changes using the procedures described in "Validate GDDR parameter set (M,P,V)" on page 352 and "Activate GDDR parameter set (M,P,A)" on page 354.

   Specify the following validation and activation parameters:

```
Specify GDDR Parameter Load Type :  FULL   (PARTIAL,FULL)
Specify GDDR State Variables Action :  RESET  (RESET, ASIS, NOUPDATE)
Propagate to Other C-systems :  YES  (YES/NO/TRY)
Clear the GDDR Command Queue ?  YES  (YES/NO)
Enforce consistency : RETRY=5 (YES/NO/RETRY(1-5)
Ignore Backup Failure :  NO  (YES/NO)
```

# Changing global variable DIV dataset or WORKER parameters

To change the global variable DIV dataset or WORKER parameters, complete the following steps:

1. Edit the GDDRPARM file and make the needed changes on one C-system.
2. Propagate the modified GDDRPARM file to all C-systems and managed systems and ensure it is consistent, as described in "Install GDDRPARM file" on page 145 and "Verify GDDRPARM file consistency" on page 148.
3. Issue the GDDRMAIN MPARM command described in "MPARM" on page 216 on any GDDR system and verify that the value shown for dataset is the same for every system. If it is not, propagate the new GDDRPARM file to all systems where a difference is shown.
4. Restart the COMM subtask of GDDRMAIN on all GDDR systems, even though the COMM parameters have not changed. If you do not do this, you will receive message GDDM144W informing you that the dataset and the in-use values are possibly different. You can accomplish this using the GDDRMAIN BC command described in "BC and BR" on page 179.
5. Restart the other GDDRMAIN subtasks related to the changed parameters on the systems affected by the changes (GVT for DIV dataset definitions, WORKMGR for WORKER parameters).

   You do not need to restart these tasks on systems unaffected by the changes. Note that you can use the GDDRMAIN BC command to do this.

# Increasing GDDR DIV size

Dell EMC recommends defining the DIV to GDDRMAIN using the GDDRPARM GDIVDSN parameter (described in "GVDIVDSN" on page 265). This allows you to increase the GDDR DIV size on the fly, using the following procedure:

1. Define a new DIV dataset with larger capacity.
2. Stop the GVT subtask of GDDRMAIN.
3. IDCAMS repro the existing DIV into the new one.
4. Modify the GDDRPARM file GDIVDSN parameter to point to the new DIV dataset.
5. Restart the GVT subtask.
6. Copy the updated GDDRPARM file to all GDDR systems (C-systems and P-systems).

**Note:** There is no need to interrupt any GDDR processing to do this.

If you have allocated the GDDR DIV dataset in the GDDRMAIN JCL (not recommended), use the following procedure to increase the size of the DIV dataset:

1. Stop the GVT subtask; but do not stop GDDRMAIN.
2. At your earliest convenience, stop all GDDR processing (EVM, HBM, scripts, online updates, and so forth).
3. Run the GDDRGVX utility with DSPSAVE.
4. Define a new DIV dataset with larger capacity.
5. IDCAMS repro the output dataset of GDDRGVX to the new dataset.
6. Do one of the following:
   - Either change the GDDRMAIN JCL to point to the new DIV dataset and restart GDDRMAIN, or
   - Stop GDDRMAIN, rename the old DIV dataset to something else, rename the new DIV dataset to the name of the old DIV, and restart GDDRMAIN.

# Restoring normal operations after MSC THAW

MSC THAW processing prevents possible loss of SRDF/A protection as a result of a ConGroup trip. See "Support for MSC THAW processing" on page 74 for more information about this feature.

**Note:** As a first step, it may be necessary to DROP SRDF/A on the B->D leg and/or run the M9 utility to bring all B->D SRDF/A groups in the same state.

- Resume SRDF/A on the B->D leg.
- Resume SRDF/S.
- Resume ConGroup.
- Disable MSC on the secondary MSC server.
- Run the MSC,REFRESH and GDDRMAIN PARM_REFRESH commands on the primary MSC server.

  **Note:** You can run script GDDRMMSC (Restart primary MSC server at DC*n*) to complete this step.

- Restart the secondary MSC server.

  **Note:** You can run script GDDRPA36 (Restart secondary MSC server at DC*n*) to complete this step.

# Handling special types of datasets

In an SQAR environment, special consideration must be given to devices that are not SRDF/A linked to DC3 or DC4, specifically volumes containing the following system data:

- Page datasets
- Non-LOGR coupled datasets

These datasets are not managed by the GDDR restart process. GDDR management of these volumes is limited to coupled dataset realignment during site swap scripts, ensuring the primary coupled dataset is on the primary DASD site, with an alternate on the secondary DASD site. These devices can be defined to GDDR as external devices. "GDDR support for external devices" on page 81 provides more information.

Use of the "Perform test IPL from BCVs at DC3 or DC4" scripts requires no manual actions. GDDR will control the BCV SPLIT operations for volumes outside of GDDR control.

## Page datasets

In configurations that include AutoSwap, page datasets must be on ConGroup-protected volumes, and these volumes are treated as high-priority swap volumes which are swapped ahead of the data volumes. No production datasets should be allocated on these volumes.

All volumes containing page datasets must be paired up (using an adaptive copy SRDF group defined specifically for this purpose) to appropriate secondary region volumes and allowed to synch up. Once in synch, the pairs must be suspended. This action must be carried out any time paging volumes are added or moved in the configuration.

Volumes used for page datasets *must* be dedicated volumes, that is, they must contain no other system or user data.

## Coupled datasets

All volumes containing coupled datasets must be paired up (using an adaptive copy SRDF group defined specifically for this purpose) to appropriate R2 volumes and allowed to synch up. Once in synch, the pairs must be suspended.

This action must be carried out any time coupled dataset volumes are added or moved in the configuration. Additionally, after any policy management the relevant coupled datasets must be resynchronized with the corresponding volume(s) at R2 to ensure the latest changes will be available in the event of a regional disaster.

Volumes used for coupled datasets *must* be dedicated volumes, that is, they must contain no other system or user data.

## Standalone dump considerations

Special considerations for SADMP IPL volumes and DASD DUMP volumes should be made for each site in a GDDR infrastructure. Many SADMP generation options and the definitions of the DASD DUMP volumes embed device addresses for use by SADMP

operations after the IPL of SADMP. It is very likely that these addresses have no meaning in a mirrored environment (SRDF/S or SRDF/A), resulting in unpredictable SADMP operations at the recovery sites.

In addition, the contents of the SADMP DASD DUMP volumes is of very little consequence at DC2, DC3, and DC4 and the mirroring only slows down SADMP's operation. Consequently, you should review your SADMP generation procedures and business continuance operations and consider maintaining separate SADMP IPL and DASD DUMP volumes at DC1, DC2, DC3, and DC4.

# CHAPTER 11
# Troubleshooting

This chapter includes the following topics:


- Detecting and resolving problems ..... 594
- Tracing and debugging ..... 594
- Using GDDRXCMD batch utility ..... 594
- Verifying maintenance level ..... 595

# Detecting and resolving problems

Keep the following points in mind:

- GDDR scripts should always end with return code=0. This indicates that the desired action has been completed successfully.
- If a GDDR script does not complete with return code=0, you need to identify the problem. First, examine the JOBLOG of the job that ran the script. Then, examine the SYSTSPRT and SYSPRINT DDs for any error messages and non-zero return codes.
- Look up GDDR error messages in the *GDDR Message Guide*. Follow the actions recommended in that document.

# Tracing and debugging

GDDRMAIN verbose messaging may be enabled using the VERBose EXEC parameter (see "VERBose" on page 172) or toggled on and off using the F GDDRMAIN,SET VERBose={ON|OFF} command (see "SET" on page 232). GDDRMAIN verbose messages are written to the console.

GDDRMAIN debugging may be enabled using the DeBuG EXEC parameter (see "DeBuG" on page 171) or toggled on and off using the F GDDRMAIN,SET DeBuG={ON|OFF} command (see "SET" on page 232). GDDRMAIN debugging is written to the console or the SYSPRINT DD.

GDDRMCMD debugging may be enabled in any address space using the DEBUG DD DUMMY statement. This applies to all commands issued in that address space, including those issued programmatically.

To help diagnose "GDDRSSVT not found" errors, the DBUGSSVL DD DUMMY statement may be used to enable GDDRSSVL debugging. Note that this generates large amount of output and should only be used under the direction of Dell EMC Customer Support.

GDDRMAIN tracing is enabled by default. It can be toggled on and off using the F GDDRMAIN,SET TRaCe={ON|OFF} command (see "SET" on page 232). It is recommended that tracing always be enabled in case of a serious issue where Dell EMC requires access to the trace in a dump. The GDDRMAIN trace may be formatted at any time using the GDDRTRCP job in SAMPLIB.

Debugging, tracing, and verbose messaging for GDDR REXX modules may be enabled using the option **D** in the Setup and Maintenance Menu panel (M), as described in "Set message, debug and trace options (M,D)" on page 358. The output goes to SYSTSPRT.

# Using GDDRXCMD batch utility

As part of troubleshooting or correcting GDDR issues, you may need to print or clear the GDDR command queue. To do so, use the GDDRXCMD batch utility described in "GDDR Command Queue utility (GDDRXCMD)" on page 514.

IMPORTANT

Only clear the queue if you have been advised to do so by Dell EMC Customer Support.

# Verifying maintenance level

The MAINTENANCE command described in “MAINTENANCE” on page 211 validates the GDDR maintenance level.

The MAINTENANCE command is issued automatically at GDDRMAIN startup, and the module-level maintenance report is written to the SYSPRINT DD in the GDDRMAIN job log. The RELOAD command also automatically issues the MAINTENANCE command if select maintenance is applied that way under the direction of Dell EMC Customer Support.

**Note:** GDDRMAIN must be recycled to apply regular maintenance.