DB2 Information Management Software

IBM

Disaster Recovery with DB2 UDB for z/OS

- Examine your choices for local or remote site recovery
- Recover your DB2 system to a point in time
- Adopt best practices for recovery execution

Paolo Bruni
Pierluigi Buratti
Florence Dubois
Judy Ruby-Brown
Christian Skalberg
Tsumugi Taira
Kyungsoon Um

Redbooks

ibm.com/redbooks

International Technical Support Organization

Disaster Recovery with DB2 UDB for z/OS

November 2004

SG24-6370-00

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

First Edition (November 2004)

This edition applies to Version 8 of IBM Database 2 Universal Database for z/OS (program number 5625-DB2).

Contents

Tables . . . xiii

Figures . . . xv

Examples . . . xix

Notices . . . xxv
Trademarks . . . xxvi

Preface . . . xxvii
The contents of this redbook . . . xxvii
The team that wrote this Redbook . . . xxviii
Become a published author . . . xxxi
Comments welcome . . . xxxi

Part 1. The whole picture . . . 1

Chapter 1. Business continuity . . . 3
1.1 Business Continuity definition . . . 4
1.1.1 Business Continuity and Disaster Recovery . . . 4
1.1.2 What is a disaster? . . . 5
1.1.3 How to protect from a disaster . . . 6
1.2 IBM Business Continuity Strategy . . . 6
1.2.1 Disaster Recovery solutions . . . 8
1.3 General considerations on disaster recovery . . . 9
1.3.1 The lessons from September 11 and the requirements evolution . . . 12
1.4 Business Continuity and Recovery Services . . . 13

Chapter 2. DB2 disaster recovery . . . 15
2.1 Introduction to DB2 disaster recovery solutions . . . 16
2.1.1 Conventional transport . . . 17
2.1.2 Remote Copy Services . . . 18
2.1.3 Remote Copy and DB2 . . . 21
2.1.4 Data replication . . . 21
2.2 Data consistency . . . 24
2.2.1 The rolling disaster . . . 24
2.2.2 Consistency groups . . . 25
2.3 DR solutions in terms of RTO and RPO . . . 27
2.4 DB2's disaster recovery functions . . . 28
2.4.1 Backup . . . 28
2.4.2 Restart . . . 30
2.4.3 Recover . . . 34
2.5 Determining the RBA for conditional restarts . . . 35
2.5.1 Choosing an RBA for normal conditional restart . . . 35
2.5.2 Choosing an RBA value for SYSPITR . . . 36
2.6 What if DB2 utilities were running? . . . 38
2.6.1 LOAD . . . 38
2.6.2 REORG . . . 39

Part 2. Disaster recovery major components . . . 41

Chapter 3. Traditional recovery . . . 43
3.1 Overview . . . 44
3.2 Planning for disaster recovery . . . 45
3.2.1 Dump and restore of DB2 environment . . . 45
3.2.2 Consistent point-in-time recovery . . . 46
3.2.3 Continuous archive log transfer . . . 46
3.3 Recovery procedures . . . 47
3.3.1 Establish the environment . . . 47
3.3.2 Recover the BSDS . . . 48
3.3.3 Restart DB2 (conditional restart) . . . 48
3.3.4 Recover DB2 system data . . . 49
3.3.5 Prepare subsystem for use . . . 49
3.4 Recovering the BSDS . . . 49
3.5 Restart DB2 (Conditional Restart) . . . 53
3.6 Considerations . . . 56

Chapter 4. DB2 Tracker . . . 57
4.1 Overview . . . 58
4.2 Tracker site recovery . . . 59
4.2.1 Setting up the tracker site . . . 59
4.2.2 Establishing recovery cycle at the tracker site . . . 60
4.2.3 How to take over in case of disaster . . . 63
4.3 Considerations . . . 64
4.3.1 Restrictions on tracker site . . . 64
4.3.2 Pros and cons . . . 65

Chapter 5. ESS FlashCopy . . . 67
5.1 Introducing FlashCopy . . . 68
5.2 How FlashCopy works . . . 68
5.3 DFSMSdss utility . . . 71
5.3.1 COPYVOLID parameter . . . 72
5.3.2 DUMPCONDITIONING parameter . . . 72
5.3.3 FCNOCOPY and FCWITHDRAW parameters . . . 74
5.4 Incremental FlashCopy . . . 75
5.5 FlashCopy Consistency Group . . . 77

Chapter 6. SMS copy pools and DB2 point in time recovery . . . 79
6.1 DFSMS 1.5 enhancements . . . 80
6.1.1 DFSMShsm Fast Replication . . . 80
6.1.2 DFSMS copy pools . . . 80
6.1.3 Copy pool backup storage group type . . . 81
6.2 Preparing for DFSMShsm Fast Replication . . . 82
6.3 DB2 enhancements . . . 83
6.3.1 Other changes to support system level point-in-time recovery . . . 84
6.4 DB2 PITR using Fast Replication . . . 86
6.4.1 DB2 BACKUP SYSTEM utility . . . 86
6.5 DB2 Recover using Fast Replication backups . . . 92
6.5.1 DB2 RESTORE SYSTEM utility . . . 92
6.5.2 Running RESTORE SYSTEM . . . 94

Chapter 7. Peer-to-Peer Remote Copy . . . 97
7.1 Peer-to-Peer Remote Copy (PPRC) . . . 98
7.1.1 How PPRC works . . . 98
7.1.2 DB2 and critical attributes . . . 100

7.1.3 Rolling disaster with PPRC alone . . . 101
7.1.4 Consistency grouping and messages issued . . . 101
7.2 PPRC Extended Distance (PPRC-XD) . . . 102
7.2.1 Characteristics . . . 103
7.2.2 Creating a Consistency Group with PPRC-XD . . . 105
7.3 Asynchronous PPRC: Global Mirror . . . 106
7.3.1 Prerequisites for Asynchronous PPRC . . . 107
7.3.2 Terminology . . . 108
7.3.3 Asynchronous PPRC: How it works . . . 109
7.3.4 Prerequisites for Asynchronous PPRC . . . 111
7.4 PPRC feature dependencies . . . 111

Chapter 8. eXtended Remote Copy . . . 113
8.1 Introduction . . . 114
8.1.1 XRC configurations . . . 115
8.1.2 XRC hardware requirements . . . 116
8.1.3 Dynamic workload balancing . . . 117
8.1.4 Planned outage support . . . 117
8.1.5 Unplanned outage support for ESS . . . 118
8.2 XRC components . . . 118
8.2.1 Primary storage subsystem . . . 118
8.2.2 Secondary storage subsystem . . . 119
8.2.3 System Data Mover . . . 119
8.2.4 Address spaces . . . 119
8.2.5 Journal, control, and state data sets . . . 120
8.2.6 An XRC session . . . 120
8.2.7 Utility devices . . . 121
8.2.8 An XRC storage control session . . . 121
8.3 XRC operation: data flow . . . 122
8.4 Details on Consistency Groups . . . 123
8.5 XRC considerations for DB2 disaster recovery . . . 127

Part 3. General solutions for disaster recovery . . . 129

Chapter 9. Split Mirror . . . 131
9.1 Split Mirror . . . 132
9.1.1 Normal operation . . . 132
9.1.2 Suspend PPRC links . . . 133
9.1.3 Perform backup to preserve the environment . . . 134
9.1.4 Re-establish the PPRC links . . . 135
9.1.5 Suspend pairs . . . 136
9.1.6 Resume logging . . . 136
9.1.7 Dump to safety volumes . . . 137
9.2 Considerations . . . 138

Chapter 10. FlashCopy Consistency Group . . . 139
10.1 Solution overview . . . 140
10.1.1 How it works . . . 140
10.2 Design the solution . . . 142
10.2.1 Define backup environment requirements . . . 142
10.2.2 Define recovery environment requirements . . . 144
10.2.3 Define testing environment requirements . . . 144
10.2.4 Define backup procedure . . . 144
10.2.5 Define off-site procedure and timing . . . 146

10.2.6 Define the restore procedure 146
10.2.7 Define automation requirements 146
10.3 Implement the solution 147
10.3.1 Create FlashCopy Consistency Group task 147
10.3.2 Create FlashCopy Consistency Group Created task 149
10.3.3 Create ICKDSF REFORMAT jobs 151
10.3.4 Create DFSMSdss DUMP jobs 152
10.3.5 Create ICKDSF INIT jobs 152
10.3.6 Create DFSMSdss RESTORE jobs 153
10.3.7 Create a Disaster Recovery Procedure (DRP) 153
10.4 Run the solution 155
10.4.1 Execute periodic disaster recovery testing 155
10.4.2 Maintain the solution 155
10.5 Additional considerations 156

Chapter 11. Global Copy PPRC-XD 157
11.1 General PPRC-XD Global Copy Configuration 158
11.2 Creating your Consistency Group 158
11.3 Applying additional log to a Consistency Group 161
11.3.1 How to verify a log 162
11.4 Considerations 162

Chapter 12. Global Mirror PPRC 165
12.1 Overview 166
12.1.1 Terminology 166
12.1.2 How it works 168
12.1.3 Restrictions 169
12.2 Design the solution 170
12.2.1 Define primary environment 170
12.2.2 Define recovery environment 171
12.2.3 Define testing environment requirements 172
12.2.4 Select the interface to PPRC commands 173
12.2.5 Define session management commands 174
12.3 Implement the solution 175
12.3.1 Establish Async-PPRC data paths 175
12.3.2 Establish PPRC paths to the remote site 176
12.3.3 Establish PPRC-XD pairs 176
12.3.4 Establish FlashCopy pairs at the remote site 178
12.3.5 Define the Asynchronous PPRC session 179
12.3.6 Add volumes to the session 180
12.3.7 Start the session 181
12.4 Run the solution 187
12.4.1 Add/remove volumes to/from the session 187
12.4.2 Modify the session 187
12.4.3 Pause/terminate the session 188
12.4.4 Test the solution 188
12.5 Monitor Asynchronous PPRC 189
12.5.1 Determine the status of Asynchronous PPRC 189
12.5.2 Query the status of the volumes in the session 191
12.5.3 Query the number of volumes that are out-of-sync 192
12.5.4 Query the status of the FlashCopy volumes after an outage 192
12.5.5 Asynchronous PPRC Session Manager 194
12.5.6 RMF enhancements: Performance statistics reports for ESS links 195

12.6 Recovery procedures . . . 196
12.6.1 Unplanned outage and switch to the remote site . . . 197
12.6.2 Planned outage . . . 198
12.6.3 Considerations . . . 199

Chapter 13. XRC: Global Mirror for z/OS . . . 201
13.1 Solution overview . . . 202
13.2 Design the solution . . . 203
13.2.1 Verify hardware requirements . . . 203
13.2.2 Verify software requirements . . . 204
13.2.3 Select volumes for remote copy . . . 204
13.2.4 The ERRORLEVEL parameter . . . 206
13.2.5 Verify SDM requirements . . . 207
13.2.6 Verify the primary ESS subsystems . . . 209
13.2.7 Verify the secondary storage subsystem . . . 209
13.2.8 Determine SDM-primary site bandwidth . . . 210
13.2.9 Select Utility device mode . . . 212
13.2.10 Understand XADDPAIR processing . . . 212
13.3 Define XRC management . . . 212
13.3.1 Issuing XRC TSO commands . . . 213
13.3.2 XSTART command . . . 214
13.3.3 XADDPAIR command . . . 214
13.3.4 XQUERY command . . . 216
13.3.5 XDELPAIR command . . . 217
13.3.6 XEND command . . . 218
13.3.7 XSUSPEND command . . . 218
13.3.8 XSET command . . . 219
13.3.9 XRECOVER command . . . 219
13.3.10 XADVANCE command . . . 219
13.3.11 XCOUPLE command . . . 220
13.3.12 ANTRQST application programming interface . . . 220
13.4 Define XRC automation . . . 220
13.4.1 JCL-/REXX generation . . . 221
13.4.2 Error recovery . . . 221
13.4.3 GDPS/XRC . . . 221
13.5 Implement the solution . . . 221
13.5.1 Allocate the Journal data set . . . 221
13.5.2 Allocate the control data set . . . 223
13.5.3 Allocate the state data set . . . 224
13.5.4 Allocate the master data set . . . 224
13.5.5 Implement XRC commands and procedures . . . 225
13.5.6 Implement XRC security . . . 226
13.5.7 XRC testing . . . 227
13.6 Run the solution . . . 228
13.6.1 HCD reconfiguration restriction . . . 229
13.6.2 XRC volume format, track, and access mode restrictions . . . 229
13.6.3 ICKDSF and XRC volumes . . . 229
13.6.4 XRC Performance monitor . . . 230
13.6.5 Reporting on the XRC environment . . . 230
13.6.6 XRC diagnostic aids . . . 230
13.7 Database recovery with XRC . . . 232
13.8 Using XRC with FlashCopy . . . 232

Chapter 14. Geographically Dispersed Parallel Sysplex 233
14.1 GDPS overview 234
14.1.1 GDPS/PPRC 235
14.1.2 GDPS/XRC 237
14.1.3 Functional highlights 239
14.1.4 IBM Global Services offerings 240
14.2 GDPS-Metro Solution 240
14.2.1 Initial setup 241
14.2.2 The Freeze Policy 241
14.2.3 After Failover to Recovery Site 244
14.2.4 Considerations about GDPS 245
14.3 Metro/Global Mirror solution 245
14.3.1 Scenario 1 246
14.3.2 Scenario 2 247

Part 4. Implementing disaster recovery scenarios 249

Chapter 15. Set Log Suspend - FlashCopy - More Log - Restore System Log Only 251
15.1 Description of the scenarios 252
15.1.1 FlashCopy with DB2 252
15.1.2 RESTORE SYSTEM LOGONLY 253
15.2 Recovery procedures 254
15.2.1 Prepare for disaster recovery 254
15.2.2 Set up for recovery site 254
15.3 Test environment 256
15.4 Preparing for disaster recovery 256
15.4.1 -SET LOG SUSPEND 257
15.4.2 FlashCopy 257
15.4.3 -SET LOG RESUME 258
15.4.4 Dump the FlashCopy to tape 258
15.4.5 More active logs 259
15.5 Setting up the recovery site 260
15.5.1 Restore the FlashCopy data sets to disk 260
15.5.2 Keep the current archive log 261
15.5.3 Recover the BSDS 262
15.5.4 DSNJU003 to create a SYSPITR CRCR 265
15.5.5 Restart DB2 with a SYSPITR CRCR 266
15.5.6 RESTORE SYSTEM LOGONLY 268
15.5.7 Recover all objects in RECP or RBDP status 270
15.6 Considerations 275
15.6.1 How to find the SYSPITR RBA on the last archive log 275
15.6.2 Considerations 279

Chapter 16. FlashCopy Consistency Group and restart 281
16.1 Overview 282
16.2 The procedure 284
16.2.1 Backup using FlashCopy Consistency Group 284
16.2.2 Dump the FlashCopy 284
16.2.3 Restore and restart at the recovery site 284
16.3 The test environment 285
16.4 Backup using FlashCopy Consistency Group 286
16.4.1 Define the FlashCopy Consistency Group Creation Task 286
16.4.2 Define the FlashCopy Consistency Group Created Task 290
16.4.3 Create consistent copy 291

16.5 Workload at local site . . . 292
16.6 Dumping the FlashCopy . . . 292
16.7 DB2 restart at recovery site . . . 294
16.7.1 Restore the backup of FlashCopy . . . 294
16.7.2 DB2 restart . . . 295
16.7.3 Recover all objects in the restricted status . . . 296
16.8 Considerations . . . 300

Chapter 17. PPRC - FlashCopy from secondary . . . 301
17.1 Overview of the test scenarios . . . 302
17.1.1 Testing environment . . . 303
17.2 Configuring the PPRC environment . . . 303
17.2.1 Tasks for PPRC testing . . . 304
17.2.2 Define task to Establish PPRC Path . . . 304
17.2.3 Define task to establish PPRC volume pair . . . 307
17.3 Scenario 1: Using FlashCopy Consistency Group . . . 310
17.3.1 Establish PPRC path . . . 310
17.3.2 Establish PPRC volume pair . . . 310
17.3.3 Start DB2 workload JCL . . . 312
17.3.4 FlashCopy with Consistency Group . . . 312
17.3.5 FlashCopy Consistency Group Created task . . . 313
17.3.6 Stop the DB2 subsystem . . . 313
17.3.7 Stop the PPRC pair . . . 314
17.3.8 Terminate PPRC path . . . 314
17.3.9 Vary offline the primary volumes . . . 314
17.3.10 Vary online the volumes at the recovery site . . . 314
17.3.11 Restart DB2 at recovery site . . . 315
17.3.12 Check objects in restricted status . . . 316
17.3.13 Procedures to recover objects from restricted status . . . 317
17.4 Scenario 2: Using DB2 Set Log Suspend . . . 319
17.4.1 Establish PPRC path . . . 320
17.4.2 Establish PPRC volume pair . . . 320
17.4.3 Start the DB2 workload . . . 320
17.4.4 DB2 Set Log Suspend . . . 320
17.4.5 FlashCopy from the secondary to the third volumes . . . 321
17.4.6 DB2 Set Log Resume . . . 321
17.4.7 Stop the DB2 subsystem . . . 322
17.4.8 Stop the PPRC pair . . . 322
17.4.9 Terminate the PPRC path . . . 322
17.4.10 Vary offline the primary volumes . . . 322
17.4.11 Vary online the recovery site volumes . . . 322
17.4.12 Restart DB2 at the recovery site . . . 322
17.4.13 Check the objects in restricted status . . . 323
17.4.14 Procedures to recover the objects in restricted status . . . 324
17.5 Scenario 3: Using PPRC Consistency Group . . . 325
17.5.1 Establish PPRC with Consistency Group . . . 325
17.5.2 Establish PPRC volume pair . . . 326
17.5.3 Start the DB2 workload . . . 326
17.5.4 PPRC Freeze task with Consistency Group . . . 326
17.5.5 FlashCopy from secondary to third volumes . . . 327
17.5.6 PPRC Consistency Group Created task . . . 328
17.5.7 Reestablish PPRC path . . . 328
17.5.8 Resynchronize volume pairs . . . 329

17.5.9 Stop the DB2 subsystem . . . 330
17.5.10 Stop the PPRC pair . . . 330
17.5.11 Terminate the PPRC path . . . 330
17.5.12 Vary offline volumes of the primary site . . . 330
17.5.13 Vary online volumes of recovery site . . . 330
17.5.14 Restart DB2 at the recovery site . . . 330
17.5.15 Check the DB2 objects in restricted status . . . 332
17.5.16 Procedures to recover objects in restricted status . . . 333
17.6 Considerations . . . 333

Chapter 18. XRC and restart . . . 335
18.1 Description of the test environment . . . 336
18.2 Preparing for using XRC . . . 337
18.2.1 Allocating the XRC data sets . . . 337
18.2.2 Establishing the XRC session environment . . . 339
18.3 Scenario 1: Simple DB2 restart . . . 342
18.3.1 Preparation . . . 342
18.3.2 Recovery on the secondary volumes . . . 343
18.3.3 Restart DB2 . . . 345
18.3.4 Summary . . . 347
18.4 Scenario 2: Disaster recovery . . . 347
18.4.1 Preparation . . . 347
18.4.2 Disaster . . . 348
18.4.3 Recovery on the secondary volumes . . . 348
18.4.4 Restart DB2 . . . 349
18.4.5 Summary . . . 355
18.5 Scenario 3: Using XRC to get PIT copies . . . 355
18.5.1 Preparation . . . 356
18.5.2 Suspend the XRC volumes pairs . . . 357
18.5.3 Resynchonize the XRC volume pairs . . . 358
18.5.4 Suspend the XRC volume pairs . . . 359
18.5.5 Recovery to the secondary volumes . . . 359
18.5.6 Restart DB2 . . . 360
18.5.7 Summary . . . 361

Chapter 19. Local recovery: System PITR . . . 363
19.1 The need for system PITR . . . 364
19.2 Description of the scenarios . . . 365
19.2.1 Non-data sharing . . . 365
19.2.2 Data sharing . . . 366
19.3 Description of the test environment . . . 368
19.4 SMS definitions . . . 368
19.5 Restoring to an arbitrary point-in-time . . . 369
19.5.1 Creating the backup . . . 370
19.5.2 Generating log activity . . . 372
19.5.3 Cleaning up before recovery . . . 373
19.5.4 Recovering the data to an arbitrary prior point-in-time . . . 374
19.6 Considerations . . . 379
19.6.1 What if you added volumes between backup time and restore time? . . . 379
19.6.2 Restart checkpoint not found . . . 382
19.6.3 Fast Log Apply during RESTORE . . . 383
19.6.4 Reducing objects recovery status . . . 386

Chapter 20. Restart using tape dump of copy pools . . . 389

20.1 Extending Fast Replication backups via tape dumps . . . 390
20.2 Description of the scenarios . . . 390
20.2.1 Non-data sharing . . . 390
20.2.2 Data sharing . . . 391
20.3 Description of the test environment. . . . 392
20.4 Setting up the SMS environment . . . 392
20.4.1 Preparing for using the Fast Replication function for DB2 . . . 393
20.4.2 Defining the copy pools for DB2 logs and data. . . . 394
20.4.3 Defining the copy pool backup storage group. . . . 395
20.4.4 Validating the Fast Replication environment. . . . 397
20.5 Creating the backup . . . 399
20.5.1 Using the BACKUP SYSTEM utility . . . 399
20.5.2 Without the BACKUP SYSTEM utility . . . 401
20.6 Dumping the backup . . . 404
20.6.1 What to dump? . . . 404
20.6.2 When to dump?. . . . 405
20.6.3 How to dump?. . . . 406
20.7 Restoring the dump at the recovery system . . . 407
20.8 Restarting DB2 . . . 407

Part 5. Additional considerations . . . 409

Chapter 21. Data sharing . . . 411
21.1 Data sharing overview. . . . 412
21.2 Data sharing and disaster recovery . . . 414
21.3 Planning for disaster recovery. . . . 415
21.3.1 Dump and restore of DB2 environment. . . . 416
21.3.2 Consistent Point-in-Time Recovery. . . . 416
21.3.3 Continuous archive log transfer . . . 417
21.4 ARCHIVE LOG command for disaster backup . . . 417
21.4.1 MODE(QUIESCE) versus SCOPE(GROUP) . . . 418
21.4.2 Advantage and disadvantage of MODE(QUIESCE) . . . 418
21.4.3 ARCHIVE LOG command verification. . . . 419
21.5 Recovery procedures in data sharing environment. . . . 419
21.5.1 Establish environment. . . . 420
21.5.2 Recover the BSDS . . . 420
21.5.3 Restart DB2 (Conditional Restart). . . . 421
21.5.4 Recover DB2 system data . . . 422
21.5.5 Prepare the subsystem for use . . . 422
21.6 Considerations . . . 423
21.6.1 Cleaning up coupling facility structures. . . . 423
21.6.2 Determine ENDLRSN for the conditional restart record . . . 423
21.6.3 Choosing an LRSN . . . 425
21.6.4 Restart versus Restart Light . . . 427
21.6.5 GRECP/LPL recovery recommendations . . . 428
21.6.6 CLOSE (YES) versus CLOSE (NO) data sets . . . 429

Chapter 22. Validation and performance. . . . 431
22.1 Health checking. . . . 432
22.1.1 What is health checking? . . . 432
22.1.2 Monitoring system availability . . . 432
22.1.3 Testing recovery procedures. . . . 436
22.1.4 Identifying single points of failure . . . 437
22.2 Mass recovery best practices . . . 438

22.2.1 Introduction 438
22.2.2 Recommendations for fast recovery 439
22.2.3 Special considerations for log-based recovery 445
22.2.4 Recommendations for log-based recovery 446
22.2.5 Tools for processing the DB2 log 447
22.3 Maintenance 448

Part 6. Appendixes 451

Appendix A. REXX procedures 453
A.1 CPBULIST: JCL to dump and restore backup disk volumes 454
A.2 BSDSSTAT: Log fill time statistics from BSDS 460

Appendix B. PITR definitions 463
B.1 Setting up copy pools for PITR 464
B.2 Sample scenario: Restoring to an arbitrary PITR 468

Appendix C. Additional material 471
Locating the Web material 471
Using the Web material 471
System requirements for downloading the Web material 471
How to use the Web material 472

Abbreviations and acronyms 473

Glossary 475

Related publications 483
IBM Redbooks 483
Other publications 483
Online resources 484
How to get IBM Redbooks 484
Help from IBM 484

Index 485

Tables

1-1 Disaster recovery solutions general comparison chart 8
2-1 DR solutions comparison table 27
6-1 The DB2 token 84
6-2 DB2 token contents breakout of Example 6-1 output listing 84
7-1 Minimum requirements for PPRC features 112
8-1 Address space descriptions 119
12-1 Web Interface and ICKDSF commands 173
12-2 Indication of revertible or non-revertible 193
13-1 XRC and FlashCopy feature codes 203
13-2 ERRORLEVEL and suspension of volume pairs 207
13-3 XRC TSO commands 213
13-4 Example of TSO command required to run an XRC solution 225
17-1 Tasks for PPRC testing 304
19-1 Restore with and without FLA 386
22-1 Current disaster recovery related APARs 448

Figures

1-1 Business Continuity and Disaster Recovery relationship. 4
1-2 IBM BCRS Methodology 7
2-1 Data loss (RPO) and recovery time (RTO) 16
2-2 PTAM configuration 17
2-3 Rolling disaster 25
2-4 How to determine the correct log and ENDRBA to use for Conditional Restart 35
3-1 Dump and restore of DB2 environment 44
3-2 A daily business approach. 45
4-1 Overview of Local and Tracker Site. 58
5-1 FlashCopy with background copy 69
5-2 DFSMSdss COPY with COPYVOLID 72
5-3 DFSMSdss COPY with DUMPCONDITIONING 73
5-4 Incremental FlashCopy 76
5-5 FlashCopy consistency group 78
6-1 Copy pool structure 82
6-2 BACKUP SYSTEM utility execution for DSNDB0G 87
6-3 Two BACKUP SYSTEM FULL, and one DATA ONLY are taken 88
6-4 Recovering to an arbitrary point-in-time using the RESTORE SYSTEM utility 94
7-1 Metro Mirror features 98
7-2 PPRC data flow 99
7-3 PPRC Extended Distance background 103
7-4 Characteristics of PPRC Extended Distance. 104
7-5 Global Mirror features 107
7-6 Asynchronous PPRC 109
7-7 Example of an Asynchronous PPRC configuration 110
8-1 XRC configuration 114
8-2 XRC basic configuration 115
8-3 Coupled Extended Remote Copy (CXRC). 116
8-4 XRC data flow 122
8-5 Creation of Consistency Group - CG1 124
8-6 Creation of Consistency Group - CG2. 126
9-1 Split Mirror - Logs Current - Data not current. 132
9-2 Suspend PPRC connections between the logs 133
9-3 Changing the environment. 134
9-4 Split MIrror resynchronize all volumes. 135
9-5 Split Mirror - 99% of tracks RESYNC and SET LOG SUSPEND. 135
9-6 Split Mirror - Suspend all PPRC pairs and Resume DB2 logging 136
9-7 Normal operational configuration returned. 137
10-1 FlashCopy V2 establish pairs performance test 141
10-2 ESS Copy Services Web Interface - Volumes panel 147
10-3 ESS Copy Services Web Interface - Select task type panel 148
10-4 ESS Copy Services Web Interface - Select Copy Option panel 148
10-5 ESS Copy Services Web Interface - Tasks panel 149
10-6 ESS Copy Services Web Interface - Logical Subsystems panels 150
10-7 ESS Copy Services Web Interface - LSS panel - Select task type 150
10-8 ESS Copy Services Web Interface - LSS panel - Select copy option 151
11-1 PPRC Extended Distance background 158
11-2 The DB2 Consistency Group Procedure for Global Copy Scenario 159

12-1 Asynchronous PPRC . . . 168
12-2 Example of an Asynchronous PPRC configuration . . . 169
12-3 Create a set of test volumes at the remote site . . . 173
12-4 Implementation environment . . . 175
12-5 PPRC-XD pairs . . . 177
12-6 Establish Inband FlashCopy - nocopy, changerecording, inhibit write to target . . . 179
12-7 Asynchronous PPRC session started, primary volumes join pending . . . 181
12-8 PPRC-XD primary volume first pass in progress, volume status: join pending . . . 182
12-9 Secondary volume status. . . . 183
12-10 Logical Subsystem view of Asynchronous PPRC . . . 183
12-11 Status of the Asynchronous PPRC session. . . . 184
12-12 Primary volumes 00 and 01 joined the session, volume 02 is still join pending. . . . 184
12-13 PPRC-XD primary volume with first pass complete and part of the session . . . 185
12-14 PPRC target volume shows the FlashCopy sequence number. . . . 185
12-15 FlashCopy target volume with Consistency Group number. . . . 186
12-16 Asynchronous PPRC information panel, all volumes in session . . . 186
13-1 Our XRC environment . . . 202
13-2 Allocation of journal data sets . . . 223
14-1 GDPS/PPRC topology . . . 235
14-2 GDPS/PPRC HyperSwap . . . 237
14-3 GDPS/XRC topology . . . 238
14-4 GDPS setup . . . 241
14-5 Freeze options . . . 242
14-6 Cross-Site® reconfiguration. . . . 243
14-7 Site 1 failure . . . 244
14-8 A three-site, continental distance solution with no data loss . . . 246
14-9 A three site-double failure solution. . . . 247
15-1 FlashCopy with SET LOG SUSPEND . . . 252
15-2 Restore FlashCopy - Forward Apply Logs . . . 253
15-3 FlashCopy Test Environment. . . . 256
15-4 Establish the SYSPITR CRCR. . . . 265
16-1 FlashCopy Consistency Group . . . 283
16-2 Restore FlashCopy - Restart DB2 . . . 283
16-3 FlashCopy Consistency Group Test Environment . . . 285
16-4 ESS Copy Services - Volumes . . . 286
16-5 Establishing the FlashCopy pair . . . 287
16-6 Specifying FlashCopy Consistency Group. . . . 287
16-7 Task definition . . . 288
16-8 List of tasks . . . 288
16-9 Tasks grouping . . . 289
16-10 LSS panel . . . 290
16-11 Defining Consistency Created task . . . 290
16-12 Select Consistency Group Type . . . 291
16-13 Run the Freeze option task . . . 292
17-1 Test scenario overview . . . 302
17-2 PPRC test environment . . . 303
17-3 Select path . . . 305
17-4 Select tasks . . . 305
17-5 Select outgoing path . . . 306
17-6 Select path options . . . 306
17-7 Define task . . . 307
17-8 Select Source and Target volumes . . . 308
17-9 Establish synchronous PPRC copy pair . . . 308

17-10 Copy option 309
17-11 Define task 309
17-12 Synchronizing volume pair 311
17-13 Volume pair synchronized 311
17-14 FlashCopy PPRC target to FlashCopy target 313
17-15 After Freeze task issued with PPRC Consistency Group 327
17-16 FlashCopy Secondary to Third volumes 328
17-17 PPRC Resync volume pairs while the resynchronization is going on 329
18-1 XRC test environment 336
18-2 XRC journal data set definitions 337
18-3 XRC control data set definitions 338
18-4 XRC state data set definitions 338
18-5 Establishment of the XRC environment – System load 341
18-6 Scenario 3 – Part 1 356
18-7 Scenario 3 – Part 2 356
19-1 System PITR test environment 368
19-2 SMS definitions for system PITR test 369
19-3 Scenario – Restoring to an arbitrary point-in-time 370
20-1 SMS definitions for source volumes 394
20-2 Copy pool DSN$DB8X$LG 394
20-3 Copy pool DSN$DB8X$DB 395
20-4 Copy pool backup DB8XCPB 395
20-5 Storage group DB8XL 396
20-6 SMS definitions for source and target volumes 397
20-7 Status of the volumes after the two backups 404
20-8 DUMPCONDITIONING effect 406
21-1 The data sharing environment 412
21-2 DB2 and coupling facility 413
21-3 Recovery site requirements 415
21-4 Finding the ENDLRSN to use in a conditional restart 426
B-1 Select option P, Copy Pool 464
B-2 Define the database copy pool 465
B-3 Define the source storage group to the database copy pool 465
B-4 Define the log copy pool 466
B-5 Define the source storage group to the log copy pool 466
B-6 Select the Storage Group 467
B-7 Alter by Storage Group Name 467
B-8 Associating source and target storage groups 468

Examples

2-1 General contents of a SYSLGRNX entry for a page set 32
2-2 Adding the last archive log to the BSDS 36
2-3 Snippet Checkpoint queue from Print Log Map 37
2-4 End Checkpoint RBA used as SYSPITR 37
2-5 Last Log RBA written from DSN1LOGP 37
2-6 Example of last RBA from DSN1LOGP as SYSPITR 37
3-1 Determine the name and RBA of archive log. 49
3-2 Repro BSDS from the last archive log tape 50
3-3 Register the active logs in BSDS 50
3-4 Register the last archive log in BSDS 51
3-5 DSNJU004 sample output 51
3-6 DSN1LOGP sample JCL 52
3-7 Create a CRCR 53
3-8 Verify a CRCR 53
3-9 Conditional Restart 54
3-10 -DIS DATABASE(*) SPACENAM(*) RESTRICT 56
3-11 -DIS UTIL(*) 56
5-1 DFSMSdss COPY FULL with COPYVOLID 72
5-2 Backup-restore cycle with DUMPCONDITIONING and background copy 73
5-3 Backup to tape with no-background copy 75
6-1 DFSMShsm LIST command output 84
6-2 DSNJU004 output. 85
6-3 Invoking DFSMShsm Replication with DB2 utility 88
6-4 Invoking DFSMShsm Fast Replication with DB2 utility (DATA ONLY option) 89
6-5 DFSMShsm LIST command and output on DSN$DB8B$DB copy pool 89
6-6 DFSMShsm LIST command and output on DSN$DB8B$LG copy pool 89
6-7 DB2 Backup System Full 90
6-8 FRBACKUP output in syslog 90
6-9 FRBACKUP output in DFSMShsm backup log data set 91
6-10 Sample JCL of RESTORE SYSTEM utility 94
6-11 Sample JCL of RESTORE SYSTEM with LOGONLY option 94
6-12 RESTORE SYSTEM utility output 95
6-13 FRRECOVER messages in DFSMShsm log 96
10-1 ICKDSF job to reformat a z/OS volume and set DUMPCONDITIONING option . . . 152
10-2 DFSMSdss DUMP job 152
10-3 ICKDFS job to INIT a volume. 153
10-4 DFSMSdss RESTORE job. 153
11-1 DSN1LOGP Sample JCL for an active log 162
12-1 Establish PPRC paths 176
12-2 Establish PPRC-XD pairs 176
12-3 Establish FlashCopy Pairs 178
12-4 Establish FlashCopy Pair. 179
12-5 Defining and opening a session. 179
12-6 Populating the Asynchronous PPRC session with volumes 180
12-7 STARTASYNCCOPY - command syntax 181
12-8 Removing volumes from a session 187
12-9 Modify a session 188
12-10 Pause a session. 188

12-11 terminate a session 188
12-12 Withdraw a FlashCopy Pair 189
12-13 Query the status of Asynchronous PPRC - command syntax 189
12-14 Messages about the status of Asynchronous PPRC 189
12-15 Query Session devices - command syntax 191
12-16 Messages about Query Session devices command 191
12-17 Query the number of out-of-sync volumes - command syntax 192
12-18 Messages from Query the number of out-of-sync volumes 192
12-19 Query INCREMENTSTATUS - command syntax 193
12-20 Messages from Query INCREMENTSTATUS command 193
12-21 Query Relations - command syntax. 193
12-22 Messages from Query Relations command. 194
12-23 New RMF Link Performance Report 196
13-1 XADDPAIR command syntax examples by option 215
13-2 XQUERY command syntax example. 216
13-3 XQUERY MASTER example 217
13-4 XDELPAIR command syntax examples by option. 217
13-5 XEND command syntax example 218
13-6 XSUSPEND command syntax examples. 218
13-7 Restart ANTAS000 address space 231
13-8 MODIFY command 231
15-1 -SET LOG SUSPEND 257
15-2 DFSMSdss COPY with DUMPCONDITIONING 257
15-3 Sample output of DUMPCONDITIONING 258
15-4 -SET LOG RESUME 258
15-5 DFSMSdss DUMP 258
15-6 The utilities suspended during FlashCopy. 259
15-7 The threads suspended during FlashCopy 259
15-8 MODIFY CATALOG command to unallocate ICF catalogs 260
15-9 DFSMSdss RESTORE 261
15-10 Restore the second archive logs to disk 261
15-11 Repro to restore BSDS from the last archive log. 262
15-12 Find the last archive log in syslog 262
15-13 Register the last archive log in BSDS 263
15-14 Register the active logs in BSDS. 263
15-15 DSNJU004 output 263
15-16 Repro to make the second BSDS from the restored BSDS. 264
15-17 DSNJU003 to create a SYSPITR CRCR 265
15-18 Restart Control Record 266
15-19 Restart DB2 with a SYSPITR CRCR 266
15-20 DSNJU004 after DB2 restart 267
15-21 DSN1PRNT to retrieve RBLP in DBD01 269
15-22 DSN1PRNT output sample in DBD01 269
15-23 RESTORE SYSTEM LOGONLY 269
15-24 RESTORE SYSTEM output using LOGONLY. 270
15-25 start DB2 to remove access maintenance mode 270
15-26 -DIS UTIL(*) output 271
15-27 Display any active utilities 271
15-28 -DIS DB(*) SP(*) RESTRICT output 271
15-29 Recover objects used by utility in progress 272
15-30 Recover objects used by LOG NO event. 273
15-31 Recover objects created and loaded with LOG NO after FlashCopy. 274
15-32 Rebuild all indexes used by workload suspended during FlashCopy 274

15-33 Validate that recovery was successful 275
15-34 Find RBA of the latest checkpoint 276
15-35 DSN1LOGP using RBASTART(xxxxxx) SUMMARY(ONLY) 277
15-36 DSN1LOGP to find the RBA of last valid log record 277
15-37 DSN1LOGP using RBASTART(yyyyyy) SUMMARY(NO) DBID(FFFF) 279
15-38 DSN1LOGP to find the RBA of last valid log record easily 279
16-1 -DIS THD(*) 292
16-2 ICKDSF using REFORMAT DUMPCOND(SET) 293
16-3 Vary all FlashCopy target disks online 293
16-4 Sample output of DUMPCONDITIONING 293
16-5 DFSMSdss DUMP 294
16-6 MODIFY CATALOG command to unallocate ICF catalogs 294
16-7 DFSMSdss RESTORE 295
16-8 -DB8X STA DB2 295
16-9 -DIS UTIL(*) 296
16-10 -DIS DB(*) SP(*) RESTRICT LIMIT(*) 297
16-11 -TERM UTIL(DSNTE*) 297
16-12 -DIS DATABASE(*) SPACE(*) RESTRICT LIMIT(*) 298
16-13 REBUILD INDEX 298
16-14 CHECK DATA 299
16-15 REPAIR SET NOCHECKPEND 299
16-16 -DIS DB(*) SPACE(*) RESTRICT LIMIT(*) 299
17-1 Extended Long Busy State in MVSLOG 312
17-2 SDSF DA OJOB during FlashCopy” 312
17-3 SDSF DA OJOB after FlashCopy completes 313
17-4 Stop DB2 313
17-5 Offline volumes 314
17-6 Online volumes 314
17-7 DB2 restart log 315
17-8 Display utility 317
17-9 Display database 317
17-10 Terminate utility 318
17-11 Recovery pending resolution 318
17-12 Rebuild index 318
17-13 Repair Set Nocopypend 318
17-14 Check Data 319
17-15 Repair Set Nocheckpend 319
17-16 Display Database 319
17-17 .DB2 Set Log Suspend 320
17-18 SDSF ‘ DA OJOB during FlashCopy” 321
17-19 Set Log Resume 321
17-20 SDSF ‘ DA OJOB after FlashCopy completes 321
17-21 Restart Subsystem MVS LOG 322
17-22 Display utility 324
17-23 Display database 324
17-24 SDSF DA OJOB during PPRC Freeze operation 326
17-25 ‘DA OJOB’ after PPRC Consistency Group Created 328
17-26 During resynchronization of volumes in MVSLOG 329
17-27 After resynchronization completed in MVSLOG 330
17-28 DB2 restart log 330
17-29 Display utility 332
17-30 Display database 333
18-1 XSTART command 339

18-2 XSTART command – Messages in syslog. 339
18-3 XADDPAIR commands 339
18-4 XSTART command – Messages in syslog. 340
18-5 XQUERY command – Session level 340
18-6 XQUERY command – Session level – Summary report 340
18-7 XQUERY command – Volume level 340
18-8 XQUERY command – Volume level – Volume report 341
18-9 Establishment of the XRC environment – Completion message 341
18-10 XQUERY command – Session level – Summary report 342
18-11 Scenario 1 – Initial situation. 342
18-12 Scenario 1 – Stop the DB2 members 343
18-13 Scenario 1 – XEND command. 343
18-14 Scenario 1 – XEND command – Messages in syslog 343
18-15 Scenario 1 – Vary all primary volumes offline 343
18-16 Scenario 1 – XRECOVER command. 344
18-17 Scenario 1 – XRECOVER command – Messages in syslog 344
18-18 Scenario 1 – Vary all secondary volumes online 345
18-19 Scenario 1 – Clean up the DB2 structures (1/2) 345
18-20 Scenario 1 – Clean up the DB2 structures (2/2) 345
18-21 Scenario 1 – Clean up the structures – Verification. 345
18-22 Scenario 1 – Restart DB2 346
18-23 Scenario 2 – Initial situation. 347
18-24 Scenario 2 – Disaster. 348
18-25 Scenario 2 – Vary all primary volumes offline 348
18-26 Scenario 2 – XRECOVER command. 348
18-27 Scenario 2 – XRECOVER command – Messages in syslog 348
18-28 Scenario 2 – Vary all secondary volumes online 349
18-29 Scenario 2 – Clean up the DB2 structures (1/2) 349
18-30 Scenario 2 – Clean up the DB2 structures (2/2) 349
18-31 Scenario 2 – Clean up the structures – Verification. 350
18-32 Scenario 2 – Restart DT22 (1/2) 350
18-33 Scenario 2 – Restart DT24 (1/2) 350
18-34 Scenario 2 – Retained locks 351
18-35 Scenario 2 – Group buffer pool recovery pending (GRECP). 351
18-36 Scenario 2 – Logical page list (LPL) 352
18-37 Scenario 2 – Restart DT22 (2/2) 352
18-38 Scenario 2 – Restart DT24 (2/2) 353
18-39 Scenario 2 – Restricted state of DB2 catalog and directory 353
18-40 Scenario 2 – DSNDB01 restricted states. 353
18-41 Scenario 2 – DSNDB06 restricted states. 354
18-42 Scenario 2 – GRECP and LPL recovery for DB2 catalog and directory 354
18-43 Scenario 2 – GRECP and LPL recovery – Messages 354
18-44 Scenario 2 – DISPLAY UTIL 355
18-45 Scenario 3 – Initial situation. 357
18-46 Scenario 3 – XSUSPEND command. 357
18-47 Scenario 3 – XSUSPEND command – Messages in syslog 357
18-48 Scenario 3 – XQUERY volume report 357
18-49 Scenario 3 – XADDPAIR command. 358
18-50 Scenario 3 – XADDPAIR command – Messages in syslog 358
18-51 Scenario 3 – XQUERY summary report 358
18-52 Scenario 3 – XSUSPEND command – Messages. 359
18-53 Scenario 3 – Vary all primary volumes offline 359
18-54 Scenario 3 – XRECOVER command. 359

18-55 Scenario 3 – XRECOVER command – Messages in syslog 360
18-56 Scenario 2 – Vary all secondary volumes online 360
18-57 Scenario 3 – Clean up the DB2 structures (1/2) 360
18-58 Scenario 3 – Clean up the DB2 structures (2/2) 361
18-59 Scenario 3 – Clean up the structures – Verification 361
19-1 BACKUP SYSTEM utility 370
19-2 BACKUP SYSTEM utility – Output 370
19-3 BSDS content after BACKUP SYSTEM – DSNJU004 output 371
19-4 DBD01 header page – DSN1PRINT output 371
19-5 SET LOG SUSPEND command 372
19-6 Clean up the DB2 structures (1/2) 373
19-7 Clean up the DB2 structures (2/2) 373
19-8 Clean up the structures – Verification 373
19-9 MODIFY CATALOG commands 374
19-10 Change Log Inventory utility (DSNJU003) – SYSPITR CRCR record 374
19-11 Print Log Map utility (DSNJU004) – BSDS content after SYSPITR CRCR 375
19-12 DB2 restart with active SYSPITR CRCR 375
19-13 RESTORE SYSTEM utility 376
19-14 Messages sent to the console during RESTORE SYSTEM 376
19-15 RESTORE SYSTEM utility – Output 377
19-16 DISPLAY UTIL command 378
19-17 DISPLAY DATABASE 378
19-18 Added volumes – Messages sent by DFSMShsm 379
19-19 Output from the Restore utility 380
19-20 Added volumes – System log from RESTORE SYSTEM utility 380
19-21 Deleting a specific DB2 orphan data set 382
19-22 Restart checkpoint not found – Error messages 383
19-23 Restart checkpoint not found – Creating a SYSPITR CRCR record 383
19-24 RESTORE job output 384
19-25 System log for RESTORE with FLA 384
19-26 FLA informational messages to the console 385
19-27 MVS console - RESTORE SYSTEM messages with FLA 385
20-1 Filter lists for SMS storage classes 393
20-2 FRBACKUP command – PREPARE parameter 397
20-3 FRBACKUP command – PREPARE parameter – DFSMShsm activity log output . 398
20-4 LIST command – COPYPOOL parameter 398
20-5 LIST command – COPYPOOL parameter – DSN$DB8X$LG output 398
20-6 LIST command – COPYPOOL parameter – DSN$DB8X$DB output 399
20-7 BACKUP SYSTEM utility 399
20-8 BACKUP SYSTEM utility – Output 400
20-9 LIST COPYPOOL output after BACKUP SYSTEM 400
20-10 SET LOG SUSPEND 401
20-11 FRBACKUP command – EXECUTE parameter 401
20-12 FRBACKUP command – Completion message – Success 402
20-13 FRBACKUP command – Completion message – Failure 402
20-14 FRBACKUP command – EXECUTE parameter – DFSMShsm activity log output . 402
20-15 SET LOG RESUME 403
20-16 LIST COPYPOOL output after FRBACKUP 403
20-17 QUERY command – COPYPOOL parameter 405
20-18 QUERY command – COPYPOOL parameter – Output (1/2) 405
20-19 QUERY command – COPYPOOL parameter – Output (2/2) 406
20-20 COPY FULL 406
20-21 DFSMSdss DUMP command 407

20-22 MODIFY CATALOG command to unallocate ICF catalogs 407
20-23 DFSMSdss RESTORE command 407
20-24 DB2 restart messages – Syslog. 408
20-25 -DISPLAY UTILITY(*) output 408
20-26 -DISPLAY DB(*) SPACE(*) RESTRICT LIMIT(*) output 408
21-1 ARCHIVE LOG MODE(QUIESCE) command fail 418
21-2 Archive Log Command History on Print Log Map 419
21-3 Print Log Map - member D8F1 424
21-4 Print Log Map - member D8F2 424
21-5 DSN1LOGP output for member D8F1 425
21-6 DSN1LOGP output for member D8F2 425
21-7 Conditional restart record for data sharing. 426
21-8 Adding the last archive log to each BSDS for data sharing 427
21-9 ARM policy statement coded for Restart Light option 427
21-10 Issuing START commands 429
A-1 Sample JCL to execute CPBULIST in batch 454
A-2 Sample specification of PRIOTITY DD statement input. 456
A-3 CPBULIST sample output report 457
A-4 CPBULIST sample DUMP job JCL 458
A-5 CPBULIST sample RESTORE job JCL output 459
A-6 Sample JCL to execute BSDSSTAT in batch 460
A-7 Sample output report from BSDSSTAT 461

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

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

Trademarks

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

AIX®
CICS®
Cross-Site®
DataPropagator™
DB2®
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DRDA®
Enterprise Storage Server®
ESCON®
@server®
FlashCopy®
FICON®
Geographically Dispersed Parallel Sysplex™
GDPS®
HyperSwap™
IBM®
ibm.com®
IMS™
Language Environment®
MVS™
NetView®
OS/390®
Parallel Sysplex®
QMF™
Redbooks™
Redbooks (logo) ™
RACF®
RAMAC®
RMF™
S/390®
Sequent®
Sysplex Timer®
System/390®
Tivoli®
TotalStorage®
VSE/ESA™
VTAM®
WebSphere®
z/OS®
z/VM®
zSeries®

The following terms are trademarks of other companies:

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

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

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

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

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

Preface

DB2® for z/OS® is the database of choice for critical data for many enterprises. It is becoming more and more important to protect this data in case of disaster and to be able to restart with a consistent copy of the DB2 data as quick as possible and with minimal losses.

A broad range of functions can be used for the disaster recovery of DB2 subsystems. The traditional DB2 based solution consists of safe keeping and restoring image copies and logs. More general functions, applicable not only to DB2 data, but to the whole system, are hardware related, such as tape vaulting or disk volumes mirroring. Other functions are specific to DB2 such as the Tracker Site. There are also products providing replication capabilities which can be used for specific propagation requirements.

DB2 UDB for z/OS Version 8 has introduced two new subsystem wide utilities, BACKUP and RESTORE, which, by interfacing the copy pools functions of DFSMS 1.5, are able to provide Point-In-Time recovery capabilities.

The disaster recovery solution consists of the combination of coherent options that best fit in with the requirements, the current environment, and the investment.

In this IBM® Redbook we first introduce the main concepts, and the primary components for possible solutions. We then describe the most common solutions, and implement several recovery scenarios. All our tests were implemented with *DB2 UDB for z/OS Version 8*. We also include criteria for choosing a solution, and recommendations based on recovery best practices.

We focus on requirements and functions available for a disaster recovery strategy for data stored and managed by DB2 for z/OS. It is worth remembering that the non-DB2 data, logically or physically related to the DB2 applications, should be treated with equivalent and congruent solutions.

The contents of this redbook

The primary objective of this book is to document realistic scenarios related to DB2 disaster recovery. The focus is on *mirroring* IBM solutions using ESS, and on System PIT Recovery.

Before we get to that, we need to provide sufficient amount of background information to understand the requirements for those scenarios, and the functional components that are needed to implement them. We try not to repeat what is already documented in the standard DB2 manuals, but instead provide just enough detail about the storage functions to allow you to understand the implementation part without having to go too often to the storage manuals.

In Part 1, “The whole picture” on page 1, we introduce the concepts of business continuity, disaster recovery, and the services that IBM offers in this area, as well as providing a brief overview of the disaster recovery techniques available for DB2 for z/OS.

Part 2, “Disaster recovery major components” on page 41, we introduce the main concepts and the individual technologies for possible disaster recovery solutions. The intent here is to offer just enough information to help you understand the solutions. If you already know the DB2 traditional recovery concepts, and have the basic storage skills related to ESS Remote Copy Services, you can jump to Part 3, “General solutions for disaster recovery” on page 129. The topics covered are:

- Traditional recovery
- DB2 Tracker
- ESS FlashCopy
- SMS copy pools and DB2 point in time recovery
- Peer-to-Peer Remote Copy
- eXtended Remote Copy

In Part 3, “General solutions for disaster recovery” on page 129, we show how the previously described components can be combined in general disaster recovery solutions. The topics covered are:

- Split Mirror
- FlashCopy Consistency Group
- Global Copy PPRC-XD
- XRC: Global Mirror for z/OS
- Global Mirror PPRC
- Geographically Dispersed Parallel Sysplex

In Part 4, “Implementing disaster recovery scenarios” on page 249, we describe the step-by-step implementation of several realistic scenarios and provide pertinent recommendations. The scenarios are:

- Set Log Suspend - FlashCopy - More Log - Restore System Log Only
- FlashCopy Consistency Group and restart
- PPRC - FlashCopy from secondary
- XRC and restart
- Local recovery: System PITR
- Restart using tape dump of copy pools

In Part 5, “Additional considerations” on page 409, we summarize hints and tips for recovery best practices, and add information specific to data sharing environments.

Part 6, “Appendixes” on page 451 contains information on REXX execs developed during the project and how to download them, as well as the referenced documentation and a glossary.

The team that wrote this Redbook

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

Paolo Bruni is a DB2 Information Management Project Leader at the International Technical Support Organization, San Jose Center. He has authored several Redbooks™ about DB2 for z/OS and related tools, and has conducted workshops and seminars worldwide. During Paolo’s many years with IBM, in development, and in the field, his work has been mostly related to database systems.

Pierluigi Buratti is a Senior IT Architect working at IBM Global Services in Italy. He has 10 years of experience in Business Continuity and Disaster Recovery field. He provides fee-based services for designing and implementing disaster recovery solution for IBM BCRS customers. He also plays the role of technical consultant in validating and reviewing disaster recovery solution based on the new technologies that are (or are going to be) available on the market. His areas of expertise include Storage and z/OS. He has co-authored the following IBM Redbooks: *Disaster Recovery Library: S/390 Technology Guide,* GG24-4210-01, *Continuous Availability: Systems Design Guide,* SG24-2085-00, and *Continuous Availability: S/390 Technology Guide,* SG24-2086-00.

Florence Dubois is an Advisory IT Specialist at the IBM Software Group in France. She has supported DB2 for z/OS since 1997, providing technical expertise in benchmarks, proofs of concept, on-site customer system health checks, and performance reviews. She also teaches IBM classes on SAP implementation on zSeries®. Her areas of expertise include performance and tuning, database administration, backup, and recovery. She has co-authored the following Redbooks: *DB2 for z/OS Version 7: Selected Performance Topics*, SG24-6894; *SAP on DB2 UDB for z/OS: High Availability Solution Using System Automation,* SG24-6836; and *SAP on DB2 UDB for z/OS: Multiple Components in One Database (MCOD)*, SG24-6914.

Judy Ruby-Brown is a Consulting IT DB2 Specialist in the US with the Advanced Technical Support (ATS) organization. She is also an IBM Senior Certified IT Specialist. She has supported DB2 for OS/390® and z/OS for 16 years in the IBM Dallas Systems Center (now ATS). Her areas of expertise include disaster recovery, parallel sysplex and DB2 data sharing, high availability, and JDBC/SQLJ capabilities. She has presented on these topics at IDUG in the US, Europe, and Asia Pacific, as well as at the DB2 Technical Conferences and SHARE. She published the first successful DB2 offsite disaster recovery scenario in 1989. That scenario has been incorporated into each *DB2 Administration Guide* since DB2 V2R3. She holds a degree in Mathematics from the University of Oklahoma. Judy has co-authored the Redbooks *SAP R/3 and DB2 for OS/390 Disaster Recovery*, SG24-5343, and *DB2 for z/OS and OS/390: Ready for Java,* SG24-6435.

Christian Skalberg is a Consulting IT Specialist with IBM Software Group in Denmark. He has supported DB2 almost from the day it became available on MVS™ in the early 1980's and ever since. He has worked with most of the large customers in the Nordic region covering areas such as performance tuning, database and application design, restart/recovery, and disaster recovery. For last 25 years Christian has been the IBM representative to the G.U.I.D.E/Share Europe, GSE, Nordic Region DB2 and IMS™ workgroups and has presented at GSE conferences in Scandinavia, UK, and Germany, as well as at IDUG. He has co-authored two Redbooks in the distant past (DB2 Version 1 Performance and DB2PM Usage Guide). He has experience in data corruption and massive data recovery situations, including the coding and usage of pertinent tools.

Tsumugi Taira is a Project Leader with NIWS Co, Ltd. in Japan, a company specialized in advanced services in the Web/Server Business System Field, joint venture between IBM Japan and Nomura Research Institute. Tsumugi has six years of experience in DB2 for AIX® administration, and is currently on a one-year assignment to the IBM Silicon Valley Lab, DB2 for z/OS Performance Department, where she has worked on system set up, tuning, and recovery utilities. Her areas of expertise include ESS operations, Tivoli® Storage Manager, fault tolerant systems, and backup and recovery solutions.

Kyungsoon Um is a Senior IT DB2 and DB2 Tools Specialist in the IBM Software Group in Korea. She joined IBM Korea 14 years ago as a Database Administrator and Systems Programmer, supporting DB2 for OS/390. She had been working for 7 years as a DB2 instructor across all platforms in IBM Global Learning Services. For the last 3 years she has been involved in performance tuning projects using IBM DB2 tools. Her areas of expertise include performance tuning and database recovery. She has co-authored the Redbooks *DB2 for OS/390 and z/OS Powering the World's e-business Solutions*, SG24-6257.

A photo of the team is shown in Figure 1.

Figure 1 Left to right: Paolo, Pierluigi, Judy, Tsumugi, Christian, Kyungsoon, and Florence (photo courtesy of Bart Steegmans)

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

Chris Akey
Muni Bandlamoori
Eric Bateman
John Campbell
Julie Chen
Beth Hamel
Jeff Josten
Chun Lee
Tom Majithia
Debbie Matamoros
Chris Munson
Manfred Olschanowsky
Jim Ruddy
Jack Shedden
Jim Teng
Ping Wang
Chung Wu
IBM Silicon Valley Lab

Yufen Davies
Lisa Gundy
Bob Kern
Glenn Wilcock
IBM Tucson Lab

Udo Pimiskern
IBM Austria

David Petersen
IBM Washington

John Iczkovits
IIBM ATS Americas

Rich Conway
Bob Haimowitz
Emma Jacobs
Mary Lovelace
Yvonne Lyon
Bart Steegmans
Cathy Warrick
International Technical Support Organization

Roy Cornford
IBM UK

Johannes Schuetzner
IBM @server® Software Development, Boeblingen, Germany

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099

Part 1

The whole picture

In this part of the book, we introduce the concepts of business continuity, disaster recovery, and the services that IBM provides in this area.

We then provide a brief overview of disaster recovery techniques with DB2. The base is taking a copy of the system; the farther apart are the copies, the more data you can lose, or the longer it can take to recover. If the copy is keeping you current, then you can restart as if doing a local restart. We also discuss the importance of data consistency for DB2, introducing the concepts of rolling disaster and consistency groups.

Part 1 contains the following chapters:

- Chapter 1, “Business continuity” on page 3
- Chapter 2, “DB2 disaster recovery” on page 15

1

Business continuity

In this chapter we introduce the concepts of Business Continuity and Disaster Recovery, the need for them, their relationship, and the service provided by IBM in this area.

This chapter covers the following topics:

- Business Continuity definition
- IBM Business Continuity Strategy
- Business Continuity and Recovery Services

1.1 Business Continuity definition

Business Continuity as a concept is still maturing to adequately address the typologies of disasters. Its definition and perception continue to change. However, in the latest years, the BS 7799 (British Standard -Information technology - Code of practice for information security management) has generically defined the objective of business continuity management as:

To counteract interruptions of business activities and to protect critical business processes from the effects of major failures or disasters.

This definition has been adopted by ISO as ISO/IEC 17799:2000.

Here are some references for these topics:

- For BS7799:

  http://www.bsi.org.uk or http://www.bsi-global.com/index.xalter

- For ISO/IEC 17799:

  http://www.iso.org

1.1.1 Business Continuity and Disaster Recovery

Business Continuity (BC) and Disaster Recovery (DR) act at different levels in the organization: BC is the strategy at the enterprise level, while DR is the solution at the IT level.

Figure 1-1 helps in understanding the relationship between BC and DR.

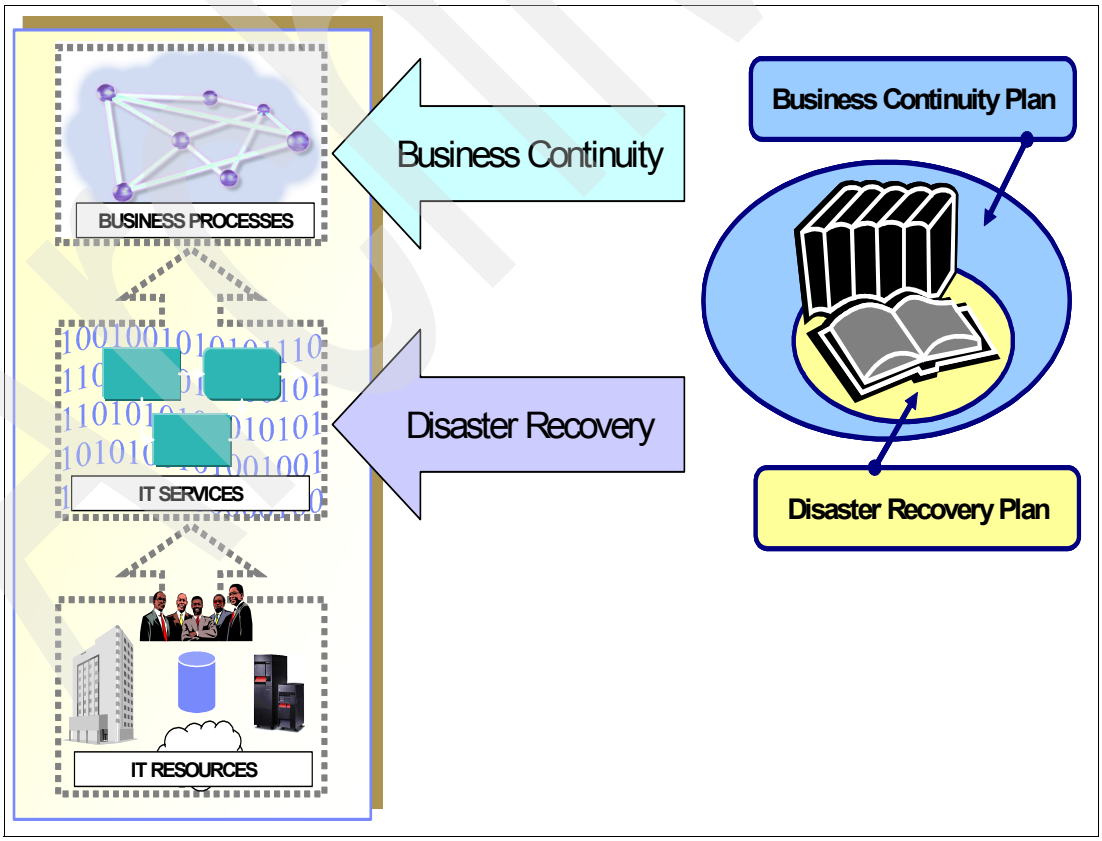


Figure 1-1 Business Continuity and Disaster Recovery relationship

Business Continuity relates to business processes and includes all methodologies and procedures required by the business processes to be continuously available. Those are described in detail in the Business Continuity Plan.

Disaster Recovery relates to IT services required by business processes in order to work properly and includes all methodologies and procedures to allow IT services to be supplied from an alternate IT site, should a disaster prevent the primary IT site from providing services.

The Disaster Recovery methodologies and procedures are described in detail in the Disaster Recovery Plan which is contained in the Business Continuity Plan.

1.1.2 What is a disaster?

An information disaster, following an unplanned event, either natural or resulting from someone's intentional or unintentional act, can cause the total or partial cessation of the operability of information systems and, as a consequence, the interruption of information processing activities. An interruption of information services is considered “disastrous” only if protracted for a significant period of time.

The duration of this period is the first parameter for defining a disaster for each type of organization: in some cases it can be measured on the order of minutes and, in other cases, weeks. The second parameter to be considered, and certainly as important as the duration of the unavailability of data and computer services, is the integrity of the information.

The destruction of data processing equipment leads to the destruction of all the information it supports. It should be said that, normally the data managed by the information system is backed up at predefined periods of time and kept in a secure location, far from the Data Processing Center (although, unfortunately, this doesn't always happen); but in any case, the data entered after the last backup before the disaster is irredeemably *lost*. If such data cannot be recovered by the personnel involved through manual procedures, or by adopting technological systems, the resulting damage could be much more serious than a prolonged loss of operability, with implications of a civil or penal nature.

In addition, for information environments distributed over several interconnected sites, the loss of data and operability at one of the computer centers could have repercussions on the operability of the entire network.

For this reason, identifying the maximum acceptable time for the unavailability of information services and the maximum supportable loss of data constitutes one of the basic steps for anyone who intends to confront the problem of “business continuity.”

Two very commonly used terms used to qualify the recovery solutions are now *Recovery Point Objective* (RPO) and *Recovery Time Objective* (RTO). These terms have replaced the former possibly more intuitive terms *lost data* and *downtime.* They are complementary and *both* must be considered when evaluating a disaster recovery solution. The enterprise's requirements for minimizing downtime and lost data vary by application and are determined by the cost to the organization of these two factors.

The term *disaster recovery plan* indicates the process that must be activated following a disaster for the purpose of restoring information services in the anticipated manner and time. Such a process manages and resolves a contingent situation (a disaster recovery plan is a particular “contingency plan”). It includes the procedures necessary for the restoration of the data and the network, and has, as its ultimate purpose, the reactivation of the operability of the users of information services. It must describe *how, where, and when* the user will resume their own working activities.

It is obvious that the structure of the disaster recovery process will not depend on the nature of the event or the origin of the disaster. Disaster recovery focuses on the process of reactivating information services according to the defined plans and scenarios, leaving aside the nature of the triggering event.

1.1.3 How to protect from a disaster

Any company, in order to protect its own interests, can undertake useful initiatives to defend itself from a hypothetical information disaster. Such initiatives can be divided into two fundamental categories: preventative and reactive.

Measures of a preventative character have the purpose of reducing the probability that a disaster will occur through the introduction of suitable security counter-measures. The cost of preventative measures is generally much less than the cost consequent to the loss of the products or services and to the cost for their reintegration.

Nevertheless, no security counter-measure of a preventative character, at an acceptable cost, can offer an absolute guarantee of invulnerability. It is, therefore, reasonable to provide for the possibility of a disaster and, in addition, it is advisable to make a business continuity plan capable of performing a rapid restoration of the damaged products or services, limiting the associated (and always serious) financial losses. Thus, measures of a reactive character have the purpose of mitigating the consequences of a disaster. In this case, the objective consists in minimizing the damage caused by the disastrous event.

1.2 IBM Business Continuity Strategy

In order to adequately protect operating processes in the event of an information disaster, it is important that an organization ensure an adequate balance between initiatives of a preventative character and those of a reactive character. The identification of an adequate protection and restoration strategy requires an understanding of the peculiar characteristics of the organization's mission.

IBM has developed a conceptual model, the Business Protection Model, that intends to systematically represent and confront all the aspects that must be considered for the purpose of adequately protecting and restoring the goods that are instrumental to the company's mission. The model conforms to the British Standard BS7799, “Code of Practice for Information Security Management,” published in 1999.

This model is part of the Business Continuity and Recovery Services (BCRS) methodology utilized by IBM when providing Business Continuity services. See Figure 1-2.

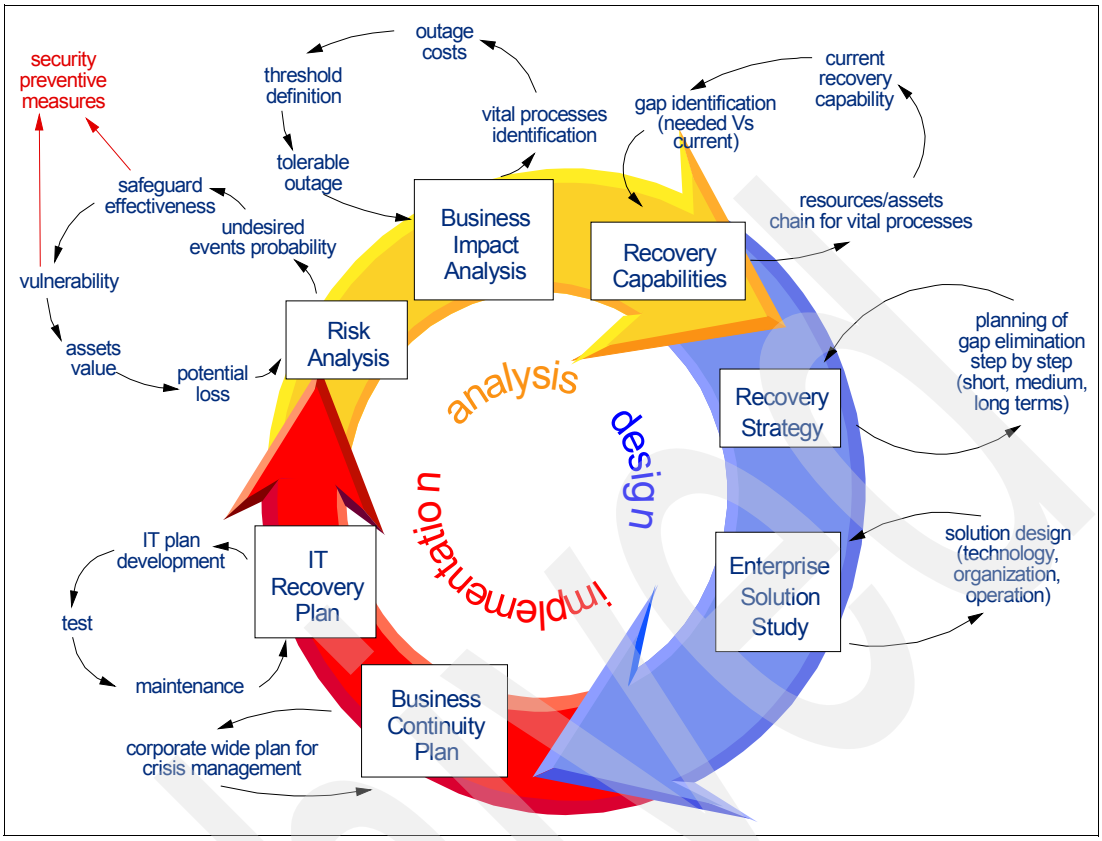


Figure 1-2 IBM BCRS Methodology

The first aspect to consider on the prevention side is to identify the threats that could impact an organization's fundamental goods, and to establish which of these would constitute unacceptable elements of risk. In fact, the Risk Management process has the purpose of defining which strategy to apply for each risk identified.

The Business Impact Analysis process can help in determining the impact consequent to the unavailability of critical resources as a function of the duration of the unavailability and of the loss of data. Once the dependence of the productive processes on the information services has been estimated, it is necessary to evaluate the current recovery capacity of the information systems (Recovery Capability Analysis).

Such an evaluation allows identifying an effective recovery strategy that is commensurate to the real needs for operational continuity of the various operational processes of a particular sector or environment of the company.

Once a *business continuity system* (or solution) has been created (Enterprise Solution Study and Implementation), it is necessary to prepare an emergency plan (Business Continuity Plan, Contingency Plan) that, if kept updated and periodically tested, is able to manage the various phases of the restoration in a proper and documented way.

Finally, once the emergency phase is overcome, it is necessary to reestablish all the operations necessary for guaranteeing the return to normal operations and the continuation of your activities.

1.2.1 Disaster Recovery solutions

There are many strategies that can be adopted for guaranteeing the restorability of a data processing center. Each of these, as already anticipated, is characterized by specific performances (restoration time, maximum loss of data) and, as a consequence, by specific costs. In fact, today's technology offers the possibility to create a broad range of solutions, all the way up to the de facto guarantee of the continuity of IT services (no loss of data, no perceptible interruption of service by the user) despite any undesirable event.

There are various meanings associated with some of the terms used to describe the various business continuity/recovery strategies (such as hot site, cold site, warm site, warm/cold start-up, and so on).

At this point we can anticipate that business continuity/recovery solutions can simply be divided into two broad families:

- *Warm ("continuity")* solutions are based on the adoption of on-line data-duplication techniques between the working center and the alternative center (at a secure distance), capable of restoring IT services in a few minutes/hours with a maximum data loss tending towards zero.
- *Cold ("disaster recovery")* solutions are based on the daily production of data backups on tape carried off-site (also at a secure distance), and provide the possibility of resuming service within about 48 hours in an alternative center.

In addition to the differences highlighted in terms of ability to react to an emergency condition, these two families of solutions have different characteristics that should be taken into consideration before making a choice. Table 1-1 summarizes the characteristics of these two families of solutions.

Table 1-1 Disaster recovery solutions general comparison chart

	WARM	**COLD**
Restoration time	A few hours	Days
Maximum data loss	Tending to 0	24 hours (assuming daily backups)
Protects from:	Up to all the undesirable "physical" events (from the unavailability of the office to hardware problems), since the solution's continuity levels also make it useful for facing problems of lower impact but of much higher frequency	► Prolonged unavailability (days) of the working center (disaster) ► "Logical" disasters (the complete daily back-up could be used as the base from which to restart in the face of application problems of data corruption)
Does not protects from:	"Logical" disasters (application problems of data corruption)	All events (physical and logical) with high frequency and low impact

	WARM	COLD
Characteristics	► Normally based on two interconnected centers ► Adopts "remote copy" (**should it be mirroring or replication)** techniques for handling data ► Since the alternative processing capacity must be available in minutes, it is almost always a proprietary solution (based on the purchase of technology for the alternative center) ► Complex periodic test modes	► Normally based on a second center to be used for emergency conditions and for periodic simulations ► Introduces a daily process of copying data onto tape, almost always in addition to backups already made (archive, application) because incompatible with each other ► Daily handling of data off-site ► In the majority of cases, the alternative center is "rented" from suppliers, the most cost-effective form when compared with proprietary solutions
Costs	**Investments:** Purchase of redundant processing capacity, doubling of storage **One-time:** Preparing space for the second center, double network certificates, design (complex) **Annual costs:** Space, maintenance on purchased hardware, software fees, line fees, managing the second center, maintaining the solution, periodic tests	**Investments:** Expanding storage to support the daily production of backups **One-time:** Design, double network certificates **Annual costs**: *Rent* for the alternative center, maintenance on the purchased hardware, software fees, daily maintenance of the solution, periodic tests, daily handling of data off-site
Implementation times	8 – 24 months, this includes searching for the *right* alternative site, large investments, management expenses, determining the roles of the centers, availability of a broadband connection	2 – 6 months, determined mainly by the time necessary to introduce daily data backup procedures
Constraints, dependencies and critical issues	► Implementing the Business Continuity Plan for IT users ► Maximum distance between the two centers, both due to technological limitations and the interconnection costs	► Implementing the Business Continuity Plan for IT users ► The availability of a complete and consistent copy of the data outside the scene of the disaster

1.3 General considerations on disaster recovery

Let us suppose that you have a system that satisfies your need for continuous operations, and you have also established backup and recovery procedures. But you realize that a disaster can still strike and that, without providing services for a length of time, your business

will have a major financial loss. There are tables that can tell you the typical cost by hour of a system outage for a large company by type of industry. There are also studies showing that an outage of 48 hours can put a large percentage of companies out of business.

A disaster is an event that renders the IT services unavailable for a period of time long enough to justify moving the IT facilities from the current production, or primary, location to a backup, remote, or secondary location. The remote location will be at a distance sufficient to not be impacted by the disaster.

Disasters can come from extraordinary events, such as earthquakes, hurricanes, or terrorist bombs, but also from power outages, floods, airplane crashes, or simply picketing lines during a strike. Generally, hardware failures and natural disasters are the most common causes. It can be wise to conduct a risk assessment for your enterprise and then invest in an effective business recovery plan. Computing costs have decreased, and several options are available to satisfy the demand for remote site recovery, but the cost depends on your requirements, and it definitely goes up with the currency of your data and the speed of recovery.

Disaster recovery solution

A disaster recovery solution consists of a set of processes put in place to restore computer operations after the declaration of disaster by the appointed management.

Once you are satisfied with your current local service levels, the starting point of your DR plan is to understand your business processes, determine and prioritize their recovery requirements, define *disaster* for your environment, and invest accordingly. The solution includes an extension to the set of procedures for backup of the primary site, and a set of procedures for recovery at the secondary site.

Since you want to resume service, you need to reactivate your applications and eventually restore access to the data. The recovery of your DB2 data from the disaster is a key part of your contingency plan, but DB2 data is only one component of your IT environment. You must first consider the people involved, communication, responsibilities, well-documented procedures, back-up premises, call desk, operating system, network, and so on.

The solution is based on determining the best options for your specific case, by evaluating your requirements in terms of:

- Data loss level
- Maximum allowed outage including time to resume service (service levels)
- Impact on current applications
- Consistency of data (within and across systems)
- Distance
- Data volumes

It also involves examining in detail the current environments, the platforms involved, and the availability levels already provided.

Refer to the currently available documentation and business recovery service providers for general information on risk assessments and business contingency plans. You can start with the general free Internet site maintained by the Disaster Recovery Journal:

http://www.drj.com/freelinks/links.html

The IBM Business Recovery Services Internet site is:

http://www-1.ibm.com/services/us/index.wss/it/bcrs/a1000411

The SHARE defined tiers

SHARE is a volunteer-run association providing user-focused education, professional networking, and a forum to influence Information Technology. In their Automatic Remote Site Recovery project in the early 1990's, they classified the levels of readiness in case of disaster according to six tiers ranging from all data lost to zero data lost and, more or less proportionally, from the least expensive to the most expensive. Such a classification is still useful today, as reported below:

- Tier 0 — No disaster recovery (DR) plan:

  No DR plan: All data is lost, and recovery is not possible.

- Tier 1 — Pickup Truck Access Method:

  Pickup Truck Access Method (PTAM): The system, the subsystem, and the application infrastructure, along with application data, is dumped to tape and transported to a secure facility. All backup data, such as image copies and archived logs, still on-site are lost in the event of a disaster (typically up to 24-48 hours). DR involves securing a DR site, installing IT equipment, transporting backup tapes from the secure facility to the DR site, restoring the system, the subsystem, and application infrastructure along with data, and restarting the workload (typically more than 48 hours). Cost factors include creating the backup copy of data, backup tape transportation, and backup tape storage.

- Tier 2- PTAM and hot site:

  PTAM and hot site: This is the same as Tier 1, except that the enterprise has secured a DR facility. Data loss is up to 24-48 hours, and the recovery window will be 24-48 hours. Cost factors include owning a second IT facility, or a DR facility subscription fee in addition to the Tier 1 cost factors.

- Tier 3 — Electronic vaulting:

  Electronic vaulting: This is the same as Tier 2, except that the enterprise dumps the backup data to a remotely attached tape library subsystem. Data loss will be up to 24 hours or less (depending upon when the last backup was created), and the recovery window will be 24 hours or less. Cost factors include telecommunication lines to transmit the backup data, and a dedicated tape library subsystem at the remote site in addition to the Tier 2 cost factors.

- Tier 4 — Active secondary site (electronic remote journaling):

  Active secondary site: This is the same as Tier 3, except that transaction manager and database management system log updates are remotely journaled in real time to the DR site. Data loss will be seconds, and the recovery window will be 24 hours or less (the recovery window could be reduced to 2 hours or less if updates are continuously applied to a shadow secondary database image). Cost factors include a system to receive the updates, and disk to store the updates, in addition to the Tier 3 cost factors.

- Tier 5 — Two site two-phase commit

  Two-site two-phase commit: This is the same as Tier 4, with the applications performing two-phase commit processing between two sites. Data loss will be seconds, and the recovery window will be 2 hours or less. Cost factors include maintaining the application in addition to the Tier 4 cost factors. Performance at the primary site can be affected by performance at the secondary site.

- Tier 6 — Zero data loss (remote copy):

  Zero Data Loss (remote copy): The system, the subsystem, and application infrastructure along with application data is continuously mirrored from the production site to a DR site. Theoretically, there is no data loss if using a synchronous remote copy, and only seconds if using an asynchronous remote copy. The recovery window is the time required to restart the environment using the secondary disks if they are data consistent (typically less than 2

hours). The synchronous solution conceptually allows you to reach zero data loss, but performance may be impacted, and care must be taken when considering a rolling disaster, which will leave inconsistent data at the secondary site.

The asynchronous solution means any distance, up to a few seconds of data loss, but with consistent recovery time and data.

Cost factors include the cost of the telecommunications lines used to shadow all of the data updates in real time, and possibly related CPU usage for transmission, in addition to the Tier 4 cost factors.

The return on investment

In order to assess the cost of your DR solution, there are several questions that need to be answered. What level of data currency is required? Can the data at the disaster site be a few seconds, a few minutes, or a few hours old? How consistent is the data expected to be? Multiple table consistency? Transaction consistency? Subsystem-wide consistency? A very high degree of data consistency will have a substantial cost in recovery time and resources.

When would this solution be rolled out? When would all of the hardware and software need to be available? When can the customer install the needed hardware and software levels? The costs for the remote site are very substantial and must be carefully evaluated against the benefits in data consistency and recovery time. The volume and type of accesses (reads against writes) directly impact telecommunication costs, disk space and processing power. Customers often need to segment their sets of data, so that the differing requirements can be met at a reasonable cost. There is no completely general, very low cost solution. Where some of the requirements can be relaxed, costs and the probability of success increase dramatically.

A valid approach is to build a solution in phases and increase currency with more expensive solutions as soon as the previous level has been validated and integrated in the standard operational procedures.

A simple example of a three phase solution is:

- Establish system-wide points of consistency with periodic vaulting.
- Secure a remote site with compatible environment.
- Add tools and techniques to increase data currency.

1.3.1 The lessons from September 11 and the requirements evolution

The events of September 11, 2001 in the United States of America have underlined how critical it is for businesses to be ready for disasters. The Federal Reserve, the Office of the Comptroller of the Currency, the Securities and Exchange Commission, and the New York State Banking Department (the agencies) have met with industry participants to analyze the lessons learned from the events of September 11. The agencies have released an interagency white paper on sound practices to strengthen the resilience of the US financial system.

For more information on this, refer to:

http://sec.gov/news/studies/34-47638.htm

The following list is a summary of lessons learned about IT service continuity:

- Geographical separation of facilities and resources is critical to maintaining business continuity. Any resource that cannot be replaced from external sources within the RTO should be available within the enterprise, in multiple locations. This not only applies to buildings and hardware resources, but also to employees and data, since planning

employee and data survival is very critical. Allowing staff to work out of a home office should not be overlooked as one way of being DR ready.

- Depending on the RTO and RPO (RTO and RPO are typically expressed in hours or minutes) it may be necessary for some enterprises to implement an in-house DR solution. If this is the case, the facilities required to achieve geographical separation may need to be owned by the enterprise.
- The installed server capacity at the second data center can be used to meet normal day-to-day data processing needs and fallback capacity can be provided either by prioritizing workloads (production, test, development, data mining) or by implementing capacity upgrades based on changing a license agreement, rather than by installing additional capacity. Disk resources need to be duplicated for disk data that is mirrored.
- Recovery procedures must be well-documented, tested, maintained, and available after a disaster. Data backup and/or data mirroring must run like clockwork all the time.
- It is highly recommended that the DR solution be based on as much automation as possible. In case of a disaster, key skills may not be available to restore IT services.
- An enterprise's critical service providers, suppliers, and vendors may be affected by the same disaster, therefore, enter into a discussion with them about their DR readiness.
- The recovery plan should also cover the following aspects:
  - Extensive loss of communication lines
  - Total loss of data on desktops and laptops
  - Public transport breakdown

Information technology is taking an ever more prominent role in company disaster recovery plans. The “e-business” model is spreading throughout all different kinds of companies: businesses increasingly run their databases over office-wide networks, link employees' computers via local area network connections, provide services over the internet and rely on e-mail. Losses of service to key IT systems can be extremely damaging. New regulations, like the Basel II rules for the European banking sector, are requesting resilient back office structures and highlighting IT business continuity issues. Analysts agree that investing in this area is growing.

However, often expensive solutions are being let down by lack of proper management, focus on the wrong types of emergencies, and insufficient review and testing. Many companies have prepared a backup infrastructure, but are not realizing that it needs to be constantly revisited to adapt to changes in the applications and in the evaluation of risks, and the recovery needs to be properly managed by the designated people with established processes. Change is the one constant in today’s business, and business continuity planning is not a prerogative of the IT structure, it is a responsibility across all business units. The disaster recovery plans need to be continually revisited to stay aligned with dynamic business realities and goals, and periodically tested to ensure that people and procedures perform as expected.

1.4 Business Continuity and Recovery Services

IBM Global Services operates at worldwide level in the context of continuity and disaster recovery services, making use of a world-wide development team, called Business Continuity and Recovery Services (BCRS).

BCRS is dedicated solely to business continuity concerns of IBM's customers. BCRS professionals have the expertise and tools necessary to design the right business continuity plan for your enterprise.

Whether providing an individual component or an end-to-end solution, their services include:

- Assessment of continuous operation readiness for critical processes.
- Development of in-depth business continuity strategies that map to business and IT requirements.
- Solution design, encompassing proven continuous-availability techniques and risk management, as well as traditional Disaster Recovery disciplines, processes, and methodologies.
- Integration of business continuity with critical business applications and IT initiatives, including e-business, enterprise resource planning (ERP), availability management, asset management, and server consolidation.
- Documented plans for the entire enterprise or individual business unit that integrate the full range of business continuity and recovery strategies.
- Transformation of plans into detailed procedures and processes, including testing to help evaluate readiness.
- Strategy proposals to help prevent high-impact risks and emergency situation management preparation.
- Validation of existing recovery assumptions, including shortfalls. This validation also can include testing your plan to simulate real-life disaster declarations.

As a leading provider of business continuity and recovery solutions, IBM delivers distinct advantages. In addition to decades spent perfecting business continuity programs, BCRS offers an unrivaled track record of helping companies anticipate, prevent, and recover from the disruptions that impact their business operations.

BCRS professionals understand the role technology plays in business and the impact a technology disruption can have. Every developed solution takes into consideration both the immediate and long-term impact a disruption can have on your business.

BCRS services rely on business continuity and recovery specialists with skill and experience derived from more than 12,000 contracts, over 400 recoveries, tens of thousands of test events. They manage over 100 recovery facilities, are ISO 9001 certified, and have received the highest customer satisfaction rating in the industry.

BCRS services have been recognized with three of the top industry awards:

- Reader's Choice Award - *Today's Facility Manager Magazine*
- Hall of Fame Award - *Contingency Planning & Management Magazine*
- 1999 Solution Integrator Impact Customer Satisfaction Award - *Solution Integrator Magazine*

For additional information, refer to:

http://www.ibm.com/services/us/bcrs/html/worldwide.html

2

DB2 disaster recovery

In this chapter we introduce several topics related to disaster recovery (DR) of DB2 environments. First we provide some general considerations in order to lay the ground rules and definitions, then we briefly introduce the options and features available for a DB2 subsystem DR solution. Starting with Part 2, “Disaster recovery major components” on page 41, we go into more detail and describe components, solutions, and operational scenarios.

The chapter covers the following topics:

- Introduction to DB2 disaster recovery solutions
- DR solutions in terms of RTO and RPO
- Data consistency
- DB2’s disaster recovery functions
- Determining the RBA for conditional restarts
- Actions to take when there are active utilities

2.1 Introduction to DB2 disaster recovery solutions

IBM provides a broad range of functions to be used for disaster recovery of DB2 subsystems. Some are general functions applicable to the whole system, such as tape vaulting or remote copy services; some are specific functions and parameters within DB2 that can help with your disaster recovery solution, such as COPYDDN, RSITE, archive to two different units, LIMIT BACKOUT, and Tracker Site. There are also external functions, such as data replication products, that can be used for specific requirements. The solution consists of the combination of coherent options that best fit in with the requirements and the current environment.

Figure 2-1 shows some of the solutions, and positions them in terms of data loss against recovery time.

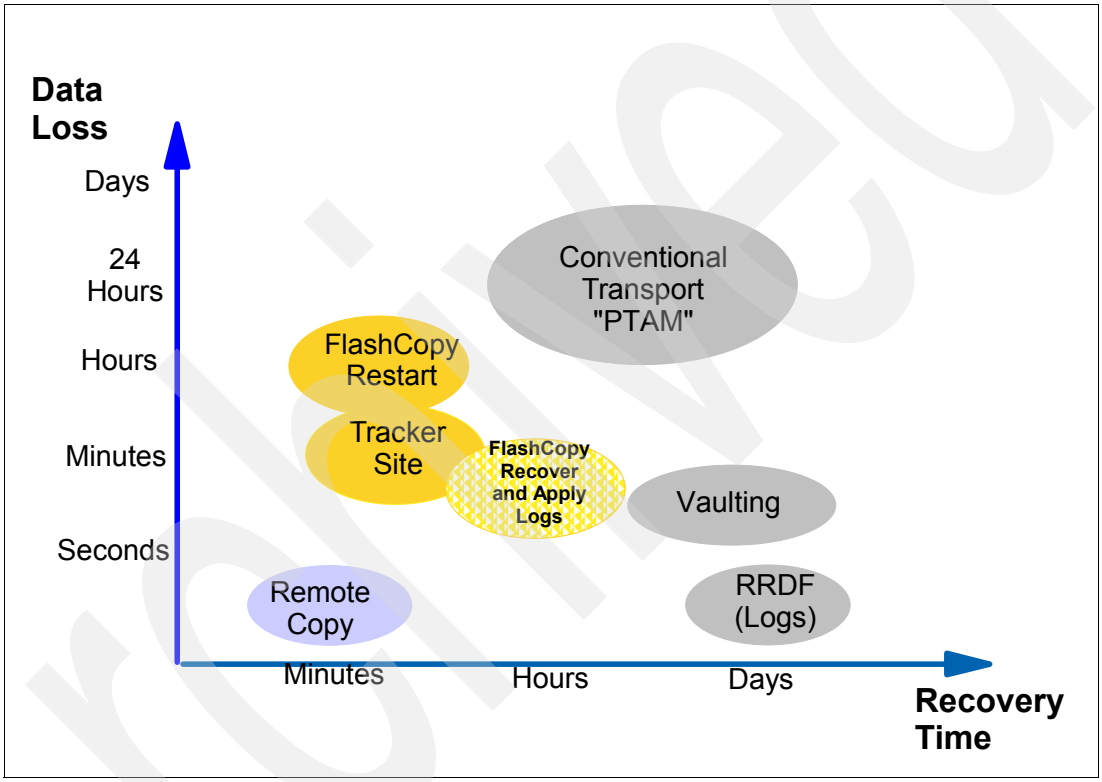


Figure 2-1 Data loss (RPO) and recovery time (RTO)

Some general types of solutions are:

- Volume dumps and conventional transport
- Image copies, log archives, BSDS and conventional transport
- Remote tape library or continuous vaulting
- Tracker Site
- RRDF E-Net
- Remote Copy:
  - Peer-to-Peer Remote Copy (synchronous PPRC) optionally with GPDS
  - eXtended Remote Copy (asynchronous XRC)

Other more specific solutions based on application modification or packages are:

- Data replication
- Data sharing (short distance; for example, a data center)
- Distributed access, multiple site update

2.1.1 Conventional transport

Conventional transport, the PTAM solution of Tier 1, or the Tier 2 modification with a predefined remote site, is still the most widespread current choice; see Figure 2-2. The *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413, documents the process in some detail. This method is often the base for others but, of course, it only allows your recovery to be as current as the last transport: with a daily pick-up and transfer off-site, you can lose up to 24 hours of data.

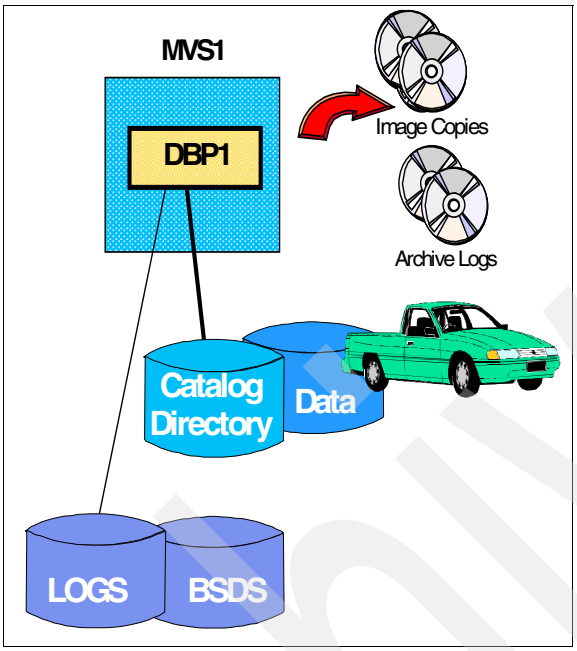


Figure 2-2 PTAM configuration

The basic assumption here is that the local environment is totally recreated at the recovery site, where sufficient compatible hardware must be present. The z/OS system has been completely restored, and ICF catalog structures are available with SMS, SMF, security subsystem, and your transaction manager. z/OS can be part of the periodic transport, or it can be a dormant operating system ready to start with a few last-minute well-documented changes.

The second assumption is that the whole standard DB2 recovery infrastructure has been backed up and is consistent: image copies have been made of all critical data, DB2 catalog, and directory; and log archives have been copied and transported to the remote site. Whatever you need for recovery at the local site, you will also need at the recovery site, plus a bit more.

The data currency is that provided by the log archives, and the time to restart depends on the length of the recoveries. Points of consistency created by storage functions which can create a consistent copy of the whole application and DB2 subsystem can certainly speed up the recovery process but need to be taken without impacting daily operations.

A daily transport of all dumps and a more frequent periodic shipment of log archives will give you more currency, but a longer recovery time.

A remote tape library provides a faster and more reliable means of transit, but the recovery techniques are essentially the same as for conventional transport.

If you can afford it, the availability of a hot remote site can accelerate the restart time by using preventive priming of hardware and software infrastructure and by log applying techniques, such as the DB2 Tracker function.

2.1.2 Remote Copy Services

The prime purpose for backing up, copying, and/or mirroring data is to be prepared for a possible disaster. Every business and every application has different requirements for the level of recovery needed to protect the data and the business needs. In some cases, not only must the data be protected, but facilities and equipment must be set up to be able to restart critical applications at a remote site. Many of these applications cannot tolerate loss of data — or they require that the data written to their volumes be consistent, and that the last writes all occurred at the same point-in-time. In other cases, a business can accept the risk and loss of a few hours or days, by rolling back to a recoverable consistent point.

The problem with traditional disaster recovery is that each software subsystem (CICS®, IMS, DB2, VSAM, and others) has its own recovery technique. Because an application is typically made up of multiple software subsystems, it is impossible to get a time-consistent backup across all subsystems unless the applications are stopped, and a dump of the overall system (point of consistency) is taken. But this impacts availability, and systems tend to run 24 hours a day, 7 days a week. Note, however, that backups are still required in a remote copy environment.

Another issue is the number of objects to be recovered. Complex environments and large ERP applications define tens of thousands of DB2 objects, and dealing with them on an individual basis is cumbersome and unproductive.

Remote Copy is a function initially introduced by the IBM 3990 Model 6 Storage Controller, later extended to the RVA family, and now integrated in the Enterprise Storage Server® (ESS family of devices. This function continuously duplicates on a remote (secondary) storage server any update done on a local (primary) storage server.

Copy Services allow you to have a consistent concurrent copy of the primary system providing the foundation for a recovery at the remote site, which appears to be the same as that experienced with a system crash at the primary site.

In each situation the solution chosen will depend on the resources available and the cost to implement the solution balanced against the business risk.

Many design characteristics and advanced functions of the IBM TotalStorage Enterprise Storage Server Model 800 contribute to protect the data in an effective manner. The ESS family of Enterprise Storage Servers (which currently includes the ESS Model 800, ESS Model 800 with Turbo option, and the new ESS Model 750) provides an extensive set of hardware and software features designed to implement storage infrastructures to help keep your business running 24 hours a day, 7 days a week. These features constitute the IBM TotalStorage® Resiliency Core Technology.

The following functions are key components of this technology:

- FlashCopy®, also known as IBM TotalStorage FlashCopy
- PPRC:
  - Synchronous PPRC, also known as IBM TotalStorage Metro Mirror
  - Asynchronous PPRC, also known as IBM TotalStorage Global Mirror
  - Asynchronous Cascading PPRC, also known as IBM TotalStorage Metro/Global Copy
  - PPRC Extended Distance, also known as IBM TotalStorage Global Copy

- XRC (Extended Remote Copy) - *Model 800 only*:
  - XRC, also known as IBM TotalStorage z/OS Global Mirror
  - Three-site solution using Synchronous PPRC and XRC, also known as IBM TotalStorage z/OS Metro/Global Mirror

Other DASD vendors have similar remote or “instant copy” services.

FlashCopy Version 1

FlashCopy is designed to provide a point-in-time copy capability for logical volumes. FlashCopy creates a physical point-in-time copy of the data, with minimal interruption to applications, and makes it possible to access both the source and target copies immediately. FlashCopy Version 1 is an optional feature on the ESS.

FlashCopy Version 2

FlashCopy Version 2 delivers new FlashCopy functions and enhancements designed to help improve business efficiency, along with FlashCopy performance improvements designed to help minimize operational disruption. FlashCopy Version 2 is an optional feature on the ESS.

FlashCopy Version 2 includes support for all previous FlashCopy functions, plus these:

- Data set FlashCopy, providing a new level of granularity for the zSeries environments
- Multiple Relationship FlashCopy, allowing a source to have multiple targets
- Persistent FlashCopy option, where the FlashCopy relationship does not automatically end when the background physical copy ends, but continues until explicitly withdrawn.
- Incremental FlashCopy, providing the capability to “refresh” a FlashCopy relationship
- Elimination of the LSS constraint: a source and target relationship can span logical subsystems (LSS)
- Establish time improvement, designed to provide up to a 10 times reduction

Peer-to-Peer Remote Copy (PPRC) Version 1

A hardware-based disaster recovery solution designed to provide real-time mirroring of logical volumes within an ESS or between two distant ESSs. PPRC has two basic modes of operation: synchronous and non-synchronous.

The Synchronous PPRC implementation (PPRC-SYNC) is a synchronous remote copy solution where write operations are completed on both copies (primary and secondary ESSs) before they are considered to be done. Thus, the recovery data at the remote site will be a constant real time mirror of the data at the local site as the applications do their updates. PPRC Version 1 is an optional feature on the ESS.

PPRC operations are entirely at the disk volume level. Write sequence consistency is preserved by the updates being propagated to the second site in real time. Databases that are spread across multiple volumes may be unrecoverable if a rolling disaster causes the secondary volumes to be at an inconsistent level of updates. Options of the PRRC and the GPDS offering can help in this situation See 2.2, “Data consistency” on page 24 for details.

PPRC Version 2

PPRC Version 2 provides new options for long-distance remote copy solutions:

- PPRC over Fibre Channel links: Fibre Channel Protocol (FCP) can be used as the communications link between PPRC primary ESSs and secondary ESSs. FCP reduces the link infrastructure by at least 4 to 1 when compared to ESCON®, and relieves logical and physical path constraints. The supported distance for Synchronous PPRC has been increased to 300 km (over FCP) but remains at 103 km for ESCON.
- Asynchronous Cascading PPRC: Asynchronous Cascading PPRC provides a long-distance remote copy solution for zSeries and open systems environments by allowing a PPRC secondary volume (involved in a PPRC synchronous relationship) to also simultaneously serve as a PPRC primary volume in a PPRC Extended Distance (PPRC-XD) relationship to the remote site. This new capability enables the creation of three-site or two-site Asynchronous Cascading PPRC configurations.
- Failover and Failback modes for Asynchronous Cascading PPRC: This is supported by the ESS Copy Services Web User Interface (WUI) and the ESS Copy Services Command Line Interface (CLI) on supported platforms, for both open systems and zSeries environments. For the zSeries environments, the ICKDSF utility can also be used to manage this function.
- Asynchronous PPRC: Designed to provide a long-distance remote copy solution across two sites using asynchronous technology. It operates over high-speed, Fibre Channel communication links and is designed to provide a consistent and restartable copy of the data at the remote site, created with minimal impact to applications at the local site. Compared to Asynchronous Cascading PPRC, Asynchronous PPRC eliminates the requirement to do a manual and periodic suspend at the local site order to create a consistent and restartable copy at the remote site.

PPRC Extended Distance (PPRC-XD)

PPRC-XD offers a non-synchronous long-distance copy option whereby write operations to the primary ESS are considered complete before they are transmitted to the secondary ESS. This non-synchronous operation results in a “fuzzy copy” at the secondary site; however, through operational procedures, a point-in-time consistent copy at the remote site can be created that is suitable for data migration, backup, and disaster recovery purposes. PPRC-XD can operate at very long distances (distances well beyond the 103 km supported with PPRC synchronous transmissions over ESCON) with the distance typically limited only by the capabilities of the network and channel extension technologies. PPRC-XD support is included at no additional charge when PPRC is purchased for the ESS Model 800.

Extended Remote Copy (XRC)

XRC is a combined hardware and software business continuance solution for the zSeries and S/390® environments providing asynchronous mirroring between two ESSs at global distances. XRC is an optional feature on the ESS.

For DB2, the recovery is easier because all volumes are brought to a *consistent status*, so a *DB2* restart can be done. The way to ensure recoverability is to use the following parameter and to place all DB2 volumes in the same XRC session:

```
ERRORLEVEL=SESSION
```

The ability to perform a DB2 restart means that recovery at the secondary site may be as quick as a recovery from a failure on the production system. The only drawback to an asynchronous implementation of remote copy is that the currency of the data may lag behind the primary system. This may result in some transactions having to be manually reentered after recovery at the secondary site. XRC externalizes a timestamp of the recovered system so that manual recovery is possible from a specified time. The time lag between the primary and the secondary sites can be minimized by performance tuning actions.

Concurrent Copy

Concurrent Copy offers another method for creating a point-in- time copy in zSeries and S/390 environments. The CONCURRENT option of DB2 COPY utility invokes DFSMSdss™ Concurrent Copy. The COPY utility records the resulting DFSMSdss concurrent copies in the catalog table SYSIBM.SYSCOPY with ICTYPE=F and STYPE=C or STYPE=J. You can subsequently run the DB2 RECOVER utility to restore those image copies and apply the necessary log records to them to complete recovery. This function can be useful for quickly cloning a subset of your DB2 data.

2.1.3 Remote Copy and DB2

When considering Remote Copy for DB2, you should consider that its mission is to keep disk volumes aligned. You must choose the set of volumes so that they include all components needed for normal recovery functions. You will need, after the restart at the remote location, all the usual bits and pieces that will allow you to resume doing normal recover locally. This means image copies, archive logs, and standard and tested recovery procedures that include DB2 catalogs. If you have image copies on tapes, and your recovery jobs on another system, special arrangements must be made.

Remote copy propagates the I/O WRITEs. If your DB2 average transaction has massive update/insert/delete activity, you will have a lot of traffic going to the remote control unit; this requires capacity planning for your bandwidth and some performance analysis.

DB2 writes to the log synchronously, so contention or delays due to the distance on the remote log will impact the commit times if you choose the synchronous solution (PPRC). DB2 writes to all other data objects mostly asynchronously as deferred writes. This is good for performance locally, but it might make things challenging when mirrored synchronously at a remote site because logical storage subsystems or control units are not inherently synchronized with each other and do not consider *dependent writes*. This is where the GDPS® functions will be needed; see 2.2, “Data consistency” on page 24.

2.1.4 Data replication

For DB2 for z/OS disaster recovery purposes, it is also possible to maintain a real-time copy of critical user data or log data at a remote site. As records are modified at the prime site, they are also transmitted to the recovery site and either vaulted or applied to receiving objects. The data loss for DBMS data can be largely eliminated depending on performance and characteristics of the mirroring application.

RRDF (E-Net)

Remote Recovery Data Facility (RRDF) for MVS was marketed by IBM several years ago. It is now available from Ubiquity, a software developer and distributor based in Melbourne Australia, and E-Net Software.

It maintains a real-time copy of log data at a remote site. As log blocks are written at the prime site, they are also transmitted to the recovery site and vaulted. The data loss for DBMS data can be kept at a minimum. The transmission uses standard SNA/VTAM® communications. Once at the recovery site, the log records are normally stored as RRDF archives until they are needed, when they are converted to DB2 archive logs.

Under a separate version of RRDF, you can have shadow data base on a different DB2 and apply the log data as source SQL statements. More information about RRDF may be obtained from:

http://www.ubiquity.com.au/content/suppliers/enet/rrdf/rrdf01.htm

DPROPR

DB2® DataPropagator™ replicates data between your central database and regional transactional databases, making business data available to the regional databases for prompt transaction processing.

DPROPR is an IBM data replication solution which has been integrated in DB2 UDB products for several years. DPROPR enables cross-platform data replication among all members of the DB2 UDB family. In combination with other DB2 Information Integration products, DPROP easily integrates non-relational data as well as data stored in non-IBM relational database systems into an enterprise-wide distributed multi-platform replication scenario.

Generally, the most common uses of DPROPR are as follows:

- Data distribution from one source database towards many target databases.
- Feeding a data warehouse from a production database, utilizing the data manipulation functions provided by the replication product. The replicated data can, for example, be enhanced, aggregated, and/or histories can be built.
- Data consolidation from several source databases towards one target database.

But it can easily be utilized to keep databases aligned, especially if subject to a limited amount of update activity.

From a technical point of view, the three main activities involved when replicating database changes from a set of source tables to a set of target tables are:

- Setting up the replication system
- Capturing changes at the source database and store them into staging tables
- Applying database changes from the staging tables to the target databases

DPROPR provides components to implement these main activities:

- The Capture component asynchronously captures changes to database tables by reading the database log or journal. It places the captured changes into change data tables, also referred to as staging tables.
- The Apply component reads the staging tables and applies the changes to the target tables.
- The Administration component generates Data Definition Language (DDL) and Data Manipulation Language (DML) statements to configure both Capture and Apply. The two main tasks are to define replication sources (also referred to as registrations) and to create replication subscriptions.

Replication sources are defined to limit the change capture activity to only those tables that are going to be replicated. Replication subscriptions contain all the settings the Apply program uses when replicating the change data to the target tables. To set up homogeneous replication between DB2 database systems, either the DB2 Control Center or the Replication Administration Center can be used.

Basically, the three components operate independently and asynchronously to minimize the impact of replication on your applications and online transaction processing (OLTP) systems. The only interface between the different components of the replication system is a set of relational tables, the DPROPR control tables. The administration component feeds these control tables when you define the replication sources and the replication targets. The runtime components (Capture and Apply) read the control tables to find out what they have to do. They also update the control tables to report progress and synchronize their activities.

Q Replication

Q Replication is a new deliverable of DB2 for Linux, UNIX® and Window on V8.2. It is an implementation of database replication over WebSphere® MQ as the transport mechanism. It provides low latency, asynchronous, and peer-to-peer replication. By keeping all the databases in sync efficiently and consistently, a company can enable employees, partners, customers and suppliers to leverage up-to-date corporate information assets to drive continued success.

DPROPR provides the IBM's SQL Replication solution, utilizing staging tables for the data propagation. Q Replication utilizes WebSphere MQ queues for its propagation technique. The Q Apply component reads transactions from WebSphere MQ queues and replays them on the target database. It is capable of applying transactions in parallel and reducing latency for replicas in peer-to-peer replication.

The Q Replication solution offers:

- Transaction Publishing in XML using WebSphere MQ queues
- Transactional, log-based capture providing high-speed, low-latency replication
- Event publishing to publish transactional data to applications in XML
- Significantly enhanced peer-to-peer replication function including support for multi-site update with robust conflict detection and resolution options
- An Enhanced Replication Center, a wizard-based GUI interface to define replication (including peer-to-peer)

A new redbook, *DB2 Information Integrator Q Replication: Fast Track Implementation Scenarios*, SG24-6487, is being prepared to illustrate the functions of this product. Check the ITSO Web site for its availability, expected early 2005.

RepliData

RepliData for z/OS (program number 5799-GKW) is a database replication tool designed to support multisite replica databases for distributed applications as well as remote hot-site backup. It is a z/OS solution which uses CICS for configuring and administering replication criteria, and DB2 Instrumentation Facility Interface (IFI) for fast and continuous access to log data. The main characteristics are as follows:

- High performance/Low latency:

  The architecture is designed to push rather than pull, allowing a change activity of 700 transactions per second, with 15 changed rows per transaction, with a latency of less than 5 seconds. This is equivalent to over 40 million database changes per hour. Changes are applied in near real-time with transmission and apply processes completed in parallel.

- Data integrity:

  It creates integral packages of changes, where a package contains all the changes successfully committed at the source in a single unit of work. Changes are transmitted using WebSphere MQ.

- Peer-to-peer replication:

  It supports “peer-to-peer” replication utilizing a collision detection function to keep two sources of data in sync and updated with the most current changes.

- Conditional scenarios:

  It provides an intelligent replication engine that can distinguish between different scenarios and can selectively replicate data to different sites.

- Industrial strength:

  It is used for business-critical applications involving high-volume database replication across multiple sites.

Note: Q Replication is the strategic solution for this environment, as it fully supports DB2 for z/OS V8, while RepliData does not.

2.2 Data consistency

We have seen that ESS Copy Services provide data mirroring capability: the automatic replication of current data from your primary site to a secondary site. The secondary site allows you to recover your data after a disaster without the need to restore DB2 image copies or apply DB2 logs to bring DB2 data to the current point-in-time.

Notice that the scenarios and procedures for data mirroring are intended for environments that mirror an entire DB2 subsystem or data sharing group, including DB2 catalog, directory, user data, BSDS, and active logs. You must mirror all volumes in such a way that they terminate at exactly the same point.

2.2.1 The rolling disaster

A rolling disaster is the typical real disaster, where your local site gradually and intermittently fails over a number of seconds. The various components fail in sequence. For example, if a data volume failed to update its secondary, yet the corresponding log update was copied to the secondary, this would eventually result in a secondary copy of the data that is inconsistent with the primary copy. The database would need to be recovered from image copies and log data. In all cases, notification of this miss must be known at secondary. When this happens for hundreds of volumes, without a clear notification of status of impacted secondary volumes, recovery can be extremely complex and long.

When using data mirroring for disaster recovery, you must mirror data from your local site with a method that does not reproduce a rolling disaster at your recovery site. To recover DB2 with data integrity, you must use volumes that end at a consistent point-in-time for each DB2 subsystem or data sharing group. Mirroring a rolling disaster causes volumes at your recovery site to end over a span of time rather than at one single point.

In a disaster (think of flood, fire, explosion), it is very likely that different logical storage subsystems fail at different times. It is also true that for each SQL UPDATE, INSERT, and DELETE issued by an application, DB2 will issue, at different times, several dependent writes on log data sets, table spaces, and index spaces allocated to DASD volumes spread across several LSSs. While the write to the log is externalized at commit time, the write to table spaces and index spaces are externalized when the buffer pool thresholds are reached for each page set in each buffer pool.

Figure 2-3 shows how a rolling disaster can cause data to become inconsistent between two subsystems at the recovery site.

The sequence of events at the primary site, which would cause data inconsistency at your recovery site, might occur as follows:

1. At 11:58 the application updates a column, and the page of the table space is updated in the buffer pool.

2. At 12:00 the application commits and the log record is written to the log device on logical storage subsystem 1.

   At 12:01 the connection between the logical storage subsystem 2 at the primary and the corresponding LSS at the remote site fails.

3. At 12:02 the update to the table space is externalized to logical storage subsystem 2 but cannot be propagated to the remote subsystem.
4. If the mirroring is implemented with **PPRC with the option CRIT(N)**, at 12:03 a log record is written to mark that the table space update was made on the log device on logical storage subsystem 1.
5. At 12:04 the logical storage subsystem 1 also fails.

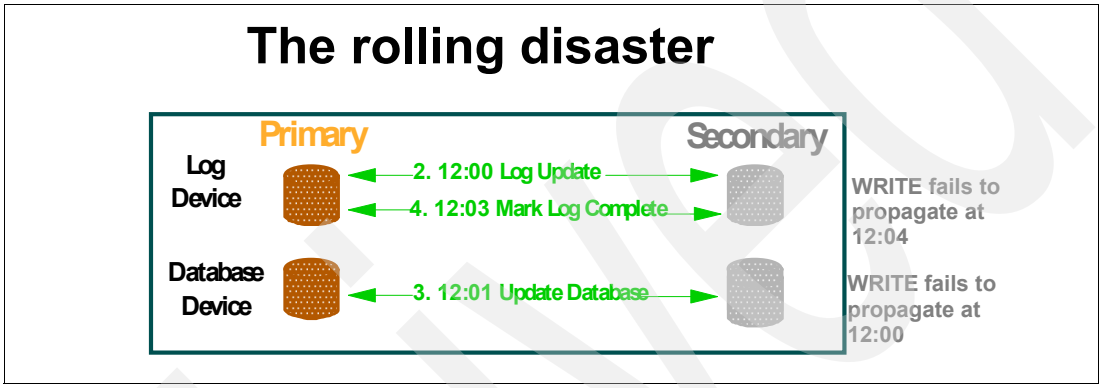


Figure 2-3 Rolling disaster

The logical storage subsystems have failed at different points in time; they contain inconsistent data. In this example, the log indicates that the update is applied to the table space, but the update is not applied to the data volume that holds this table space. At restart time that update will be lost. Other inconsistencies can apply to the indexes.

The option CRIT(Y) will prevent the WRITE to the log at the primary from completing successfully in case of any error in writing to the secondary. This will avoid the inconsistency, however, it may impact the availability of the primary. CRIT(N) needs to be helped by automation or GPDS in order to maintain data consistency.

Attention: Any disaster recovery solution that uses data mirroring must guarantee that all volumes at the recovery site contain data for the same point-in-time.

2.2.2 Consistency groups

A consistency group is a collection of volumes that contain consistent, related data. This data can span logical storage subsystems and disk subsystems. For DB2 specifically, a consistency group contains an entire DB2 subsystem or a DB2 data sharing group. The following DB2 elements comprise a consistency group:

- BSDS
- Active Logs
- DB2 Catalog
- DB2 Directory
- All user data
- ICF catalogs

Additionally, all objects within a consistency group must represent the same point-in-time in at least one of the following situations:

- At the time of a backup
- After a normal DB2 restart

When a rolling disaster strikes your primary site, consistency groups guarantee that all volumes at the recovery site contain data for the same point-in-time. In a data mirroring environment, you must perform both of the following actions for each consistency group that you maintain:

- Mirror data to the secondary volumes in the same sequence that DB2 writes data to the primary volumes.
- Temporarily suspend and queue write operations to create a group point of consistency when an error occurs between any pair of primary and secondary volumes.

To prevent your secondary site from mirroring a rolling disaster, you must suspend and queue data mirroring with the following steps after a write error between any pairs:

1. Suspend and queue all write operations in the volume pair that experiences a write error.
2. Invoke automation that temporarily suspends and queues data mirroring to all your secondary volumes.
3. Save data at the secondary site at a point of consistency.
4. If a rolling disaster does not strike your primary site, you can resume normal data mirroring after some predefined amount of time. If a rolling disaster does strike your primary site, follow the recovery procedure described in the *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413 under “Recovering in a data mirroring environment”, or refer to the procedure given later in this book.

You can use various methods to create consistency groups. The most relevant to DB2 are:

- XRC I/O timestamping and system data mover
- FlashCopy 2 Consistency Groups
- GDPS freeze policies
- The DB2 SET LOG SUSPEND command
- The DB2 BACKUP SYSTEM utility

Geographically Dispersed Parallel Sysplex

A *Geographically Dispersed Parallel Sysplex*™ (GDPS) is a multi-site availability solution that merges sysplex and remote copy technologies. The GDPS provides an integrated disaster survival capability that addresses the system, the network, and the data parts of an application environment. GDPS is available as a service offering and is described at the Internet site:

http://www.as.ibm.com/asww/offerings/mww62b1

The primary objective of GDPS is to minimize application outages that would result from a site failure, including rolling disasters, by ensuring that, no matter what the failure scenario is at the failing site, data in the surviving site is consistent and is therefore a valid base for a quick application restart. An *installation-defined policy* determines whether the switch will occur with limited loss or no loss of data.

In the event of a site failure (including disasters), the surviving site will continue to function and absorb the work of the failed site. In the event of a planned site outage, the workload executing in the site undergoing a planned outage will be quiesced and restarted at the other site. With GDPS, a single keystroke replaces a manual site switch process that could require several people to be present to perform their specialized tasks.

2.3 DR solutions in terms of RTO and RPO

Some of the RTOs involve restoring dumps from tape. Note that while we say 1-2 hours in the RTO column, this is actually a function of both the amount of data to be restored (gigabytes versus terabytes), and the number/capacity of tape devices available for the restoration.

In Table 2-1 most of the RPOs are compared to Traditional DR as a baseline. Since its RPO is 24 hours behind, we assume FlashCopy or copy pools have been produced once daily to give a more consistent comparison. If FlashCopy is performed four times daily, the RPO is reduced to six hours in the worst case.

Do not be concerned if you do not understand the solution descriptions here. We will explain each solution in detail as we go through this book.

Table 2-1 DR solutions comparison table

Solution	RPO	RTO	Cost	Explanation
Traditional DR	About 24 hours	24 hours	Least	No infrastructure required until test.
Tracker site	About 24 hours	<24 hours	More	RTO/RPO assumed once daily - must have active DB2 at recovery site.
XRC	A few seconds	1-2 hours	Bandwidth, z/OS, and SDM	
QReplication	A few seconds	0 Data available for read application at remote site.	Mirror system, bandwidth, setup	Only used for critical data.
GDPS/PPRC	0 If Freeze and Stop and Freeze and Stop Conditional.	1-2 hours	Most?	May reduce RTO if cloned workload across multiple site and HyperSwap™ used at recovery site DB2 data sets GRECP till CF Freeze.
FlashCopy Restart with Set Log Suspend and Resume	24 hours	<2 hours		No infrastructure needed until takeover. RPO based on one FC/24 hours and no additional log to apply.
FlashCopy CG Restart	24 hours	<2 hours		No infrastructure needed till takeover. RPO based on one FC/24 hours. Restart only solution.
FlashCopy with Set Log Suspend and Resume and Log Apply	Last archive log offsite	<24 hours? Restore System Logonly faster		Need transmission capability and remote automated tape library at recovery site

Solution	RPO	RTO	Cost	Explanation
PPRC-XD (Global Copy)	24 hours	1-2 hours		Assumes Set Log Suspend once per 24 hours.
PPRC-XD and log apply	Last DASD archive log	<24 hours Restore System Logonly faster than Recover, but slower than restart.		Assumes DASD archives and Set log suspend once per 24 hours.
Traditional DR and remote log	Last archive log	24 hours + More log to apply means longer recovery time.		Need transmission capability and remote automated library.
Dump copy pool to tape - restore and restart	24 hours	1-2 hours		No infrastructure needed until takeover. Assumes BACKUP SYSTEM once per 24 hours.
Dump copy pool to tape - apply more log	Last archive log	<24 hours Restore System Logonly faster than Recover, but slower than restart.		Need transmission capability and remote automated library for archive logs. Assume BACKUP SYSTEM once per 24 hours.
Global Mirror (Async PPRC)	<5 seconds	1 hour		Needs 3 sets of DASD and < 4 ESS at local site.

2.4 DB2's disaster recovery functions

The basic functions, the off-site recovery procedures for DB2 for z/OS are documented in the *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413, under "Remote Site recovery from a disaster at the local site."

In this section we review the concepts of DB2 data recovery that are common to most solutions through the three phases of:

- Backup
- Restart
- Recovery

2.4.1 Backup

As this book is focused on DASD mirroring solutions, we will only discuss the topic of image copies as a disaster recovery solution in Chapter 3, "Traditional recovery" on page 43.

However, image copies must be provided for all LOG(NO) events, as usual, for all solutions. Those table spaces are normally recovered after the fact using the inline image copy. We recommend providing image copies of the DB2 Catalog and Directory daily, in addition to any FlashCopies. The COPY utility cannot invoke FlashCopy through the use of a special keyword, such as CONCURRENT. In DB2 V8, FlashCopies are taken on a volume basis and are not registerable to SYSCOPY.

Users are tempted to let FlashCopy or another *instant* copy replace the traditional image copy, but this is not a good idea. In addition to the fact that the individual data set cannot easily be “flash-backed”, there will be a continual expansion of SYSLGRNX entries for a subsystem or a data sharing member with no way to purge old entries. The MODIFY utility requires one image copy in order to prune SYSLGRNX entries.

The following methods provide several alternatives in backing up for disaster recovery:

- SET LOG SUSPEND - FlashCopy - SET LOG RESUME
- BACKUP SYSTEM
- Consistency Groups

We now discuss all three of these methods.

SET LOG SUSPEND - FlashCopy - SET LOG RESUME

The DB2 command -SET LOG SUSPEND performs these functions:

- Takes a system checkpoint (for a non-data sharing system)
- Externalizes any unwritten log buffers to active log data sets
- Updates the BSDS with the high written log RBA value
- Holds 332K page writes unless the CI size is 32K
- Holds log write latch to suspend logging / update activities
- *Records the recovery base log point (RBLP) in DBD01* that determines the starting point for a subsequent RESTORE SYSTEM utility, if desired
- Writes a DSNJ372I message to indicate that update activity has been suspended at the current log RBA value
- Does not come to a syncpoint

After the DSNJ372I message is written, you submit jobs to FlashCopy (or another “instant copy”) for your entire DB2 subsystem: logs, BSDS, ICF Catalogs, DB2 Catalog/Directory, application data. You should provide enough bandwidth that you can accomplish this within a few minutes. When the FlashCopy relationships are established for all pairs, the copies are considered “logically” complete, and the system backup has been completed (although they may not have been physically finished or dumped to tape if that is required).

The DB2 command - SET LOG RESUME can then be issued to resume logging and update activity and to delete the DSNJ372I message from the z/OS console.

BACKUP SYSTEM

The Backup System utility, new to DB2 V8, can be generally considered to incorporate all the steps in the previous backup in “SET LOG SUSPEND - FlashCopy - SET LOG RESUME” on page 29 with just a one statement utility. It requires z/OS V1.5, DFSMShsm™ with the COPYPOOL construct, and SMS managed DASD. It can be used to back up the DATA copy pool (including ICF catalogs defining the DB2 Catalog/Directory and the application data) and the LOG copy pool (including ICF catalogs for the active logs, BSDSs, and DASD archive logs) for a full backup. It can also back up only the DATA copy pool, assuming that you will provide all the logs required since the backup to recover the entire subsystem to a given point-in-time.

BACKUP SYSTEM performs the following main tasks:

- Allows most updates
- Disables 32 KB page writes, if the CI size is not 32 KB, to eliminate integrity exposure
- Disables system checkpoint, data set creation, deletion, rename, and extension operations
- Acquires new PITR lock in X mode to ensure no restore is taking place on other members
- Serializes with the RESTORE SYSTEM process

- *Records the Recover Based Log Point (RBLP) in DBD01*. This is the starting point for the log apply phase using the RESTORE SYSTEM utility.

During the backup, DB2 does the following operations:

- DB2 invokes the DFSMShsm ARCHSEND service to take full volume DASD-to-DASD copies of the 'DB' COPYPOOL DFSMShsm, schedules multiple tasks, and FlashCopies the volumes in parallel.
- ARCHSEND returns control to DB2 once the “logical” copies have completed for ALL volumes in DATABASE copy pool (normally several minutes).
- DB2 updates the BSDS with the system backup information.
- DB2 invokes DFSMShsm ARCHSEND to take full volume copies of the LOG copy pool, in the same manner as occurred for the DATABASE copy pool.
- DB2 then resumes activities.

BACKUP SYSTEM is designed for local subsystem point-in-time recovery and is global in scope. That is, it pertains to an entire DB2 subsystem or data sharing group and not to a limited number of objects. It is not directly usable for a disaster recovery solution, as the copy pools are stored on DASD by generation (85 are possible, though IBM recommends 2).

Consistency groups

The appeal of this method is that it provides “no outage” to DB2, as opposed to the -Set Log Suspend methodology. Usually a storage person sets up the FlashCopy (2) relationships among the LSSs and DASD subsystems and then issues a storage command to create the consistency group. The DASD hardware takes care of creating the consistency group by responding to DASD requests with an extended long busy response (the default is 2 minutes but can be changed). The DASD mirrors are split using a Freeze control unit command and DB2 does not experience an outage, the saying goes.

However, there is an outage, while it may not surface to DB2. Aside from the questions of “what happens if it does not get the consistency group in 2 minutes?”, “how do I know it was successful?”, and “how do I know it failed?”, the other issue is the use to which you will put this consistency group.

You cannot use the consistency group created as input to RESTORE SYSTEM LOGONLY. You can use the consistency group in a plain restore and restart of DB2, but if you want to apply additional logs to it, you can only use the RECOVER xxxxx LOGONLY for each object in your subsystem, an onerous task.

> **Attention:** The recovery base log point (RBLP) is recorded only by the BACKUP SYSTEM utility and the -SET LOG SUSPEND command. The RBLP is the starting point for the Log Apply in the RESTORE SYSTEM utility.

We illustrate how it can be done using FlashCopy, in Chapter 10, “FlashCopy Consistency Group” on page 139.

We believe that the -SET LOG SUSPEND command process is cleaner, and also does not limit use of the RESTORE SYSTEM utility.

2.4.2 Restart

In this section we highlight the considerations on restart.

Bootstrap Data Set (BSDS) resync

DB2 uses its recovery log and the bootstrap data set (BSDS) to determine what to recover when restarting. The BSDS identifies the active and archive log data sets, the location of the most recent DB2 checkpoint on the log, and the high level qualifier of the integrated catalog facility catalog name. If DB2 was not closed in an orderly fashion, the two copies of the BSDS might be out of sync. DB2 V7 with PQ73038 and V8 will automatically resynchronize the two BSDSs if the timestamps are not the same. DB2 chooses the most current BSDS of the two.

If successful, the following message is issued:

```
DSNJ131I csect-name DUAL BSDS MODE RESTORED FROM BSDSn
```

The system window between the writing of the BSDS01 and the BSDS02 records is the most common cause of the timestamp mismatch at the local site.

This handles the situation experienced in GDPS where there was a timestamp mismatch at the recovery site due to the timing window that has always existed between the writing of BSDS01 and BSDS02.

But if the logs are on different LSSs, then you must identify the lowest RBA on whichever current active log is the earliest and truncate there.

Conditional restart

At restart time, DB2 always attempts to locate the last log RBA written before termination and continues from there. A conditional restart is a special process, activated by control records in the BSDS, which is used when you need to skip some portion of the log processing during DB2 restart.

You use the following procedure:

1. While DB2 is stopped, run the change log inventory utility using the CRESTART control statement to create a new conditional restart control record in the BSDS.
2. Restart DB2. The types of recovery operations that take place are governed by the current conditional restart control record.

A conditional restart can safely be performed only when the objects are recovered to a prior point of consistency since normal recovery operations can be partially or fully bypassed. It is very important to choose this point of consistency carefully. It is often useful to run the DSN1LOGP utility and review a summary report of the information contained in the log before setting the conditional restart.

The conditional restart control records are of two types: the *normal* conditional restart and the *system point-in-time recovery* (SYSPITR). The criteria by which the point to recovery to (RBA or LSRN) is chosen are also different.

We go in more detail reviewing the two cases and describing how to determine the point to recover to in 2.5, “Determining the RBA for conditional restarts” on page 35.

SYSLGRNX entries

The purpose of SYSLGRNX is to identify on the DB2 log where the updates are for each object. On the first *update* of a page set, an entry is created with the contents shown in Example 2-1.

Example 2-1 General contents of a SYSLGRNX entry for a page set

```
DBID/OBID
Starting RBA
Stopping RBA
Starting LRSN
Stopping LRSN
Member name if data sharing; otherwise zeroes
```

The Starting RBA (and LRSN) are updated, but the Stopping RBA/LRSN field contain zeroes. When the data set is closed or pseudo closed, those values will be completed.

In the event that the RECOVER utility is needed, DB2 knows which sections of its log contain updates for this object and it does not have to pass the log where no updates have occurred. Historically, this was implemented in DB2 2.3 and extended for data sharing in DB2 V4.

Note: There are seven data sets that do not have SYSLGNRX entries. They are the 5 table spaces of the DB2 Directory (DSNDB01), DSNDB06.SYSCOPY, and DSNDB06.SYSGROUP. We refer to them as “always open”. If we must recover any of them, we must pass all the log since the image copy was taken, whether or not any updates ever occurred.

While it has been historically used for table spaces, DB2 V6 indexes with COPY(YES) also had entries in SYSLGRNX, as they could be recovered from an image copy and the log, without being reconstructed via REBUILD INDEX.

Observe that each entry reflects updates by a specific member in data sharing. This means that in a 4-way data sharing group each of the 4 can have entries for the same times. Entries can explode in SYSLGRNX extents if a user is not diligent about pruning old entries. This activity is accomplished using the MODIFY utility. MODIFY deletes old image copies and the SYSLGRNX entries that apply to them.

Important: If you rely on FlashCopy or other forms of “instant” copies as your back up to the exclusion of the COPY utility, your SYSLGRNX entries cannot be pruned. Only an image copy recorded in SYSCOPY will allow MODIFY to be successful, so at least one valid copy must exist. Many users are not aware of this situation until the ever-increasing extents of SYSLGRNX are exposed.

Stopping SYSLGRNX entries are recorded when you STOP the data set, STOP DB2, QUIESCE WRITE(YES) the data set. It also occurs during Pseudo Close, discussed later in this section.

In the event of a DB2 abnormal termination, none of the data sets open for a given DB2 (member) are closed. This leaves SYSLGRNX with a valid starting entry, but zeroes in the closing entry. If this condition is not resolved, the Recovery consequences can be severe.

In this case, DB2 would apply the log from the starting RBA/LRSN to currency or to the next starting/stopping pair. This could encompass minutes or many hours of log data that must be passed. So, during *end restart,* the phase that follows backward recovery (Phase 4), DB2 updates all the data sets whose stopping entries are still zeroes and thus closes out the start/stop pair. During end restart, DB2 performs other housekeeping tasks necessary to hang out the “open for business” sign when restart is complete.

Pseudo Close

For both CLOSE YES and CLOSE NO page sets, DB2 automatically converts infrequently updated page sets or partitions from read-write to read-only state according to the values you specify in the RO SWITCH CHKPTS (PCLOSEN) and RO SWITCH TIME (PCLOSET) fields of installation panel DSNTIPN. If either condition is met, the page set or partition is converted from read-write to read-only state.

These values, introduced in DB2 V2.3, were originally designed to limit the amount of log that DB2 had to read during recovery. It is a way of “telling” DB2 where the updates are. It has also been called *read-only switching*. Other sections of the log can then reliably be bypassed.

The two DSNZPARM values that are used are:

- PCLOSET — amount of time without an update (default is 10 minutes
- PCLOSEN — number of DB2 system checkpoints without an update (default is 5)

These values had undesirable effects when data sharing was introduced in DB2 V4. If a data set had been pseudo closed and was not being shared at the moment, but a few minutes later GBP dependency recurred, there was significant overhead in re-establishing sharing mode. This “ping-ponging” in and out of data sharing was not desirable, so many users disabled the PCLOSEN value.

Prior to DB2 V7, DB2 system checkpoints were determined by the number of updates specified in LOGLOAD. Checkpointing was unpredictable in terms of time. A million updates occurred in 5 minutes during heavy batch and in 2 hours during an online day. Five checkpoints could occur in 5 minutes or 5 hours. This made PCLOSEN unpredictable and so many users disabled it, specifying abnormally high values for it.

DB2 V7 introduced CHKFREQ, that gave the ability to checkpoint in either number of updates or number of minutes. For data sharing members, we recommend its specification in terms of minutes, and for the z900/z990 fast processors, 5 minutes or less.

Important: DB2 V8 does not experience unexpected overhead when returning to data sharing after Pseudo Close due to the changes in child / page locking.

CLOSE (YES) versus CLOSE (NO) data sets

The default for DB2 for CREATE or ALTER TABLESPACE has been CLOSE(YES). Shortly after a data set has been pseudo closed, it will then be physically closed. Historically, users have been reluctant to specify CLOSE(YES) for performance reasons, both the overhead of physically closing a data set and the overhead of physically opening the data set in the first transaction that accesses it. CLOSE(NO) is almost universally practiced by DB2 users.

DB2 V8 behavior (and DB2 V7 after service) is such that if CLOSE(YES) is specified, the result is that many fewer data sets are likely to be open during restart after an abnormal termination of DB2, and thus the end restart phase will complete more quickly. If you can identify non-critical page sets, you may alter them to CLOSE(YES) to obtain a faster restart after a crash or for disaster recovery. When the effort in a mirroring scenario is to restart quickly in order to meet a Service Level Agreement (SLA), this approach can reduce restart.

Postponed recovery

If you have a long running unit of work that has not completed processing at the time one of your DB2 system terminates, it could take a long time to restart that DB2 system because DB2 has to back out all the uncommitted changes for that unit of work. This is one of the major reasons why we recommend that your applications be designed with frequent commit points, to minimize the amount of backout processing required at restart.

Starting with DB2 Version 6, you can limit the amount of backward log processing that occurs at restart, thus preventing long running URs from delaying the availability of a DB2 system. The LIMIT BACKOUT ZPARM on installation panel DSNTIPL controls whether you wish to postpone some backward log processing. The BACKOUT DURATION ZPARM on the same panel acts as a multiplier for your checkpoint interval.

If the LIMIT BACKOUT ZPARM is set to YES or AUTO, then at restart DB2 will limit the number of log records processed by any unit of work to the value of BACKOUT DURATION multiplied by the checkpoint frequency. If you specify AUTO for LIMIT BACKOUT, then DB2 will postpone the recovery for units of work that meet the BACKOUT DURATION criteria, then will automatically recover those units of work once the restart completes. If you specify YES for LIMIT BACKOUT, then DB2 will not attempt to recover those units of work that qualify for postponed recovery until you issue the RECOVER POSTPONED command.

See *"Active log data set parameters: DSNTIPL"* in Chapter 6 of the *DB2 UDB for z/OS Version 8 Installation Guide*, GC18-7418, for instructions on setting these ZPARMs to take advantage of postponed recovery.

2.4.3 Recover

We will not utilize the Recover utility in our disaster recovery scenarios, since our DASD mirroring solutions are desirable precisely because they emphasize DB2 *Restart*, which is fast, versus *Recovery*, which is slow. Data recovery in all our scenarios is implicitly accomplished by a DB2 restart. There are essentially two possibilities:

1. Restart of mirrored volumes, restore of tapes containing FlashCopy volume dumps, or FlashCopies themselves if there is a third set of DASD available. This is a normal DB2 restart similar to a local DB2 crash.
2. Conditional Restart and Restore System Utility if you have more log you can apply in addition to a set of volumes that you restore.

Applying the logs — Restart or Apply more log?

As we go through this book, we will show you how each solution can provide a consistent point for your entire DB2 subsystem or data sharing group. We will call that a *consistency group* and it can be used to restart DB2 normally. That is, the intent is for our consistency group to behave at the recovery site similar to a DB2 crash at the local site.

However, if you have provided additional archive logs at the recovery site, you have the capability of applying them to the consistency group in some of the scenarios. The question becomes:

1. Should you restart from your consistency group and achieve a fast RTO but increase your RPO?
2. Should you apply more log and increase the RTO while reducing the RPO?

These are business decisions that should be made in disaster recovery planning. Circumstances, however, can change it dynamically. If only a couple of hours have passed since you created the consistency group, it may be more feasible to simply restart quickly. If 20 hours have passed, there may be a strong desire to apply the log to minimize manual efforts at reprocessing work load.

Regardless of which you choose, we will try to identify any subtleties in either choice as we describe each scenario.

2.5 Determining the RBA for conditional restarts

In this section we show the conditional restart control records for the *normal* conditional restart and the SYSPITR. We also describe the criteria by which the RBA (or LSRN for data sharing) are chosen. These considerations are reported here since they apply as soon as we start discussing disaster recovery functions. More examples will be shown in the recovery scenarios of Part 4, “Implementing disaster recovery scenarios” on page 249.

2.5.1 Choosing an RBA for normal conditional restart

For a *normal* conditional restart you are likely to use the RBA at the end of an archive log.

Consider the situation where you have archive logs at the recovery site that you want to apply. Remember the format of an archive data set on tape is that the BSDS is file 1 and the archive itself is file 2. If the two are not on tape, they will have the same sequence number, but the BSDS is contains a “B” while the archive is prefixed with an “A”.

First you repro the BSDS archive to the “real” BSDS.

Then you run Print Log Map for the BSDS. See Figure 2-4, which shows a snippet of a Print Log Map for our example

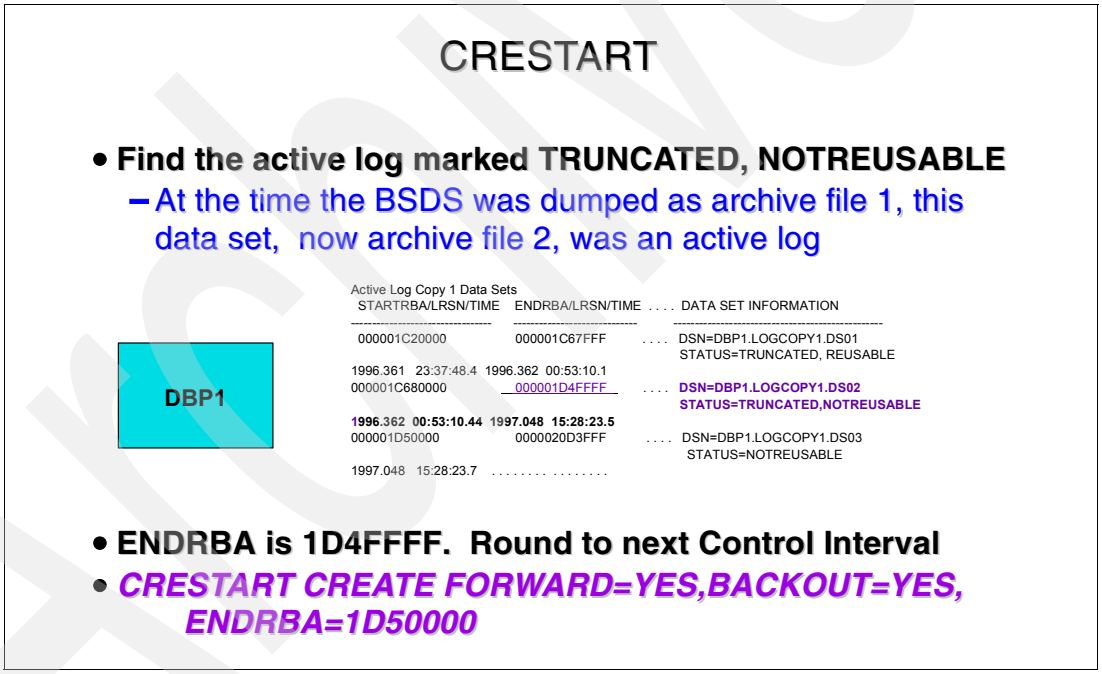


Figure 2-4 How to determine the correct log and ENDRBA to use for Conditional Restart

Relative to the abbreviated Print Log Map output shown in this figure, we look down the right side for the DBP1 active log section. We look at the status. The first one is truncated, but reusable. This log was truncated the last time it was used, but it has already been archived (since it is reusable). It is not the one we want. The status of DS02 is truncated, not reusable. It meets both our criteria: truncation as a result of -ARC LOG, and not reusable, since the BSDS was offloaded while this log (now file 2 of the archive) was still active. DS03 is not reusable and it is the current active log (at the local site). So DS02 is the log we want and we record its ENDRBA value. Note if two data sets are marked TRUNCATED, NOTREUSABLE, the one which should be chosen is the one with the lowest ENDRBA. Users who vault their archive logs probably will not have been truncated; they will be marked only NOTREUSABLE.

The RBA used for a Conditional Restart requires an even Control Interval (CI) boundary — that is, one that ends in '000'. We must round any value up. Before data sharing, DB2 had no need to have another value: it either wrote an entire CI or it didn't.

This log has not yet been added to the BSDS. The data set name is the same one as is in the BSDS you recovered, with the exception of the low order qualifier, which is Annnnnnn instead of Bnnnnnnnn.This is shown in Example 2-2.

Example 2-2 Adding the last archive log to the BSDS

```
NEWLOG DSNAME=hlq.ARCLG1.Dyyddd.Thhmmsst.Annnnnnn, COPY1VOL=xxxxxx,CATALOG=YES,
STARTRBA=00001C68000,ENDRBA=000001D4FFFF
```

You must add the second copy of the archive log and active logs to the BSDS. This page does not show that activity. All the Change Log Inventory statements can be added in one invocation of the utility.

After the updates have been made, you can REPRO this BSDS to the second copy. If you prefer, you can REPRO before the update and include both BSDS data sets in Change Log Inventory job.

Many users restore some number of archives to DASD. This has two advantages: faster recovery due to direct reads of the archive, and parallel recoveries, as multiple jobs can share one DASD archive. Another benefit is that a long running unit of recovery can progress much faster if it doesn't have to read a tape backward (this only comes into play during restart).

If you are going to restore TAPE archives to DASD for disaster recovery, make sure the BLKSIZE specified in DSNZPARM is 24576 when the archive was created, so that you can fit two blocks per track (the default is 28K and wastes DASD at a time when you are more likely to be constrained). The archive log block size cannot be changed after the data set was created.

2.5.2 Choosing an RBA value for SYSPITR

You must choose a value for CRESTART CREATE,SYSPITR=value for the RESTORE SYSTEM or RESTORE SYSTEM LOGONLY utility. The value will be an RBA if you do not do data sharing, or LRSN if your environment is data sharing.

Unlike the ENDRBA described previously in “Conditional restart” on page 31, there are different assumptions for development of the SYSPITR RBA:

- It must be an starting RBA of a log record, not a rounded up value.
- It is an “inclusive” value.

DB2 will read the log record with the specified RBA, and truncate the log right after this log record. If you use the RBA of the last log record written, which is produced using DSN1LOGP, no log truncation occurs, but DB2 will come up in PITR mode.

You can obtain the value in several different ways:

- **Method 1:** Run the QUIESCE utility on a dummy table space through automation for this purpose. They can query SYSCOPY for the dummy table space and use the RBA returned directly as the SISPITR.

  Other users should start with Print Log Map for the last archive offsite.

- **Method 2:** This one requires less effort, but not as much log is included.

  a. Find the Checkpoint Queue.

b. Find the first checkpoint in the queue (that is, the most current one). Look down the list until the last checkpoint before the end of the archive is found.

c. Find its END CHECKPOINT RBA as the SYSPITR value.

This example shows a snippet of a Print Log Map and the checkpoint that is used.

Example 2-3 Snippet Checkpoint queue from Print Log Map

```
         CHECKPOINT QUEUE
                     23:43:56 JULY 18, 2004
      TIME OF CHECKPOINT       23:24:48 JULY 18, 2004
      BEGIN CHECKPOINT RBA            00007E90D000
      END CHECKPOINT RBA              00007E9174A8
      TIME OF CHECKPOINT       23:22:19 JULY 18, 2004
      BEGIN CHECKPOINT RBA            00007C752000
      END CHECKPOINT RBA              00007C75E24E
      TIME OF CHECKPOINT       23:19:49 JULY 18, 2004
      BEGIN CHECKPOINT RBA            00007A586000
       END CHECKPOINT RBA              00007A5912FA
```

Example 2-4 shows the SYSPITR.

Example 2-4 End Checkpoint RBA used as SYSPITR

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DB8XU.BSDS02
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 CRESTART CREATE,SYSPITR=00007C75E24E
```

- **Method 3:** This one requires more effort, but returns the last RBA written on that archive.

  a. From the Print Log map, identify the last checkpoint before the end of the log. (Make sure its END CHECKPOINT RBA is still on this log.)

  b. Select the corresponding START CHECKPOINT RBA as input to DSN1LOGP in one of two ways:

     - STARTRBA(xxxxxxxxxxxx) SUMMARY(ONLY)
     - STARTRBA(xxxxxxxxxxxx) SUMMARY(NO) DBID(FFFF)

  The output of both is the last RBA written as shown in Example 2-5.

Example 2-5 Last Log RBA written from DSN1LOGP

```
DSN1213I DSN1LGRD LAST LOG  RBA ENCOUNTERED 0000653FE401
```

The latter option (specification of a non-existent DBID) is nicer since the printed report is very small. However, the first one shows URs in process at the SYPITR point which might be useful in some cases. See Example 2-6.

Example 2-6 Example of last RBA from DSN1LOGP as SYSPITR

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DB8XU.BSDS02
//SYSPRINT DD  SYSOUT=*
```

```
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 CRESTART CREATE,SYSPITR=0000653FE401
//*
```

We obtained an additional 34 MB of log data applied by finding the last valid record as opposed to using the End Checkpoint record RBA.

2.6 What if DB2 utilities were running?

In the *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413, there is a section called “What to do about utilities in progress”. The procedure described there does not apply to the scenarios we describe in this book. In the former case, all you have are image copies and archive data set. In our case the DASD is mirrored. So our process is opposite that of traditional disaster recovery described in the *Administration Guide*.

Though we describe the scenarios as looking as though DB2 crashed at the local, that is not entirely true. In certain cases we might be able to restart utilities after a local crash. For the scenarios we describe, we cannot restart them. We do not assume that your sort work or other utility work data sets are mirrored, whether DASD or tape. Most utilities can be terminated and can be rerun later. Examples of those would be Runstats and Copy.

A single phase utility will back out at restart. Those with internal commit points, like COPY, just need to be termed.

The two utilities we are concerned with in this section are LOAD and REORG. We suggest actions that you should take. We believe you should make those decisions at the local site, long before it becomes an issue during a disaster recovery exercise involving critical data.

We are not assuming work data sets can or will be mirrored — certainly not Sort Work data sets. Some work data sets might be on tape also.

An inline copy is not useful, as our data is mirrored and is at the recovery site. It will not be at the recovery site unless it is on DASD or the tape unit is within the supported distance. While some of the phases create inline copies on tape, there is not enough time to have them transmitted. So we can't assume they are available to us.

2.6.1 LOAD

LOAD sets objects during the RELOAD phase in RECP. If there are RI relationships, it will also set CHKP.

- SHRLEVEL CHANGE (V7)
  - This is insert processing. It backs out during restart to the latest commit point. You can term it and finish it at your own convenience.
- LOG(NO) SHRLEVEL NONE
  - If you have finished the RELOAD phase, your data is intact. Your indexes can be rebuilt and you must decide what to do about any CHKP. Are you going to run CHECK DATA on the objects after you have rebuilt the indexes? For large objects with many RI relationships, that can consume valuable time. If you believe RI relationships are not in jeopardy, you can use repair to reset CHKP.
- In all other cases, you must recover to the start of the utility.

– You will need an image copy taken prior to the LOAD at the recovery site. You also need all the log that must be applied to that copy to bring the table space to currency.

CHKP resolution

LOAD will set CHKP in the RELOAD phase if RI relationships exist. It persists throughout the SORT/BUILD phases. We do not assume you have SYSERR or any other interim data sets. You will never know the results of the LOAD. CHKP is a preemptive restriction, set in anticipation of RI relationships. It doesn't mean that RI conflicts actually exist. Indexes in RBDP can be rebuilt even while CHKP is on the table space.

However, what will you do about CHKP objects? You can REPAIR to reset CHKP or you can run CHECK DATA on the object.

If RELOAD has completed, you have the data. You want to make the application available ASAP. Do you reset CHKP so you can use the data? It is probably expedient. Is it productive to run CHECK DATA on large table spaces where it might run a long time? You get to make that choice. You should make that choice according to business requirements and reach consensus while you are planning your disaster recovery procedures scenario, not when you are at the recovery site.

2.6.2 REORG

One advantage of online Reorg is that it can be easily terminated and the data sets are intact for more situations than either LOAD or the other REORG options:

- REORG SHRLEVEL NONE
  - If the RELOAD phase has completed, the reorged data is on the DASD at the recovery site. You may need to rebuild indexes.
  - If the RELOAD phase has not started, then the data and indexes are still okay. You need take no further action until you want to REORG the objects.
  - However, if the RELOAD phase is in progress, the data on DASD is in an indeterminate state. In that case, you must recover to the start of the utility. You need a "before" image copy at the recovery site, and all the log that must be applied to that copy to bring it to currency.
- REORG (all other SHRLEVELs)
  - If the SWITCH phase has not occurred yet, the original data sets are intact. After you term the utility, you need take no further action.
  - If the utility is in the SWITCH phase, the original data sets remain intact. After you term the utility, you need take no further action.
  - If the SWITCH phase has completed, the reorged data sets are on DASD. If there is a BUILD2 phase, you may have to rebuild NPI Logical Partitions.

Part 2

Disaster recovery major components

In this part of the book, we include chapters where we describe the functions which constitute the major building blocks in a disaster recovery solution. We offer just enough information about the individual technologies, to allow to understand the solutions. If you already have basic knowledge of them, you can jump ahead to Part 3, “General solutions for disaster recovery” on page 129.

Part 2 contains these chapters, listed roughly in the order of decreasing RTO and RPO:

- Chapter 3, “Traditional recovery” on page 43
- Chapter 4, “DB2 Tracker” on page 57
- Chapter 5, “ESS FlashCopy” on page 67
- Chapter 6, “SMS copy pools and DB2 point in time recovery” on page 79
- Chapter 7, “Peer-to-Peer Remote Copy” on page 97
- Chapter 8, “eXtended Remote Copy” on page 113

3

Traditional recovery

In this chapter we describe the traditional procedures for DB2 subsystem back-up (dump and image copy) and recovery at a recovery site. The traditional DB2 disaster recovery restores the DB2 system from periodic dump, and recovers more current DB2 data using image copies and archive logs transferred on an ongoing basis to a recovery site.

If you transport image copies and archive logs daily to a recovery site, DB2 subsystem is typically up to 24 hours behind and requires up to 24 hours to recover. The frequency at which the data is shipped determines how much data is lost in case of disaster.

The following publications should be referenced to complement the information presented in this chapter:

- *DB2 UDB for z/OS V8 Administration Guide*, SC18-7413
- *DB2 UDB for z/OS Version 8 Utility Guide and Reference,* SC18-7427

3.1 Overview

The objective of the traditional DB2 disaster recovery is to transfer processing to a recovery site while losing the minimum amount possible of:

- Data
- Work
- Time

Various methods can be used to move the data to a recovery location. Traditionally, many organizations use physical tapes transported by road. An alternative is to transmit the data over the network. The frequency at which the data is shipped determines how much data is lost in case of disaster.

This approach typically leaves the system less than a day out of date and does not require a shutdown of the local DB2. However, restoring requires DB2 knowledge at the remote site and the approach is more difficult to synchronize with IMS and CICS.

In Figure 3-1, we consider a DFDSS dump of all DB2 related volumes that include DB2 catalog, DB2 directory, user data, BSDS, active logs, ICF catalogs, SMP data sets, and program libraries. The dump is taken *periodically (typically, once a week),* shipped to the recovery site and restored.

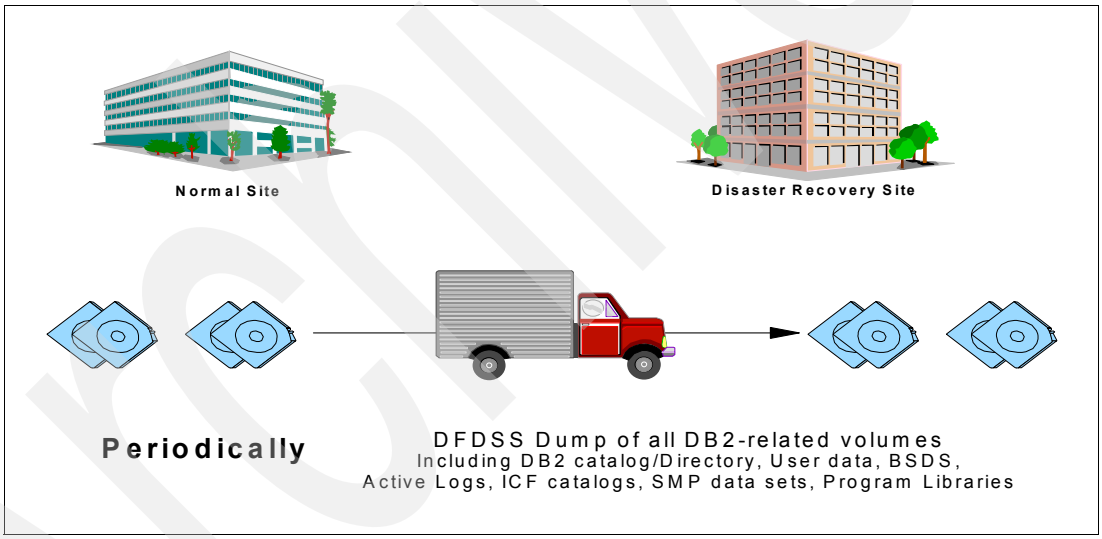


Figure 3-1 Dump and restore of DB2 environment

In Figure 3-2, on a more frequent basis, say *nightly* at the least, we additionally consider taking and shipping image copies of the DB2 catalog, the DB2 directory, user table spaces, and integrated catalog facility (ICF) catalog backup. We also force the log to archiving in order to be able to recover to the point-in-time when the archive log created.

For disaster recovery to be successful, consider taking image copies of user page sets, archive logs, and BSDS lists on *an ongoing basis*, and send to the recovery site. Additional image copies can be created anytime after creation of the original copy by using the COPYTOCOPY utility.

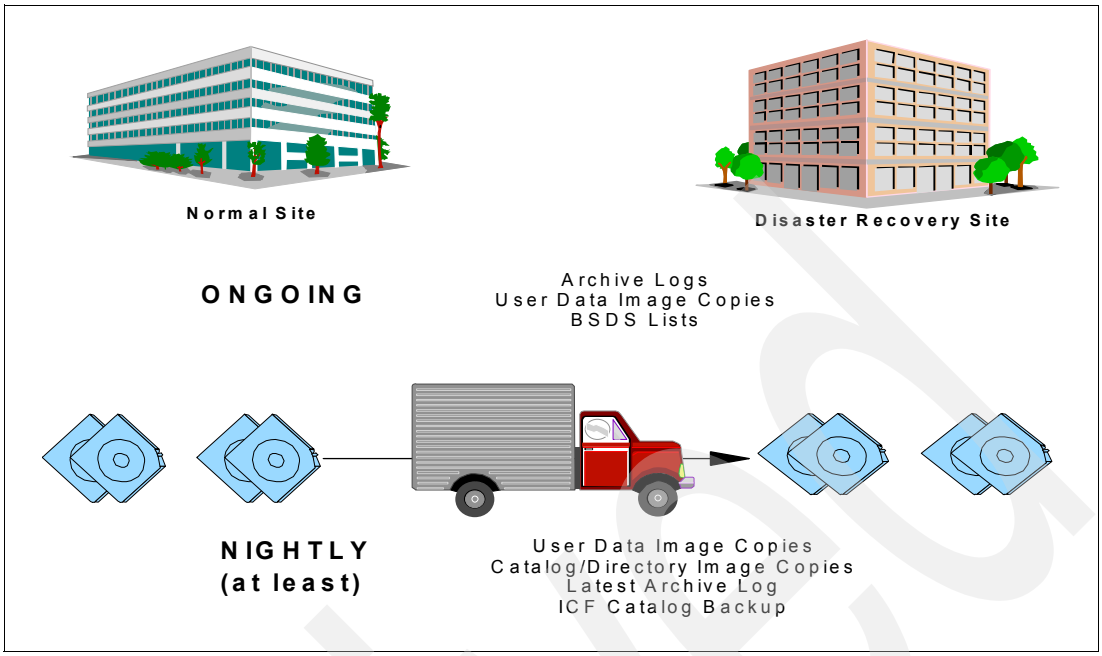


Figure 3-2 A daily business approach

With a daily business approach that includes a nightly basis and an ongoing basis (see Figure 3-2), once recovered, the data will be up to date through the last archive log sent. This scenario bases recovery on the latest available archive log and assumes that all copies and BSDS lists (print log map utility output) have arrived at the recovery site.

In this chapter, we will only discuss traditional disaster recovery for a non-data sharing environment. For a detailed description of the data sharing environment, refer to Chapter 21, “Data sharing” on page 411.

3.2 Planning for disaster recovery

There are multiple ways of preparing for disaster recovery. Independent of the way you choose, you should plan and practice your recovery procedures before the disaster strikes. Some of the popular options for disaster backup and recovery are described in the following sections.

3.2.1 Dump and restore of DB2 environment

With this option, you dump the entire DB2 environment at the local site and reestablish it at the recovery site periodically as follows:

1. Copy everything periodically with a DFSMSdss volume dump.
2. Send it to the recovery site, and then restore it there.

This is an inexpensive and easy backup option, but it can only be used to *recover to the point-in-time when the dump was taken*.

A disadvantage of this option is that, to obtain data consistency, you must bring down DB2 for the length of time it takes to dump the DB2 environment.

3.2.2 Consistent point-in-time recovery

With this approach, the DB2 environment is reestablished, whereby the DB2 Catalog, the DB2 Directory, and the user tables spaces are recovered *to the point-in-time when the ARCHIVE LOG command created the archive logs* at the recovery site.

Important: Planning the timing of backup is essential for a successful recovery. You can back up in this order:

1. Image copies of the user table spaces
2. Image copies of the DB2 catalog and DB2 directory
3. Truncate Active log
4. MVS catalog backup

To perform the backup, copy and send the following objects to the recovery site on a nightly basis, at least after dumping has been taken periodically as in the previous method (see 3.2.1, “Dump and restore of DB2 environment” on page 45):

1. Image copies of the DB2 catalog, the DB2 directory and the user table space:

   Run COPY utility with RECOVERYDDN (most case in tape).

2. Force archiving to create a consistent set of archive log and BSDS lists:

   Consider the use of the -ARCHIVE LOG command to force archiving. Consider also issuing the -SET LOG LOGLOAD(0) or -SET LOG CHKTIME(0) command before issuing the -ARCHIVE LOG command so that you have the checkpoint on the truncated active log. Although there is no need to stop DB2 or even to quiesce during this procedure, MODE(QUIESCE) has some advantages. It simplifies restart since there will be no indoubt or inflight URs. There still may be incommit URs and pending writes. It may also make it possible to establish a coordinated quiesce point in IMS or CICS.

   Once the message DSNJ139I LOG OFFLOAD TASK ENDED is issued, you can release the batch job to run your DSNJU004 report.

3. Integrated catalog facility (ICF) catalog EXPORT and list:

   Synchronize ICF catalog backup with the cataloging of image copies and archive logs.

3.2.3 Continuous archive log transfer

As for the previous method, you must take image copies of the DB2 catalog, the DB2 directory and the user table spaces and issue the ARCHIVE LOG command to create a consistent set of archive log data sets on a nightly basis at least. Thereafter, on an ongoing basis, you continuously send the image copies of the user tables spaces to the recovery site. Whenever an archive log data set is created, you also send a copy of it and BSDS list to the recovery site. In case of disaster, *the data will be recovered up to date through the last archive sent*.

In 3.3, “Recovery procedures” on page 47, we will concentrate on this approach. Potential loss of data for this method corresponds to the size of an active log data set.

With this option, in addition to copies taken in previous method (see 3.2.2, “Consistent point-in-time recovery” on page 46), copy and send the following objects to the recovery site *as you produce them* on an ongoing basis:

1. Image copies of user page sets:

   Run COPY utility with RECOVERYDDN (in most cases on tape). If you have a lot of changes of DB2 catalog and DB2 directory during day time, you should also have frequent image copies of them.

COPYTOCOPY allows you to create additional image copies any time after the original has been created.

2. Archive logs:

   You usually use dual archive logging. The second archive log (in most cases sent directly to tape via install option) is intended as a backup or can be sent to the recovery site in preparation for disaster recovery.

3. BSDS lists:

   You will use the report to determine the fully qualified name of the archive log you need to restore from and the RBA you will need to give in CRESTART statement of DSNJU003. Your system should usually schedule search for the message DSNJ139I LOG OFFLOAD TASK ENDED. Once the message is issued, you can release the batch job to run your DSNJU004 report.

3.3 Recovery procedures

The procedures in this scenario is to recover using image copies and archive logs at a remote site in the case of disaster. We will concentrate on that approach as described in 3.2.3, “Continuous archive log transfer” on page 46 in this section.

3.3.1 Establish the environment

This section shows the preparatory steps that are necessary before the actual data recovery can begin:

1. If an ICF catalog does not already exist at the recovery site, run job DSNTIJCA to create a user catalog.
2. Use the access method services IMPORT command to import the any ICF catalogs.
3. Restore DB2 libraries:

   Include DB2 reslibs, SMP libraries, user program libraries, user DBRM libraries, CLISTs, SDSNSAMP, or where the installation jobs are, JCL for user defined table spaces, etc.

4. Use IDCAMS DELETE NOSCRATCH to delete all catalog and user objects:

   Because step 2 imports a user ICF catalog, catalog reflects data sets that do not exist on disk.

5. Define required data sets:

   Use DSNTIJIN job to define required data sets (that include active logs, BSDS, DB2 catalog and DB2 directory). During this step, job DSNTIJIN should be modified. Of course, the VSAM data sets need only be defined if they have not yet been defined. Comment out the steps that create DB2 non-VSAM data sets, if these data sets already exist.

6. Change the ZPARM needed for disaster recovery:
   - The first two entries in the DSN6SPRM macros should be **DEFER** and **ALL**. This will indicate to defer recovery of all objects, including DB2 catalog objects on DB2 start up.
   - ARC2FRST should be **YES** to indicate to DB2 to read COPY2 of the archive log first.
   - SITETYP should be **RECOVERYSITE** to indicate to the RECOVER utility that it should use image copies registered to RECOVERYDDN, if this is the case.

3.3.2 Recover the BSDS

For the detailed information, refer to 3.4, “Recovering the BSDS” on page 49.

1. Review the DSNJU004 report:

   Determine the fully qualified name of the archive log you need to restore from and the RBA you will need to give in CRESTART statement of DSNJU003.

2. Copy the BSDS from the last archive log tape.
3. (Optionally) copy archive log to disk:

   Doing this eliminates multiple tape mounts and speeds up recovery.

4. Update the BSDS:
   a. Delete the active logs and add new active logs in the BSDS. Do not specify a STARTRBA or ENDRBA in NEWLOG statement.
   b. Register the last archive log data set in the archive log inventory of the BSDS just restored.
5. If you are using the DB2 distributed data facility, run the DSNJU003 to update the LOCATION and LUNAME in the BSDS.
6. Use the print log map utility (DSNJU004) to ensure that the BSDS correctly reflects the active and archive log data set inventories.
7. If you are using dual BSDSs, make a copy of the newly restored BSDS data set to the second BSDS data set.
8. Run DSN1LOGP to show any activity that was ongoing during the archive processing:
   a. Input is the latest archive log (file 2).
   b. STARTRBA is the starting RBA of the last complete checkpoint on the last archive log.

3.3.3 Restart DB2 (conditional restart)

For the detailed information, refer to 3.5, “Restart DB2 (Conditional Restart)” on page 53.

1. Use DSNJU003 to create conditional restart record:

   Specify CRESTART CREATE, ENDRBA=nnnnnnnnn000, FORWARD=YES, BACKOUT=YES. ENDRBA equals a value one more than the ENDRBA of the last archive log.

2. Run DSNJU004 to verify BSDS contents:

   Verify that the conditional restart control record that you created in the previous step is active with correct ENDRBA and checkpoint.

3. Start DB2 using ACCESS(MAINT):

   During restarting, all URs are backed out or committed except for indoubt URs and DB2 may result in resource unavailable messages, regardless of DEFER.

   Attention: Log on with INSTALL SYSADM TSO user id.

4. Recover indoubt URs:

   The recover utility will fail on any table space that has indoubt URs. Because of this, you must resolve them first.

   Check the DSN1LOGP report for any indoubt threads against the catalog and/or directory and decide if you are going to ABORT or COMMIT them. In most cases you will choose ABORT. The command to abort the thread is RECOVER INDOUBT ACTION(ABORT).

3.3.4 Recover DB2 system data

You must recover the DB2 catalog and DB2 directory before recovering user table spaces:

1. Recover DB2 system data and the order of recovery is very important:
   a. The SYSUTILX table space must be first and alone.
   b. The SYSUTILX indexes must be alone.
   c. Recover the rest of the DB2 catalog and DB2 directory objects starting with DBD01, in the order shown in the description of the RECOVER utility in Part 2 of *DB2 UDB for z/OS Version 8 Utility Guide and Reference,* SC18-7427.
2. Verify your subsystem:
   a. Display DSNDB01 and DSNDB06.
   b. Run DSN1COPY with CHECK to do a validity check on each page.
   c. Run DSN1CHKR to verify integrity of the DB2 directory and DB2 catalog table spaces.
   d. Run CHECK INDEX to verify that the DB2 directory and DB2 catalog indexes are consistent with the data they index.

3.3.5 Prepare subsystem for use

1. Define and initialize the work file database (DSNDB07).
2. Modify ZPARM to restart all databases:

   The first two entries in the DSN6SPRM macros should be **RESTART** and **ALL**.
3. Stop and start DB2.
4. Make a full image copy of the DB2 catalog and DB2 directory.
5. Recover user data.

   The recovery process for your application data should be the same as you use at your local site, except that you will be using your RECOVERYDDN data sets.
6. Make a full image copy of all table spaces and indexes with the COPY YES.
7. Reprocess lost work.

   Rerun online transactions and batch jobs since the last archive log was created.

3.4 Recovering the BSDS

In this section we consider how to recover the BSDS.

Review DSNJU004 report

Example 3-1 lists the RBA and the name of the archive log you will need to register in the DSNJU003 job. It was arrived from the local site, as the archive log file had been generated.

Example 3-1 Determine the name and RBA of archive log

```
ARCHIVE LOG COPY 1 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
   000080A3A000           000082BF9FFF       2004.211  13:46 DSN=DB8XU.ARCHLOG1.D2004211.T1346385.A0000064
 2004.210  21:53:26.5  2004.211  17:46:37.9                   PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                              CATALOGUED
   000082BFA000           000084DB9FFF       2004.211  13:49 DSN=DB8XU.ARCHLOG1.D2004211.T1349081.A0000065
```

```
 2004.211  17:46:37.9  2004.211  17:49:08.0                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
   000084DBA000           000086F79FFF       2004.211  13:51 DSN=DB8XU.ARCHLOG1.D2004211.T1351385.A0000066
 2004.211  17:49:08.0  2004.211  17:51:38.4                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED

ARCHIVE LOG COPY 2 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
  000080A3A000           000082BF9FFF       2004.211  13:46 DSN=DB8XU.ARCHLOG2.D2004211.T1346385.A0000064
2004.210  21:53:26.5  2004.211  17:46:37.9                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
  000082BFA000           000084DB9FFF       2004.211  13:49 DSN=DB8XU.ARCHLOG2.D2004211.T1349081.A0000065
2004.211  17:46:37.9  2004.211  17:49:08.0                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
  000084DBA000           000086F79FFF       2004.211  13:51 DSN=DB8XU.ARCHLOG2.D2004211.T1351385.A0000066
2004.211  17:49:08.0  2004.211  17:51:38.4                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
```

Repro the BSDS from the last archive log tape

in Example 3-2, we restore BSDS from the last archive log file.

Example 3-2 Repro BSDS from the last archive log tape

```
//REPRO    EXEC PGM=IDCAMS
//INPUT    DD   DISP=OLD,DSN=DB8XU.ARCHLOG2.D2004211.T1351385.B0000066,
//              UNIT=3490,VOL=SER=PRB000,LABEL=1
//OUTPUT   DD   DISP=OLD,DSN=DB8XU.BSDS01
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 REPRO INFILE(INPUT) OUTFILE(OUTPUT) REUSE
```

Delete the active logs and add new active logs in BSDS

The BSDS contains information about the active logs at your local site. You want to delete the old active logs and add the new active logs as shown in Example 3-3.

Example 3-3 Register the active logs in BSDS

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
   DELETE DSNAME=DB8XU.LOGCOPY1.DS01
   DELETE DSNAME=DB8XU.LOGCOPY1.DS02
   DELETE DSNAME=DB8XU.LOGCOPY1.DS03
   DELETE DSNAME=DB8XU.LOGCOPY2.DS01
   DELETE DSNAME=DB8XU.LOGCOPY2.DS02
   DELETE DSNAME=DB8XU.LOGCOPY2.DS03
   NEWLOG DSNAME=DB8XU.LOGCOPY1.DS01,COPY1
   NEWLOG DSNAME=DB8XU.LOGCOPY1.DS02,COPY1
   NEWLOG DSNAME=DB8XU.LOGCOPY1.DS03,COPY1
   NEWLOG DSNAME=DB8XU.LOGCOPY2.DS01,COPY2
   NEWLOG DSNAME=DB8XU.LOGCOPY2.DS02,COPY2
   NEWLOG DSNAME=DB8XU.LOGCOPY2.DS03,COPY2
```

Register the last archive log data set in BSDS

The assumption is that we have restored all the second archive logs to disk. It depends on your site.

The BSDS that you restored from the last archive log does not contains information about it. You should register the last archive log in BSDS as shown in Example 3-4.

Example 3-4 Register the last archive log in BSDS

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 NEWLOG DSNAME=DB8XU.ARCHLOG2.D2004211.T1351385.A0000066,
 COPY2VOL=DB8X02,
 STARTRBA=000084DBA000,ENDRBA=000086F79FFF,UNIT=3390,CATALOG=YES
//*
```

Run the print log map utility (DSNJU004)

You can run DSNJU004 to ensure that the BSDS correctly reflects the active and archive log data set inventories as shown in Example 3-5.

Example 3-5 DSNJU004 sample output

```
DSNJCNVB CONVERSION PROGRAM HAS NOT RUN   DDNAME=SYSUT1
   LOG MAP OF BSDS DATA SET COPY 1, DSN=DB8XU.BSDS01
   LTIME INDICATES LOCAL TIME, ALL OTHER TIMES ARE GMT.
         DATA SHARING MODE IS OFF
         SYSTEM TIMESTAMP   - DATE=2004.211  LTIME=13:51:22.91
         UTILITY TIMESTAMP  - DATE=2004.211  LTIME=14:19:50.56
         VSAM CATALOG NAME=DB8XD
         HIGHEST RBA WRITTEN       000086F780D2  2004.211  17:51:38.4
         HIGHEST RBA OFFLOADED     000084DB9FFF
         RBA WHEN CONVERTED TO V4  000000000000

ACTIVE LOG COPY 1 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
                EMPTY DATA SET              2004.211  14:18 DSN=DB8XU.LOGCOPY1.DS01
 0000.000  00:00:00.0  0000.000  00:00:00.0  PASSWORD=(NULL) STATUS=NEW, REUSABLE
                EMPTY DATA SET              2004.211  14:18 DSN=DB8XU.LOGCOPY1.DS02
 0000.000  00:00:00.0  0000.000  00:00:00.0  PASSWORD=(NULL) STATUS=NEW, REUSABLE
                EMPTY DATA SET              2004.211  14:18 DSN=DB8XU.LOGCOPY1.DS03
 0000.000  00:00:00.0  ........  ..........  PASSWORD=(NULL) STATUS=NEW, REUSABLE
ARCHIVE LOG COPY 1 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
   000080A3A000          000082BF9FFF       2004.211  13:46 DSN=DB8XU.ARCHLOG1.D2004211.T1346385.A0000064
 2004.210  21:53:26.5  2004.211  17:46:37.9                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
   000082BFA000          000084DB9FFF      2004.211  13:49 DSN=DB8XU.ARCHLOG1.D2004211.T1349081.A0000065
 2004.211  17:46:37.9  2004.211  17:49:08.0                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
```

```
ARCHIVE LOG COPY 2 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
  000080A3A000           000082BF9FFF       2004.211  13:46 DSN=DB8XU.ARCHLOG2.D2004211.T1346385.A0000064
2004.210  21:53:26.5  2004.211  17:46:37.9                   PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                            CATALOGUED
  000082BFA000           000084DB9FFF       2004.211  13:49 DSN=DB8XU.ARCHLOG2.D2004211.T1349081.A0000065
2004.211  17:46:37.9  2004.211  17:49:08.0                   PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                            CATALOGUED
  000084DBA000           000086F79FFF       2004.211  14:19 DSN=DB8XU.ARCHLOG2.D2004211.T1351385.A0000066
0000.000  00:00:00.0  0000.000  00:00:00.0                   PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                            CATALOGUED

                        CHECKPOINT QUEUE
                    18:20:51 JULY 29, 2004
     TIME OF CHECKPOINT       17:51:38 JULY 29, 2004
     BEGIN CHECKPOINT RBA            000086F83000
     END CHECKPOINT RBA              000086F902FA
     TIME OF CHECKPOINT       17:49:08 JULY 29, 2004
     BEGIN CHECKPOINT RBA            000084DC8187
     END CHECKPOINT RBA              000084DD53A6
     TIME OF CHECKPOINT       17:46:38 JULY 29, 2004
     BEGIN CHECKPOINT RBA            000082C0A4A3
     END CHECKPOINT RBA              000082C215AA
     TIME OF CHECKPOINT       22:26:53 JULY 28, 2004
     BEGIN CHECKPOINT RBA            000080A45090
     END CHECKPOINT RBA              000080A475B2
```

Run DSN1LOGP

The STARTRBA to be used with DSN1LOGP for summary report should represent the latest checkpoint on the archive log volume. This corresponds to the second entry in the checkpoint queue as shown in Example 3-5.

You should run DSN1LOGP with RBASTART (xxxxxx) SUMMARY(ONLY), where xxxxxx is the RBA of the latest checkpoint BEGIN on the archive log volume, as shown in Example 3-6.

If utilities are inflight, special recovery is required later, and if URs are indoubt, note the modified page sets.

Example 3-6 DSN1LOGP sample JCL

```
//BSDS01   EXEC PGM=DSN1LOGP
//ARCHIVE  DD   DISP=OLD,DSN=DB8XU.ARCHLOG2.D2004211.T1351385.A0000066
//SYSPRINT DD   SYSOUT=*
//SYSSUMRY DD   SYSOUT=*
//SYSIN    DD   *
 RBASTART (000084DC8187) SUMMARY(ONLY)
/*
```

3.5 Restart DB2 (Conditional Restart)

In this section we discuss how to do a conditional restart of DB2.

Run DSNJU003 to create conditional restart record

Example 3-7 creates a *conditional restart control record* (CRCR) with an ENDRBA equal to ENDRBA of the last archive (you can see it on Example 3-5) +1.

Example 3-7 Create a CRCR

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DB8XU.BSDS02
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 CRESTART CREATE,ENDRBA=000086F7A000,FORWARD=YES,BACKOUT=YES
//*
```

Run DSNJU004 to verify BSDS contents

You should verify that the CRCR that you created in the previous step is active with correct ENDRBA and checkpoint as shown in Example 3-8.

Example 3-8 Verify a CRCR

```
CONDITIONAL RESTART CONTROL RECORD
                    18:35:50 JULY 29, 2004
**** ACTIVE CRCR RECORD ****
CRCR IDENTIFIER  0001
     USE COUNT   0
     RECORD STATUS
         CRCR ACTIVE
         CRCR NOT USED
     PROCESSING STATUS
         FORWARD = YES
         BACKOUT = YES
     STARTRBA                        NOT SPECIFIED
     ENDRBA                          000086F7A000
     ENDLRSN                         NOT SPECIFIED
     EARLIEST REQUESTED RBA          000000000000
     FIRST LOG RECORD RBA            000000000000
     ORIGINAL CHECKPOINT RBA         000000000000
     NEW CHECKPOINT RBA (CHKPTRBA)   000084DC8187
     END CHECKPOINT RBA              000084DD53A6
     CRCR CREATED             18:35:42 JULY 29, 2004
     TIME OF CHECKPOINT       17:49:08 JULY 29, 2004
     RESTART PROGRESS                 STARTED     ENDED
                                      =======     =====
         CURRENT STATUS REBUILD          NO         NO
         FORWARD RECOVERY PHASE          NO         NO
         BACKOUT RECOVERY PHASE          NO         NO
```

Start DB2 using ACCESS(MAINT)

You can start DB2 with ACCESS(MAINT) to prevent access by users until recovery is complete. During restarting, inflight and inabort URs will be backed out, incommit will be committed, and indoubts will be noted on the console log. The deferred page sets with pending writes or UR activity will be in LPL and other deferred page sets will be in RW state.

In Example 3-9, message DSNT501I indicates system page sets marked LPL or STOPP, and the exceptions should be resolved in the next step.

Example 3-9 Conditional Restart

```
$HASP373 DB8XMSTR STARTED
IEF403I DB8XMSTR - STARTED - TIME=18.22.17 - ASID=040D - SC64
DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE NAME IS DSNZPARM
S DB8XIRLM
DSNY001I  -DB8X SUBSYSTEM STARTING
DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.211 13:51:22.91
385 DSNJ245I  -DB8X CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT RBA 000086F7A000.
REPLY Y TO CONTINUE, N TO CANCEL
R 385,Y
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  465
DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS01,
STARTRBA=000086F7A000,ENDRBA=000089139FFF
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  466
DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS01,
STARTRBA=000086F7A000,ENDRBA=000089139FFF
DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  467
STARTRBA=000086F7A000
S DB8XDBM1
S DB8XDIST
DSNR001I  -DB8X RESTART INITIATED
DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=000084DC8187
DSNR004I  -DB8X RESTART...UR STATUS COUNTS  480
IN COMMIT=0, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
DSNR007I  -DB8X RESTART...STATUS TABLE  481
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
B UTILITY  PAOLOR3L     PAOLOR3  DSNUTIL  F 000086AA7995 211 13:51:17
DSNI001I  -DB8X RESTART HAS BEEN DEFERRED  482
          REASON 00C90095
          TYPE 00000200
          NAME PAOLODB .TSLINEI
DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  483
THE LOGICAL PAGE LIST
          DATABASE NAME=PAOLODB
          SPACE NAME=TSLINEI
          DATA SET NUMBER=1
          PAGE RANGE X'00000000' TO X'FFFFFFFF'
          START LRSN=X'000082C0B104'
          END LRSN=X'000086F79FFF'
          START RBA=X'000082C0B104'
          LPL TRACE ID=00000001
          LPL REASON TYPE=LOGAPPLY

DSNI001I  -DB8X RESTART HAS BEEN DEFERRED  484
          REASON 00C90095
          TYPE 00000200
          NAME DSNDB01 .SYSUTILX
DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  485
THE LOGICAL PAGE LIST
```

```
           DATABASE NAME=DSNDB01
           SPACE NAME=SYSUTILX
           DATA SET NUMBER=1
           PAGE RANGE X'00000000' TO X'FFFFFFFF'
           START LRSN=X'000082D8BC4C'
           END LRSN=X'000086F79FFF'
           START RBA=X'000082D8BC4C'
           LPL TRACE ID=00000001
           LPL REASON TYPE=LOGAPPLY

DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  486
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  487
THE LOGICAL PAGE LIST
           DATABASE NAME=PAOLODB
           SPACE NAME=TSLINEI
           DATA SET NUMBER=1
           PAGE RANGE X'00000000' TO X'FFFFFFFF'
           START LRSN=X'000086F48071'
           END LRSN=X'000000000000'
           START RBA=X'000086F48071'
           LPL TRACE ID=00000001
           LPL REASON TYPE=LOGAPPLY

DSNR018I  -DB8X RESTART...BACKWARD RECOVERY PROCESSED  488
 FROM RBA 000086F780D2 TO RBA 000086AA7995
DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  489
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  490
THE LOGICAL PAGE LIST
           DATABASE NAME=PAOLODB
           SPACE NAME=TSLINEI
           DATA SET NUMBER=1
           PAGE RANGE X'00000000' TO X'FFFFFFFF'
           START LRSN=X'000086F7AB0E'
           END LRSN=X'000086F7AB0E'
           START RBA=X'000086F7AB0E'
           LPL TRACE ID=00000001
           LPL REASON TYPE=LOGAPPLY

DSNR002I  -DB8X RESTART COMPLETED
DSNY014I  -DB8X DSNYSTRT DB2 WAS STARTED WITH ACCESS(MAINT)
-DB8XRECOVER POSTPONED
DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
DSNT500I  -DB8X DSNGEDLC RESOURCE UNAVAILABLE  497
           REASON 00C900A6
           TYPE 00000100
           NAME DSNDB06
DSNT501I  -DB8X DSNICUBC RESOURCE UNAVAILABLE  498
           CORRELATION-ID=014.RTSTST00
           CONNECTION-ID=DB8X
           LUW-ID=DB8X.SCPDB8X.BB9730E857B9=0
           REASON 00C900A6
           TYPE 00000100
           NAME DSNDB06
DSN9022I  -DB8X DSNYASCP 'STA DB2' NORMAL COMPLETION
```

Restricted status after conditional restart

Example 3-10 lists DB2 system table spaces in restricted status after DB2 conditional restart. You should recover them first.

Example 3-10 -DIS DATABASE(*) SPACENAM(*) RESTRICT

```
DSNT360I  -DB8X ***********************************
DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                *    RESTRICTED
DSNT360I  -DB8X ***********************************
DSNT362I  -DB8X     DATABASE = DSNDB01  STATUS = RW
                   DBD LENGTH = 14200
DSNT397I  -DB8X
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
SYSUTILX TS         RW,LPL
******* DISPLAY OF DATABASE DSNDB01  ENDED      **********************
DSNT317I  -DB8X DSNTDMST ANY USER-DEFINED INDEXES ON DSNDB06
 COULD NOT BE DISPLAYED DUE TO A RESOURCE UNAVAILABILITY
 ON DSNDB01 OR DSNDB06
DSNT500I  -DB8X DSNTDMI RESOURCE UNAVAILABLE
           REASON 00C900A6
           TYPE 00000100
           NAME DSNDB06
DSNT500I  -DB8X DSNTDMI RESOURCE UNAVAILABLE
           REASON 00C900A6
           TYPE 00000100
           NAME DSNDB06
DSN9023I  -DB8X DSNTDDIS 'DISPLAY DATABASE' ABNORMAL COMPLETION
```

You cannot display utilities in progress until you recover the SYSUTILX table space as shown in Example 3-11.

Example 3-11 -DIS UTIL(*)

```
DSNU104I  -DB8X DSNUGDIS - DSNDB01.SYSUTILX IN A PENDING STATE
PROHIBITS FURTHER PROCESSING
DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

3.6 Considerations

The solution we discussed here just addressed a DB2 subsystem. If you have CICS and/or IMS installed as transaction managers, you need to consider the challenges associated with keeping resources from all the subsystems in sync. Because of difficulties in coordinating across DB2 and CICS or IMS, other more general solutions may be preferable.

4

DB2 Tracker

In this chapter we provide a description of the DB2 Tracker function. Tracker is an active DB2 subsystem at the recovery site which can keep shadow copies of the local site data close to current. It can take over in case of disaster.

Tracker Site recovery can be recovered sooner than the Traditional Disaster Recovery (DR) solution shown in Chapter 3, “Traditional recovery” on page 43, if logs are sent multiple times daily. Its currency is better than the pickup truck access method (PTAM) and the same as the traditional DR.

This chapter covers the following topics:

- Overview
- Tracker site recovery
- Considerations

4.1 Overview

A DB2 tracker site is a separate DB2 subsystem, or data sharing group, that exists solely for the purpose of keeping shadow copies of your local site's data. No independent work can be run on the tracker site. From the local site, you transfer the BSDS and the archive logs, and the tracker site runs periodic LOGONLY recoveries to keep the shadow data up-to-date.

If a disaster occurs at the local site, the tracker site becomes the takeover site. Because the tracker site has been shadowing the activity on the local site, you do not have to constantly ship image copies. The takeover time for the tracker site might be faster because DB2 recovery does not have to use image copies.

RESTORE SYSTEM LOGONLY can be used on the tracker site to establish a recovery cycle. In case you use the RESTORE SYSTEM LOGONLY instead of RECOVER utility at the tracker site, the advantages are as follows:

- It eliminates complex procedures because it is not necessary to recover all database objects independently.
- It provides better performance because it will only scan and process logs in one pass for both system and user objects, and the log apply will be also done in parallel.

DB2 Tracker is designed to address issues related to the speed at which the data can be recovered. The data can be recovered up to the last log that is secured away from the local location. An organization using the standard disaster recovery procedure described in Chapter 3, "Traditional recovery" on page 43, will be able to recover to exactly the same point with the DB2 Tracker option. How the data is shipped is not important to the use of DB2 Tracker. This means that DB2 Tracker can be implemented without changing existing procedures for shipping data.

In Figure 4-1, the DBP1 at the local site has archives and image copies taken off site to a second active site and the subsystem DBP1', a clone of DBP1, is created from them. While this figure shows a single DB2 subsystem, data sharing is also supported.

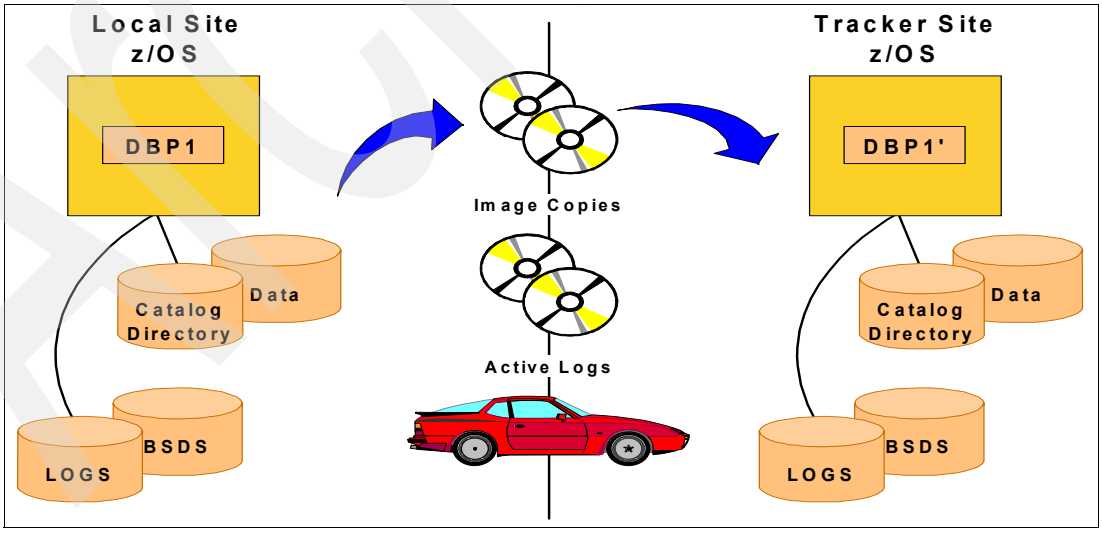


Figure 4-1 Overview of Local and Tracker Site

For more information, see "Using a tracker site for disaster recovery" in the *DB2 Administration Guide,* SC18-7413.

4.2 Tracker site recovery

The recovery time at the DB2 tracker site will be reduced depending on your log shipping frequency, the log apply frequency, the number of DB2 objects, and the amount of log to apply (transaction updates). Indexes can be rebuilt, or recovered, depending on your choice of taking image copies of them.

You can use all the procedures you previously set up for disaster recovery (refer to Chapter 3., “Traditional recovery” on page 43). They are exactly the same to set up the site as a “mirror” of the local site, but there are two exceptions:

- The CRESTART does not allow either forward or backward recovery at restart.
- The TRKRSITE parameter in DSN6SPRM in DSNZPARM must be set to YES.

4.2.1 Setting up the tracker site

The starting point is a mirror image of your local DB2 subsystem (or data sharing group) at the tracker site prepared for recovery, by specifying the DSNZPARM parameter TRKSITE=YES in DSN6SPRM. Optionally, you can also specify the DSNZPARM parameter SITETYP=RECOVERYSITE if the image copies that were received at the tracker site were created at the local site as remote site copies.

The BACKUP SYSTEM utility supports to back up an entire DB2 subsystem or an entire DB2 data sharing group with a single command. It provides a fast, easy-to-use volume level backup and recovery solution.

To use the BACKUP SYSTEM and RESTORE SYSTEM utilities that are new with DB2 for z/OS V8, all data sets that you want to back up and recover must be SMS-managed data sets. Additionally, you must have the following requirements:

- z/OS V1R5 or above
- Disk control units that support ESS FlashCopy ® API (FlashCopy V2 is recommended)
- SMS copy pools that are defined by using the DB2 naming convention
- Defined SMS copy pool backup storage groups
- DB2 has to be running in new-function mode with DB2 for z/OS V8.

It is worth noting that you can use RESTORE SYSTEM *LOGONLY* without z/OS V1.5 or disk control units that support ESS FlashCopy ® API.

Preparation steps at the local site

To prepare the tracker site:

1. BACKUP SYSTEM utility or FlashCopy using SET LOG SUSPEND / SET LOG RESUME:
   - Use the FlashCopy function to copy all DB2 volumes (User DB, System DB, Active Logs, Archive Logs, BSDS, Product Libraries, ICF catalogs).
   - Dump FlashCopy target volumes to tapes.
   - Send tapes to the tracker site.

   For more information on traditional disaster backup, refer to 3.2, “Planning for disaster recovery” on page 45.
2. Periodically send Active Logs, Archive Logs, and BSDS to the tracker site.

   It is possible to do this by means of PPRC, XRC, FTP, or tapes.

3. Send image copies to the Tracker site for the objects involved in the following situations:
   - An event for which no logging has occurred against a given table space (REORG with LOG NO, LOAD with LOG NO, REPAIR with LOG NO)
   - A point-in-time recovery for some page sets has occurred in the active system.
   - SYSUTILX If you use the RECOVER utility in the recovery cycles (refer to “Using the RECOVER utility” on page 61). Send a full image copy of DSNDB01.SYSUTILX for normal (full image copy and log) recoveries. For LOGONLY recoveries, create a DSN1COPY of DSNDB01.SYSUTILX.

Preparation steps at the tracker site

To set up the tracker site:

1. Establish the environment:

   For more information on establishing an environment of traditional disaster recovery, refer to 3.3.1, “Establish the environment” on page 47.
2. Restore data and logs from tapes.
3. Modify the subsystem parameters. If using data sharing, update it for all members:
   - TRKSITE=YES
   - Optionally, SITETYP=RECOVERYSITE to indicate to the RECOVER utility that it should use image copies registered to RECOVERYDDN, if this is the case.

4.2.2 Establishing recovery cycle at the tracker site

Each time you restore the logs and BSDS from the local site at your tracker site, you establish a new recovery cycle as shown in the following procedure.

Using RESTORE SYSTEM LOGONLY

In this section, we only mention the recovery cycle using the new RESTORE SYSTEM LOGONLY function available with DB2 for z/OS V8. In case you use the RESTORE SYSTEM LOGONLY at the tracker site, it updates the HPGRBLP in the DBD01. So it knows where to start the next recovery cycle.

To establish recovery cycle using RESTORE SYSTEM LOGONLY at the tracker site:

1. Recover BSDS. If using data sharing, the BSDS data sets on each data sharing member need to be recovered as follows:
   a. Use REPRO command to restore BSDS from the latest archive log at the tracker.
   b. Use the change log inventory utility (DSNJU003) to register this latest archive log in the archive log inventory of the BSDS just restored.

      If using data sharing: In the NEWLOG statement for the Change Log Inventory utility, you must provide its starting (STARTRBA) and ending (ENDRBA) RBAs as well as its starting (STARTLRSN) and ending (ENDLRSN) LRSNs.
   c. Delete the active logs and add new empty active logs using DSNJU003 in the new BSDS.
   d. Use the print log map utility (DSNJU004) to ensure that the BSDS is correctly restored.
   e. If you are using dual BSDSs, make a copy of the newly restored BSDS data set to the second BSDS.

2. Use DSNJU003 to create a Conditional Restart Control Record (CRCR):
   - CRESTART CREATE, ENDRBA=*nnnnnnnnn000,* FORWARD=NO, BACKOUT=NO where nnnnnnnnn000 equals ENDRBA+1 of the latest archive log.
   - If using data sharing: Create a CRCR record for each active member, as follows:

     CRESTART CREATE, ENDLRSN=nnnnnnnnnnnn, FORWARD=NO, BACKOUT=NO where nnnnnnnnnnnn is the lowest LRSN of all the members to be read during restart. To determine the ENDLRSN, refer to 21.6.2, “Determine ENDLRSN for the conditional restart record” on page 423.
   - The ENDLRSN or ENDRBA value indicates the end log point for data recovery and for truncating the archive log. With ENDLRSN, the missing log records between the lowest and highest ENDLRSN values for all the members are applied during the next recovery cycle.
3. If you used DSN1COPY to create a copy of SYSUTILX during the last tracker site recovery cycle, restore this copy with DSN1COPY. This step is required only when *not* using the RESTORE SYSTEM.
4. Start the tracker DB2.

   If using data sharing, delete all CF structures and restart all members.
5. Execute the RESTORE SYSTEM LOGONLY utility to recover to the current time (or to the time of the last log transmission from the local site).

   If using data sharing, the utility only needs to be executed on one member.

Attention: The RESTORE SYSTEM LOGONLY is a better way to support DB2 tracker site recovery. Instead of rolling logs forward using RECOVER LOGONLY at table space or index (for COPY YES indexes) level from each HPGRBRBA, the RESTORE SYSTEM LOGONLY can roll logs forward at the system level from the system HPGRBLP value in DBD01. This means that the tracker site recovery procedure becomes relatively simple with DB2 for z/OS V8 RESTORE SYSTEM LOGONLY.

6. If the RESTORE SYSTEM utility issues a return code of 4, use DSN1COPY to make a copy of SYSUTILX and indexes that are associated with SYSUTILX before you recover or rebuild those objects. DSN1COPY issues a return code of 4 if applying the log marks to one or more DB2 objects as RECP or RBDP.
7. Restart DB2 at the tracker site.
8. Recover all objects in RECP or RBDP status:
   - Use image copies to recover objects that are in recover pending state.
   - Rebuild indexes that are in rebuild pending state.
9. Stop the tracker DB2.

Using the RECOVER utility

In this section, we only mention the recovery cycle using the RECOVER utility. In case you are using the RECOVER utility at the tracker site, the ENDRBA / ENDLRSN you specified at restart is placed in HPGRBRBA at the end of each RECOVERY LOGONLY. Normally it would be the RBA / LRSN at the end of the recovery job. It is this which allows the tracker site to continuously perform LOGONLY recovery without missing any logs.

To establish recovery cycle using RECOVER utility at the tracker site:

1. Recover BSDS.

If using data sharing, the BSDS data sets on each data sharing member need to be recovered as follows:

a. Use REPRO command to restore BSDS from the latest archive log at the tracker.

b. Use the change log inventory utility (DSNJU003) to register this latest archive log in the archive log inventory of the BSDS just restored.

   If using data sharing, you must provide its starting (STARTRBA) and ending RBAs (ENDRBA) and its starting (STARTLRSN) and ending LRSNs (ENDLRSN) in the NEWLOG statement for the Change Log Inventory utility.

c. Delete the active logs and add new empty active logs using DSNJU003 in the new BSDS.

d. Use the print log map utility (DSNJU004) to ensure that the BSDS is correctly restored.

e. If you are using dual BSDSs, make a copy of the newly restored BSDS data set to the second BSDS.

2. Use DSNJU003 to create a Conditional Restart Control Record (CRCR).

   – CRESTART CREATE, ENDRBA=*nnnnnnnnn000,* FORWARD=NO, BACKOUT=NO where nnnnnnnnn000 equals ENDRBA+1 of the latest archive log.

   – If using data sharing, create a CRCR record for each active member:

     CRESTART CREATE, ENDLRSN=nnnnnnnnnnnn, FORWARD=NO, BACKOUT=NO where nnnnnnnnnnnn is the lowest LRSN of all the members to be read during restart. To determine the ENDLRSN, refer to 21.6.2, “Determine ENDLRSN for the conditional restart record” on page 423.

   – The ENDLRSN or ENDRBA value indicates the end log point for data recovery and for truncating the archive log. With ENDLRSN, the missing log records between the lowest and highest ENDLRSN values for all the members are applied during the next recovery cycle.

3. Start the tracker DB2:

   If using data sharing, delete all CF structures and restart all members.

4. Run the RECOVER utility to recover database objects. Run the RECOVER LOGONLY on all database objects that do not require recovery from an image copy. Note that you must recover all database objects independently, as in the following procedure:

   a. Restore the full image copy or DSN1COPY of SYSUTILX.

      If you are doing a LOGONLY recovery on SYSUTILX from a previous DSN1COPY backup, make another DSN1COPY copy of that table space after the LOGONLY recovery is complete and before recovering any other catalog or directory objects.

   b. Recover the catalog and directory.

      See *DB2 UDB for z/OS Version 8 Utility Guide and Reference*, SC18-7427, for information about the order of recovery for the catalog and directory objects.

   c. Recover user data.

5. Enter DIS UTIL(*) and perform the following actions for objects on which utilities are pending. Restrictions apply to these objects with because DB2 prevents you from using the TERM UTIIL command to remove pending statuses at a tracker site.

   – If a LOAD, REORG, REPAIR, or COPY is in progress on any catalog or directory object at the local site, shut down DB2. You cannot continue recovering through the list of catalog and directory objects. Therefore, you cannot recover any user data. At the next recovery cycle, send a full image copy of the object from the local site. At the tracker site, use the RECOVER utility to restore the object.

- If a LOAD, REORG, REPAIR, or COPY utility is in progress on any user data, at the next recovery cycle, send a full image copy of the object from the local site. At the tracker site, use the RECOVER utility to restore the object.
- If an object is in the restart pending state, use LOGONLY recovery to recover the object when that object is no longer in restart pending state.

6. Stop the tracker DB2.

4.2.3 How to take over in case of disaster

If a disaster occurs at the local site, the tracker site must become the takeover site. After the takeover site restarted, you must apply log data or image copies that were enroute when the disaster occurred.

Recovering at a tracker site that uses RESTORE SYSTEM LOGONLY

If you use RESTORE SYSTEM LOGONLY in the recovery cycles at the tracker site, use the following procedure after a disaster to make the tracker site the takeover site:

1. Process the last tracker cycle (refer to “Using RESTORE SYSTEM LOGONLY” on page 60) if there are additional logs.
2. Use DSNJU003 to create a CRCR record:
   - CRESTART CREATE, ENDRBA=*nnnnnnnnn000,* FORWARD=YES, BACKOUT=YES where nnnnnnnnn000 equals ENDRBA used on the last tracker cycle.
   - If using data sharing, create a CRCR record for each active member (CRESTART CREATE, ENDLRSN=nnnnnnnnnnnn, FORWARD=YES, BACKOUT=YES where nnnnnnnnnnnn equals ENDLRSN used on the last tracker cycle).
3. Update system parameter as a non-tracker (TRKRSITE=NO). If using data sharing, update it for all members.
4. Start the tracker DB2:

   If using data sharing, delete all CF structures and restart all members.
5. If you used DSN1COPY to create a copy of SYSUTILX in the last recovery cycle, use DSN1COPY to restore that copy.
6. Recover all objects in restricted status:
   - Display and terminate any in-progress utilities and recover objects.
   - Rebuild indexes, including IBM and user-defined indexes on the DB2 catalog and user-defined indexes on table spaces.
7. This is now the local DB2.
8. ZPARMs should be changed from SITETYP=RECOVERYSITE back to LOCALSITE.

Recovering at a tracker site that uses the RECOVER utility

If you use RECOVER utility in the recovery cycles at the tracker site, use the following procedure after a disaster to make the tracker site the takeover site:

1. Restore the BSDS and register the archive log from the last archive you received from the local site.
2. Use DSNJU003 to create a CRCR record.
   - CRESTART CREATE, ENDRBA=*nnnnnnnnn000,* FORWARD=YES, BACKOUT=YES where nnnnnnnnn000 equals ENDRBA used on the last tracker cycle if you received no more logs from the primary site, or is the ENDRBA + 1 of the latest archive log if you received more logs from the local site.

- If using data sharing, create a CRCR record for each active member (CRESTART CREATE, ENDLRSN=nnnnnnnnnnnn, FORWARD=YES, BACKOUT=YES where nnnnnnnnnnnn is the LRSN of the last log record to be used during restart. To determine the ENDLRSN, refer to 21.6.2, “Determine ENDLRSN for the conditional restart record” on page 423).

3. Update the system parameters. If using data sharing, update it for all members:
   - DEFER ALL
   - TRKRSITE=NO
4. Start the tracker DB2:
   - If using data sharing, delete all CF structures and restart all members.
   - After restarting DB2, run your recoveries as you did during recovery cycles if you received more logs from the local site.
5. Recover SYSUTILX from an image copy from the local site, or from a previous DSN1COPY taken at the tracker site.
6. Recover all objects in restricted status:
   - Display and terminate any in-progress utilities.
   - Recover objects either with the LOGONLY option or image copies.
   - Rebuild indexes, including IBM and user-defined indexes on the DB2 catalog and user-defined indexes on table spaces.
7. This is now the local DB2.
8. ZPARMs should be changed from SITETYP=RECOVERYSITE back to LOCALSITE.

4.3 Considerations

Since data can continually be recovered, the currency of the tracker site can be greater than that which would be obtained by the PTAM if multiple runs of the day are made to take tapes off-site. It depends on how often you ship logs, how much log must be traversed in a recovery cycle, and how many data sets you wish to shadow.

4.3.1 Restrictions on tracker site

Because the tracker site must use only the local site’s logs for recovery, you must not update the catalog and directory or the data at the tracker site. The DB2 subsystem at the tracker site disallows local updates.

- The only utilities permitted are:
  - REPORT
  - RECOVER INDEX
  - RECOVER TABLESPACE (without any point-in-time option)
  - REBUILD
  - RESTORE SYSTEM LOGONLY
  - DIAGNOSE
- You must keep objects consistent with log data:
  - No DDL, DCL, DML except SELECT
  - No Archive
  - Cannot -TERM most utilities if in progress.
- Downlevel detection is disabled (cannot write to SYSLGRNX).

4.3.2 Pros and cons

Some advantages for tracker site recovery are as follows:

- Data sets can be recovered sooner than with the traditional disaster recovery described in Chapter 3, “Traditional recovery” on page 43.
- It avoids extra bandwidth costs of *mirroring* or shipping of log record (RRDF).
- Fewer image copy tapes need to be transported.
- The recovery is relatively simple with the new RESTORE SYSTEM LOGONLY utility available with DB2 for z/OS V8.
- It is also applicable to data sharing.

The disadvantages are as follows:

- A second site with production configuration is required.
- The tracker site DB2 subsystem is unavailable for any other use.
- It is complex and needs greater people cost to develop, test, and maintain.

5

ESS FlashCopy

In this chapter we briefly introduce the FlashCopy function provided by the ESS family of Enterprise Storage Servers. FlashCopy is an optional feature that must be enabled in the ESS.

FlashCopy creates a physical point-in-time copy of the data, with minimal interruption to applications, and makes it possible to access both the source and target copies immediately. FlashCopy may also be used in conjunction with either the local or remote copies of data created by PPRC and XRC, making it easy to create additional copies for rapid recovery following application errors or for other uses. This provides a more effective way of producing volume dumps.

This chapter covers the following topics:

- Introducing FlashCopy
- How FlashCopy works
- How to invoke FlashCopy
- DFSMSdss utility
- Incremental FlashCopy
- FlashCopy Consistency Group

For more information on the ESS Copy Services and details on the differences between FlashCopy Version 1 and Version 2, refer to *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04.

5.1 Introducing FlashCopy

FlashCopy provides a point-in-time (PIT) copy of logical volumes with almost instant availability for the application of both the source and target volumes. Only a minimal interruption is required for the FlashCopy relationship to be established, so the copy operation can be initiated. The copy is then created “under the covers” by the IBM TotalStorage Enterprise Storage Server (ESS), with minimal impact on other ESS activities.

There are currently two FlashCopy versions available: Version1 and Version 2. FlashCopy Version 2 supports all the previous FlashCopy V1 functions plus several enhancements.

With FlashCopy V1:

- FlashCopy V1 works at volume level.
- The source and target volumes must be in the same ESS logical subsystem (LSS).
- A source and a target volume can only be involved in one FlashCopy relationship at a time.

With FlashCopy V2:

- FlashCopy V2 can be used for data set copies, as well as volume copies.
- The source and target of a FlashCopy can be on different LSSs within an ESS.
- Multiple FlashCopy relationships are allowed.
- Incremental copies are possible.
- Inband commands can be sent over PPRC links to a remote site.
- FlashCopy Consistency Groups can be created.
- There has been a reduction in the FlashCopy establish times.

In our tests we used FlashCopy 2. The capabilities and requirements that are important for its use for DB2 are as follows:

- The source and target volumes can be within different ESS logical subsystems and different ESS.
- FlashCopy Consistency Groups can be created across different LSSs and ESSs. For data integrity, we always recommend that BSDS01 and Active Log Copy1 reside on a different ESS from BSDS02 and Active Log Copy 2. Refer to *z/OS V1R5.0 DFSMS Implementing System Managed Storage,* SC26-7407-01 for addressing the issue of data set separation.
- The source and target volumes must have the same track format.
- The target volume must be at least as large as the source volume.
- Version 2 is about 10 times faster than Version 1.

While FlashCopy V2 can produce data set copies as well as volume copies; the DB2 V8 Backup System and Restore System utilities operate only at the volume level.

5.2 How FlashCopy works

As soon as a FlashCopy establish command is issued through the **TSO** command, DFSMSdss utility, or using the ESS Copy Services Web user interface, the ESS establishes a FlashCopy relationship between the target volume and the source volume. This relationship exists from the time you initiate a FlashCopy operation, until the ESS copies all data from the source volume to the target volume. You may optionally request FlashCopy not to execute the background copy, in this case the established relationship must be specifically withdrawn in order to terminate it. A relationship must also be explicitly withdrawn if it was established with the *Persistent FlashCopy* option (as discussed in Chapter 7, “Peer-to-Peer Remote Copy” on page 97).

As outlined in Figure 5-1, there are basically three phases that a FlashCopy relationship goes through:

1. Establishing the relationship
2. Copying the data
3. Terminating the relationship

We now examine these phases in detail.

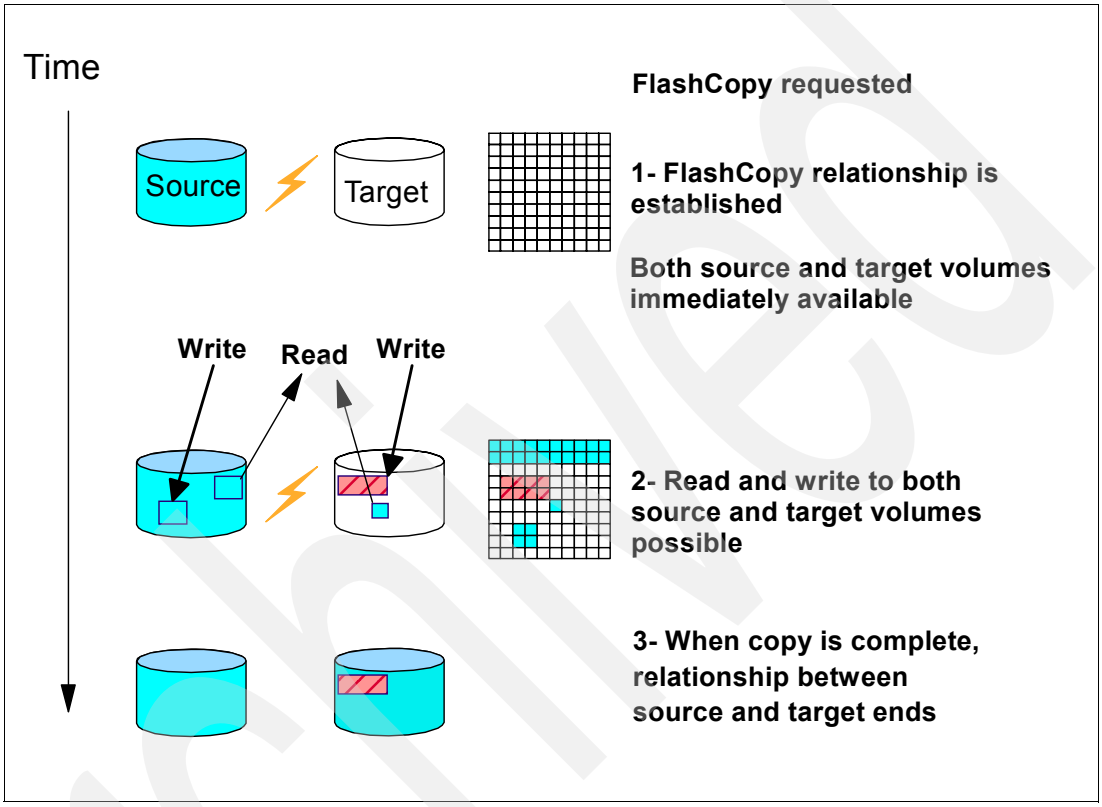


Figure 5-1 FlashCopy with background copy

1: Establishing the relationship

During the *establish* of the FlashCopy relationship, a metadata structure is created for this relationship. This metadata is used by the ESS microcode to map source and target volumes as they were at the time when the FlashCopy was requested, as well as to manage subsequent reads and updates to the source and target volumes. Updates to the source volume *after* the FlashCopy relationship is established will not be reflected on the target device.

The establish process suspends the updates, but takes a minimum amount of time. As soon as the relationship is established, user programs have access to both the source and target copies of the data.

2: Copying the data

With the relationship already established, and the source and target volumes already available for the applications to use them, the copy phase begins. How this copy phase is conducted depends on the background copy option selected for this FlashCopy operation.

The FlashCopy relationship may be established either with or without *background copy*. FlashCopy will manage the copy process differently according to the option specified.

- FlashCopy: Background copy (default)

  By default, FlashCopy will do a background copy. The background copy task does a physical copy of all tracks from the source volume to the target volume. De-staging algorithms are used to efficiently manage the under-the-covers ESS copy process. The background copy task runs at a lower priority than normal I/O on the ESS, not to impact the normal application I/O processing.

  The ESS, using the metadata structure created during establish, keeps track of which data has been copied from the source to the target, and manages the integrity of both copies. If an application wants to read some data from the target that has not yet been copied, the data is read from the source; otherwise, the read can be satisfied from the target volume.

  Before updating a not-yet-copied track on the source volume, the ESS does an *on-demand copy* of the track to the target volume, thus preserving the original track. Subsequent reads to this track on the target volume will be satisfied from the target volume. Before updating a not-yet-copied track on the target volume, the ESS will perform an on-demand copy of this track to the target volume.

  After some time, all tracks will have been copied to the target volume, and the FlashCopy relationship will automatically end unless the persistent FlashCopy option was specified.

  When the **Persistent FlashCopy** option is selected, the FlashCopy relationship does not automatically end when the background physical copy ends. The FlashCopy relationship persists until explicitly withdrawn. Persistent FlashCopy relationships can help protect against inadvertent updates of recently created target volumes. For example, if a source volume is regularly copied to alternating target volumes (thereby ensuring that a complete copy of the source volume is always available), the persistent relationship will identify the target volume for the most recently completed FlashCopy.

- FlashCopy: No background copy

  When selecting *not* to do the background copy, the relationship is established but the background copy task — of all the source volume tracks — is *not* initiated. Only the source tracks which receive application updates will be copied to the target. Before updating a track on the source volume, the ESS does an *on-demand copy* of the track to the target volume, thus preserving the T_0 copy. Similarly, before updating a track on the target volume, the ESS will perform an on-demand copy of this track to the target volume.

  Some DB2 users want only to take the FlashCopied volumes off site for Disaster Recovery. In that case they may choose the NOCOPY option and direct the copy to tape. The advantage is that no background write to the ESS is performed.

3: Terminating the relationship

The FlashCopy relationship is *automatically ended* when all tracks have been copied from source volume to target volume. The relationship can also be *explicitly withdrawn* by commanding that this be done.

How to invoke FlashCopy

FlashCopy can be invoked in the following ways:

- DFSMSdss utility
- TSO commands
- ICKDSF
- ANTRQST macro: an interface that a program can call
- ESS Copy Services Web user interface

- Starting with LIC level 2.3.0, the ESS API is enhanced to support ESS Copy Services configuration and use for PPRC and FlashCopy for the ESS Model 800.

We describe and provide examples of invoking FlashCopy using DFSMSdss utility inasmuch as that is the primary way we ran our tests in Part 4, “Implementing disaster recovery scenarios” on page 249. The other methods are described in greater detail in the redbook.

IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries, SG24-5680-04 and its bibliography.

5.3 DFSMSdss utility

In this section we discuss how to use the DFSMSdss utility to invoke FlashCopy functions.

For more detailed information on DFSMSdss, refer to publication *z/OS DFSMSdss Storage Administration Reference,* SC35-0424.

DFSMSdss can use FlashCopy to perform a full volume copy of the data from the source device to the target device when the following conditions are met:

- The requested DFSMSdss operation is COPY FULL.
- Both the source and target devices are in the same ESS and the ESS has the FlashCopy feature enabled.
- The source and target devices must have the same track format.
- The target must be at least as large as the source volume.
- The source and target volumes must be online.
- The designated target volume cannot also be a primary volume in an XRC or PPRC volume pair. With OA06196 and LIC 2.4.0, FCTOPPRCPRIMARY keyword can be specified on the COPY command to allow a PPRC primary volume to become a FlashCopy target.
- With FlashCopy V2, any source track can have up to 12 FlashCopy targets.

Whenever the conditions for using FlashCopy are met, DFSMSdss will automatically invoke FlashCopy. The copy job will complete after a minimum time when the FlashCopy relationship has been established. If another copy job is started while the source still is in a FlashCopy relationship with another volume, in this case DFSMSdss will start a software host copy.

With V2, not all tracks on the volume are copied when DFSMSdss invokes FlashCopy for a full volume copy. DFSMSdss requests FlashCopy for allocated extents only since FlashCopy V2 can manage source-target relationship at the extent (contiguous set of allocated tracks) level. In order to balance between excluding free space and saving the number of FlashCopy relationships, up to 255 relations will be created for each full volume copy. If there are more than 255 extents on the volume, extents will be merged (to reduce the number of extents), resulting in some free space being copied.

With OW57347, DFSMSdss supports data set FlashCopy on ESS LIC 2.2.0 and later releases. Therefore, DFSMSdss COPY TRACKS, COPY DATASET, and DEFRAG operations will attempt to use FlashCopy if the device supports FlashCopy V2. There are changes in the COPY FULL operation as well when the device supports FlashCopy V2. For example, the volumes no longer need to reside in the same LSS.

5.3.1 COPYVOLID parameter

The COPYVOLID parameter for the COPY command specifies that the volume serial number from the source volume is to be copied to the target volume. This creates an identical copy of the source volume, including the volume serial. The target volume will go offline.

Figure 5-2 shows how the COPYVOLID parameter affects the output of a DFSMSdss full volume COPY operation.

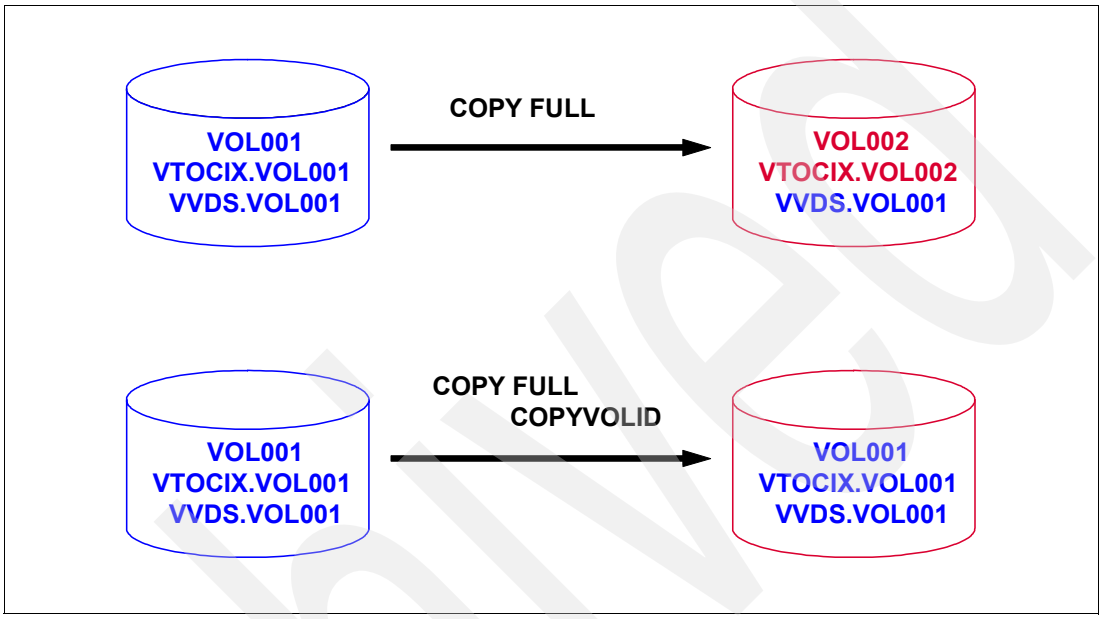


Figure 5-2 DFSMSdss COPY with COPYVOLID

Example 5-1 illustrates how DFSMSdss can be used to invoke FlashCopy in batch. In this example, because the COPYVOLID keyword is specified the volume serial (volser), the VTOC index, and VVDS of the target volume will be identical to the source volume.

Example 5-1 DFSMSdss COPY FULL with COPYVOLID

```
//COPYFULL JOB .....
//*
//CPYVOLID EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  COPY FULL INDYNAM(SRCDEV) OUTDYNAM(TRGDEV) COPYVOLID
/*
```

5.3.2 DUMPCONDITIONING parameter

The DUMPCONDITIONING parameter of the DFSMSdss COPY FULL command allows both the source and target volumes to remain online after a full volume copy operation, thus creating an interim copy for a subsequent dump to tape (or DASD) that can be done using the same system.

Figure 5-3 illustrates in three stages how the volume serial, VTOC index, and VVDS of the copies change during a backup-restore cycle using DFSMSdss DUMPCONDITIONING option.

1. When DUMPCONDITIONING is specified, the volume serial (volser) of the target volume does not change. The VVDS and VTOC index names on the target volume will not change to match the target volume serial, instead they will continue to match the source volume. This volume is a *conditioned volume*. A conditioned volume is not usable in its current state, except for the DFSMSdss DUMP operation, because the volume serial, the VTOC index and VVDS names are not consistent.
2. A full volume dump of the conditioned volume results in a dump data set that looks as if it had been created by dumping the source volume.
3. This allows the dump data set to be restored and used without having to clip back the volser.

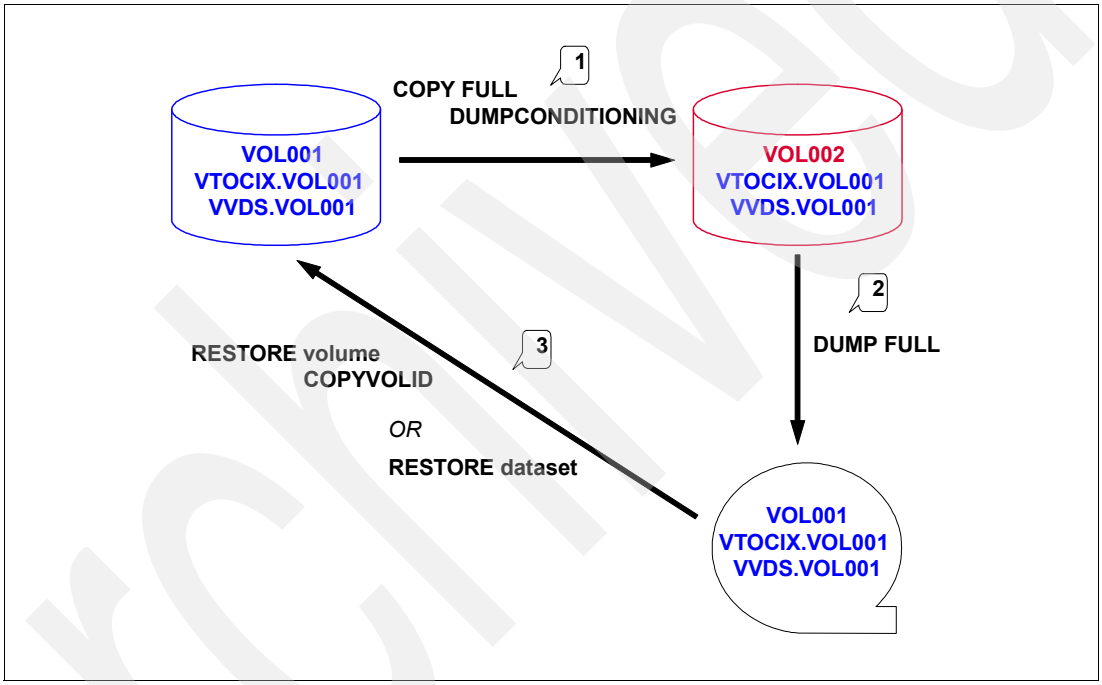


Figure 5-3 DFSMSdss COPY with DUMPCONDITIONING

Example 5-2 illustrates sample JCL that may be used for each of the three stages of the backup-restore cycle of Figure 5-3.

Example 5-2 Backup-restore cycle with DUMPCONDITIONING and background copy

```
//BACKUP JOB .....
//*
//* Step 1 - COPY FULL with DUMPCONDITIONING
//*
//STEP1    EXEC PGM=ADRDSSU
//SYSPRINT  DD  SYSOUT=*
//INDD      DD  UNIT=3390,VOL=SER=RS1510,DISP=SHR
//OUTDD     DD  UNIT=3390,VOL=SER=RS1511,DISP=SHR
//SYSIN     DD  *
  COPY FULL INDDNAME(INDD) OUTDDNAME(OUTDD) DUMPCONDITIONING
/*
//*
//* Step 2 - DUMP FULL
```

```
//*
//STEP2    EXEC PGM=ADRDSSU
//SYSPRINT  DD  SYSOUT=*
//INDD      DD  UNIT=3390,VOL=SER=RS1511,DISP=SHR
//OUTDD     DD  DSN=BACKUP.RS1510,DISP=(,KEEP),LABEL=(1,SL),
//          UNIT=3490,VOL=SER=(TAPE01,TAPE02,TAPE03)
//SYSIN     DD  *
  DUMP FULL INDDNAME(INDD) OUTDDNAME(OUTDD)
/*
//*
//* Step 3 - RESTORE FULL with COPYVOLID
//*
//STEP3    EXEC PGM=ADRDSSU
//SYSPRINT  DD  SYSOUT=*
//INDD      DD  DSN=BACKUP.RS1510,DISP=SHR,LABEL=(1,SL),
//          UNIT=3490,VOL=SER=(TAPE01,TAPE02,TAPE03)
//OUTDD     DD  UNIT=3390,VOL=SER=RS1512,DISP=SHR
//SYSIN     DD  *
  RESTORE INDDNAME(INDD) OUTDDNAME(OUTDD) COPYVOLID
/*
```

Note that as shown in Figure 5-3 on page 73, the restore of the figure in the lower right side (#3) indicates that you can either restore from tape with the COPYVOLID or not. In a disaster recovery situation where you are going to restore at the recovery site, we assume you have initialized your volumes with the same serial numbers as those at the local site for the active DB2 subsystem. Therefore you use “RESTORE volume” without the COPYVOLID parameter, as the volume serial is the same on DASD as the one on your tape. We use that option in several scenarios discussed in Part 4, “Implementing disaster recovery scenarios” on page 249.

5.3.3 FCNOCOPY and FCWITHDRAW parameters

The FCNOCOPY and FCWITHDRAW parameters can be used to make tape backups of the source data, using FlashCopy with the no-background option. This discussion and example are provided because we use them in Chapter 19, “Local recovery: System PITR” on page 363.

FCNOCOPY parameter

When DFSMSdss uses FlashCopy to perform a copy operation, the background copy task is started by default. FCNOCOPY specifies that if FlashCopy is used to perform the copy operation, then the ESS does not perform a physical background copy of the data. If FCNOCOPY is used, you must withdraw the FlashCopy relationship when the copy is no longer needed. This can be accomplished by performing a full volume dump of the target volume with the FCWITHDRAW keyword, or by using the TSO FCWITHDR command.

FCWITHDRAW parameter

FCWITHDRAW specifies that if the volume which is dumped is the target volume of a FlashCopy relationship, then the FlashCopy relationship is withdrawn when the dump has successfully completed.

Example of FCNOCOPY and FCWITHDRAW use

Doing the volume copy without running the background copy task is an efficient procedure when making tape backups, for example. The tape dump is taken from the target volume that is holding the copy of the source volume (the ESS is keeping the integrity while the relationship exists).

The procedure can be done as follows:

1. Start the FlashCopy with the no-background copy option using the COPY command with the FCNOCOPY parameter.
2. Then dump the FlashCopy target to tape using the DUMP command with the FCWITHDRAW parameter.
3. When the tape dump completes successfully, the FlashCopy relationship will be removed automatically.

Example 5-3 illustrates how the backup cycle (steps 1 and 2) shown in Figure 5-3 on page 73 can now be implemented with the no-background copy.

Example 5-3 Backup to tape with no-background copy

```
//BACKUP JOB .....
//*
//* Step 1 - COPY FULL with DUMPCONDITIONING & FCNOCOPY
//*
//STEP1    EXEC PGM=ADRDSSU
//SYSPRINT  DD  SYSOUT=*
//INDD      DD  UNIT=3390,VOL=SER=RS1510,DISP=SHR
//OUTDD     DD  UNIT=3390,VOL=SER=RS1511,DISP=SHR
//SYSIN     DD  *
  COPY FULL INDDNAME(INDD) OUTDDNAME(OUTDD) DUMPCONDITIONING FCNOCOPY
/*
//*
//* Step 2 - DUMP FULL with FCWITHDRAW
//*
//STEP2    EXEC PGM=ADRDSSU
//SYSPRINT  DD  SYSOUT=*
//INDD      DD  UNIT=3390,VOL=SER=RS1511,DISP=SHR
//OUTDD     DD  DSN=BACKUP.RS1510,DISP=(,KEEP),LABEL=(1,SL),
//          UNIT=3490,VOL=SER=(TAPE01,TAPE02,TAPE03)
//SYSIN     DD  *
  DUMP FULL INDDNAME(INDD) OUTDDNAME(OUTDD) FCWITHDRAW
/*
```

5.4 Incremental FlashCopy

Incremental FlashCopy provides the capability to *refresh* a FlashCopy relationship. With Incremental FlashCopy, the initial relationship between a source and target is maintained. When a subsequent FlashCopy establish is initiated, the only data copied is the data required to bring the target current to the source's newly established point-in-time.

Incremental FlashCopy helps reduce the background copy completion time when only a subset of data on either the source or target has changed, giving you the option to perform a FlashCopy on a more frequent basis.

Incremental FlashCopy must be invoked from the ESS Copy Services Web user interface panels — FlashCopy TSO commands do not offer parameters to invoke this function. In order for an incremental FlashCopy to be performed, the FlashCopy relationship must first be established with the *Start Change Recording* and *Persistent FlashCopy* options enabled:

- With change recording enabled, metadata structures are created to track changes to both source and target volumes from the time the relationship was established. The ESS can then identify the changed tracks that need to be copied when an incremental FlashCopy is requested.

- Incremental FlashCopy is only possible with a persistent relationship. With persistent relationships, the relation between source and target is maintained after background copy has completed. This allows the ESS to continue tracking updates to source and target extents.

Note: Incremental FlashCopy is supported at volume level. It is not available for data set FlashCopy.

Figure 5-4 illustrates how incremental FlashCopy works:

1. When the initial FlashCopy is established from volA to volB with change recording enabled, FlashCopy creates metadata structures to track any updates that may be done on those volumes.
2. As changes are made to volA, FlashCopy will keep track of the tracks that are updated. Similarly, FlashCopy will keep track of any updates that are done on volB.
3. When an incremental FlashCopy is requested from volA to volB, FlashCopy identifies the set of tracks that must be copied from volA to volB. For example, if volA had tracks 1, 3, 5 updated, and volB had tracks 2, 7, 9 updated, then tracks 1, 2, 3, 5, 7, 9 would be copied from volA to volB. In other words, all tracks necessary to make volB look like volA at the time of the flash will be identified to be copied from volA to volB.
4. Finally, the metadata structures are reset to start tracking changes from this point, if requested.

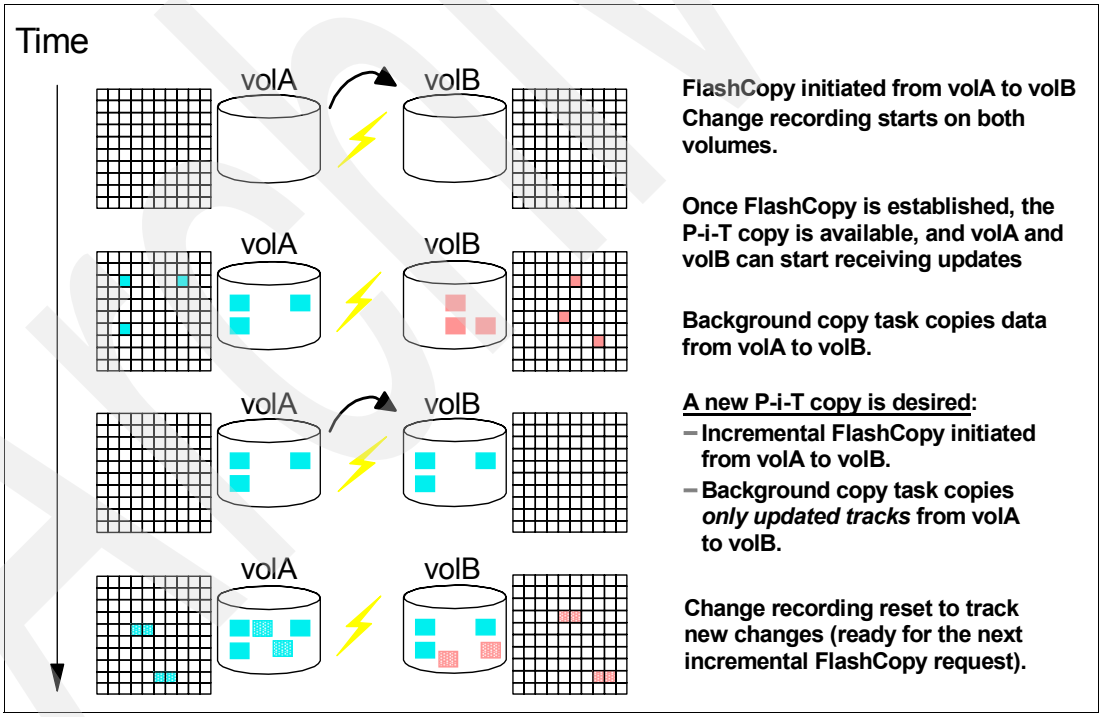


Figure 5-4 Incremental FlashCopy

For more detailed information, refer to the publication *IBM TotalStorage Enterprise Storge Server Web Interface User's Guide,* SC26-7448.

5.5 FlashCopy Consistency Group

When FlashCopy is established for a large number of source volumes, there is a finite amount of time between the first and last establish, so that copies will not be created at a consistent point-in-time.

With the **`Freeze FlashCopy Consistency Group`** option, the ESS will hold off I/O activity to a volume for a time period by putting the source volume in *extended long busy (ELB) state*. Thus, a window can be created during which dependent write updates will not occur, and FlashCopy will use that window to obtain a consistent point-in-time copy of the related volumes. I/O activity resumes when a FlashCopy consistency group created is issued, or when the extended long busy window expires (ELB window is 2 minutes by default).

Consistency groups can be used to help create a consistent point-in-time copy across multiple volumes, and even across multiple ESSs, thus managing the consistency of dependent writes.

Figure 5-5 illustrates how FlashCopy Consistency Group can be used to manage the consistency of dependent writes:

1. FlashCopy is established from srcA (source A) to tgtA (target A) with the *Freeze FlashCopy Consistency Group* option enabled. Volume srcA is placed in extended long busy (ELB) state and thus, all the I/O activity to this volume is enqueued. Because writes on srcA cannot proceed, neither will its dependent writes on the other volumes (srcB, srcC, srcD) proceed. This ensures the logical integrity of the related data spanning the whole set of volumes. Independent writes to other volumes (srcB, srcC, srcD) are not affected.
2. FlashCopy is then established from srcB to tgtB with the consistency group function enabled. Volume srcB is placed in ELB state — as srcA currently is. Thus, volumes srcA and srcB are not receiving updates. Dependent writes on srcC and srcD waiting on write completions on srcA or srcB cannot proceed. This ensures the integrity of the data over the whole set of related volumes (srcA, srcB, srcC, srcD). Independent writes to volumes srcC and srcD are not affected.
3. As FlashCopy is established from srcC to tgtC, then from srcD to tgtD, each source volume is placed in ELB state and dependent writes held.
4. When all FlashCopy pairs have been established, volumes tgtA, tgtB, tgtC and tgtD contain a consistent copy of data (the order of dependent writes is preserved).
5. The **`Consistency Group Created`** command can be issued to remove the source volumes from ELB state, so that updates can resume.

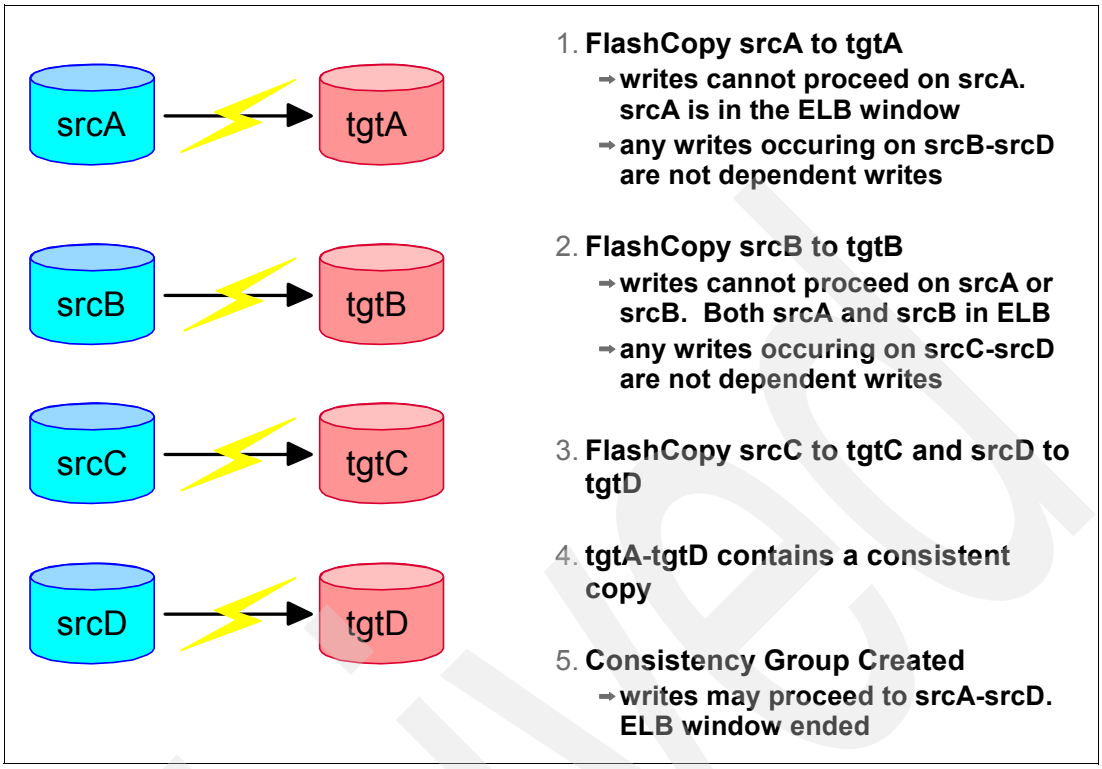


Figure 5-5 FlashCopy consistency group

The FlashCopy Consistency Group function can only be activated through the ESS Copy Services Web user interface. For detailed information, refer to the publication *IBM TotalStorage Enterprise Storge Server Web Interface User's Guide,* SC26-7448.

6

SMS copy pools and DB2 point in time recovery

In this chapter we provide information on using the DFSMS copy pools and DFSMShsm Fast Replication in DB2 environment for a point-in-time recovery.

DFSMShsm Fast Replication provides DFSMShsm management for the use of volume level fast replication. The fast replication is made possible by exploiting the FlashCopy capability of the IBM Enterprise Storage Server. With this capability, a set of storage groups can be defined as a copy pool. The volumes in this pool are processed collectively, by creating, via Fast Replication, backup versions managed by DFSMShsm.

Recovery can be performed at the volume or user defined volumes set level. With DFSMShsm Fast Replication, the backup and recovery of DB2 Storage Groups can be managed by DFSMShsm. Non-disruptive backups can be taken automatically. Recovery is also fast and easy to perform. Copy pools and Fast Replication provide a fast, easy to use backup and recovery solution designed to work specifically with DB2 Version 8 or later.

This chapter covers the following topics:

- DFSMS 1.5 enhancements
- Preparing for DFSMShsm Fast Replication
- DB2 enhancements
- DB2 PITR using Fast Replication
- DB2 Recover using Fast Replication backups

The following publications should be referenced to complement the information presented in this chapter:

- *DFSMShsm Fast Replication Technical Guide*, SG24-7069
- *DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, .. and More,* SG24-6079

6.1 DFSMS 1.5 enhancements

In this section we provide a brief overview of SMS 1.5 enhancements.

6.1.1 DFSMShsm Fast Replication

DFSMShsm is enhanced in z/OS DFSMS V1.5 to manage full volume fast replication backup versions.

Although the primary purpose for this enhancement is to support an enhanced backup/recovery environment for SAP R/3 applications running on DB2 middleware on a zSeries processor, the enhancement is general enough that it may be used for other applications.

DFSMShsm introduces the new copy pool construct and SMS copy pool backup storage group type in support of this new functionality. DFSMShsm manages the fast replication backups, and recovery can be performed only at volume or copy pool level. The following requirements must be met to take advantage of this new functionality:

- z/OS V1R5 or above
- SMS managed DB2 data sets
- Disk control units that support ESS FlashCopy
- Copy pools for data and logs
- Backup storage groups for each source storage group in a copy pool

Three new DFSMShsm commands are introduced to support this new function, and other commands are changed to provide new information related to the DFSMShsm Fast Replication function:

- **FRBACKUP:** Create a fast replication backup version for each volume in a specified copy pool.
- **FRRECOV:** Use fast replication to recover a single volume or a pool of volumes from the managed backup versions
- **FRDELETE:** Delete one or more unneeded fast replication backup versions.

LIST and QUERY commands are modified to aid monitoring of the fast replication backup versions.

6.1.2 DFSMS copy pools

SMS is also enhanced in z/OS DFSMS V1.5 to support this new function. A new construct named *copy pool* and a new *copy pool backup* storage group type are introduced.

The copy pool construct enables customer to define which storage group should be processed for fast replication functions.

The copy pool backup storage group type are used to define which volumes DFSMShsm may use as the target volumes of the fast replication backup versions.

A copy pool is a set of SMS pool storage groups that can be processed by fast replication operations as one unit with one command.

The Interactive System Management Facility (ISMF), the ISPF menu driven application used to control SMS, has been enhanced to support these SMS enhancements.

A copy pool can contain up to 256 pool storage groups to be processed for fast replication operations, and each pool storage group must be associated with a new type of storage group called the copy pool backup storage group. A pool storage group can have only one copy pool backup storage group associated, and many pool storage groups can be associated with the same copy pool backup storage group. So in a copy pool backup storage group, we can have different versions of different pool storage groups all together.

HSM keeps the inventory of the backup versions in the backup control data set (BCDS). The BSDS will request a version by specifying a token associated with the needed RBA. Each copy pool has a VERSIONS attribute that specifies how many versions should be maintained on disk, with a default of 2 and a maximum of 85.

Volumes to be copied are evaluated at processing time rather than at definition time so that changes to the copy pool after definition are reflected in future processing. The copy pool backup storage group must contain enough volumes for a unique one-to-one relationship with the volumes in the pool storage group.

Important: We recommend that you keep at least two versions of the pools because, before a new copy is created, the oldest one is invalidated and the target volumes are overwritten. If that backup fails, there is no way to recover the backup that was invalidated.

As a general rule, if *n* backups are required, *n+1* should be kept.

6.1.3 Copy pool backup storage group type

The new copy pool backup storage group is used to contain backup target volumes for DFSMShsm fast replication operations.

Restrictions:

- **For FlashCopy V2**: An eligible target volume must have the same track form as the source volume, be the exact size of the source volume, and reside in the same ESS as the source volume.
- **For FlashCopy V1**: The same restrictions apply as for FlashCopy V2, and additionally, the target and the source volume must reside in the same LSS.
- **For SnapShot**: An eligible target volume must have the same track form as the source volume, and be the exact size of the source volume.

Warning:

- The data within a copy pool should not be migrated!

The copy pool backup storage group cannot be accessed by ACS routines as SMS will prevent new data set allocations to this type of storage group. There must be a sufficient number of volumes in the backup target storage group to satisfy the number of backup versions specified for a source storage group. For example, if a system has 10 source volumes and the VERSIONS attribute has been specified as 2, the backup storage group must have at least 20 volumes to satisfy 2 backup versions of 10 volumes each.

SMS provides a new storage group attribute to associate a source storage group to a backup target storage group. Notice that SMS does not verify that extend and overflow storage groups that are associated with main source pool storage groups have been included in a copy pool definition. They must be included in the storage group list for appropriate copy pools and they also must be associated to back up target storage groups.

Figure 6-1 illustrates the copy pool structure and the relationship between source and backup target storage groups.

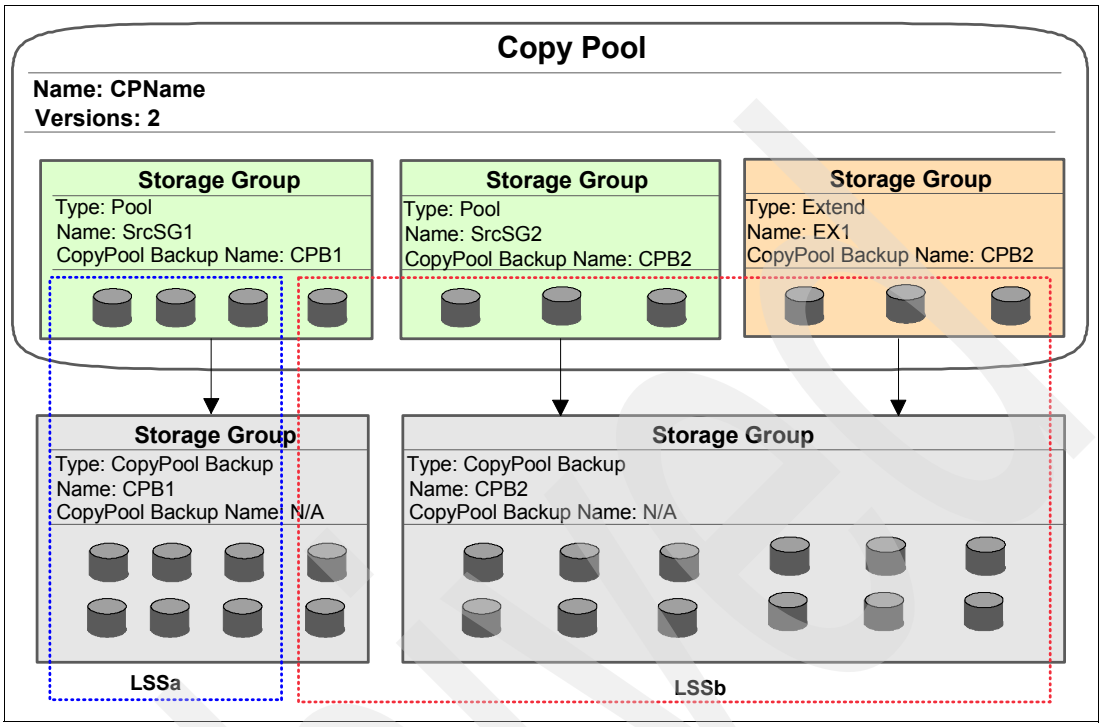


Figure 6-1 Copy pool structure

In Figure 6-1, the copy pool contains three source storage groups, two with source data (SrcSG1, SrcSG2), and an extend storage group (EX1). Two copy pool backup target storage groups (CPB1, CPB2) are associated with the source storage groups.

6.2 Preparing for DFSMShsm Fast Replication

Your storage systems administrator needs to define *database* and *log* copy pools and associated source and backup storage groups in order to use fast replication operations. The *database copy pool* should contain all volumes that contain the DB2 catalog and directory, and all user data. The *log* copy pool should contain active logs and the BSDS.

After establishing the copy pools, the Storage Administrator issues FRBACKUP PREPARE against each copy pool to validate the fast replication environment. DFSMShsm will allocate backup versions with an empty token value and the required number of volumes for each of the versions specified in the copy pool definition.

This token value is used by DB2 in the recovery process. In our example above, 2 versions of 10 backup target volumes each, or 20 backup volumes, would be allocated. This will ensure that there are sufficient target volumes and will move the target volume selection process outside of the fast replication window. If you don't use the PREPARE keyword prior to taking backups, only copy pool backup volumes for the backup in progress will be allocated.

See Appendix B.1, “Setting up copy pools for PITR” on page 464 for details on the sequence of ISMF panels useful for your storage administrator.

6.3 DB2 enhancements

DB2 for z/OS V8 is enhanced to provide backup and recover capabilities at the DB2 subsystem or data sharing group level. The purpose is to provide an easier and less disruptive way to make fast volume level backups of an entire DB2 subsystem or data sharing group with a minimal disruption, and recover a subsystem or data sharing group to any point-in-time , regardless of whether you have uncommitted units of work. SAP exploits these new recovery functions as well.

DB2 V8 provides a fast, easy, minimally disruptive way to create volume level backups and a fast, reliable way to recover to an arbitrary point-in-time with the new BACKUP SYSTEM and RESTORE SYSTEM utilities:

- The Backup System utility provides fast volume level copies of DB2 databases and logs.
- The Restore System recovers a DB2 system to an arbitrary point-in-time. RESTORE SYSTEM automatically handles any creates and drops that might have occurred between the backup and the recovery point-in-time.

DB2 V8 uses DFSMShsm functionality to simplify and improve the performance and reliability of the backup and recovery process. The BACKUP SYSTEM and RESTORE SYSTEM utilities encapsulate all tasks previously required to perform each function into one utility statement each.

Copy pools for DB2

For DB2, only two types of copy pools are valid:

- A database copy pool, containing all database objects, including DB2 catalog and directory, and all user data
- A log copy pool, containing all active logs and BSDS data sets.

The copy pools for a DB2 environment must follow a strict naming convention of the form DSN$*locn-name*$*cp-type*, where:

- DSN is the unique DB2 product identifier
- $ is a required delimiter
- *locn-name* is the DB2 location name
- $ is a required delimiter
- *cp-type* is the copy pool type; DB for database, LG for logs

For example, DB2 DB1P would have copy pools named DSN$DB1P$DB for the database copy pool and DSN$DB1P$LG for the log copy pool.

Important: We recommend that you create separate ICF catalogs for each copy pool.

The BACKUP SYSTEM utility requires that at least a database copy pool exists to take a data-only backup. If you plan to take full system backups (database, logs, BSDS) you must define the log copy pool as well. We recommend that you create separate ICF catalogs for each copy pool.

The RESTORE SYSTEM utility uses the database copy pool only.

In the BSDS, up to 50 versions are allowed.

Use of DB2 tokens

DB2 uses an 18 character long token which is stored by DFSMShsm in its backup control data set (BCDS) records. The DB2 token breakout is shown in Table 6-1.

Table 6-1 The DB2 token

Field contents	Field description	length
DB2 SSID	DB2's Subsystem ID	4
TOD	Time of Day of SYSPITR= LRSN (GMT)	8
RBA	Checkpoint RBA	6

Example 6-1 shows how it looks like from a DFSMShsm LIST command output.

Example 6-1 DFSMShsm LIST command output

```
F HSM,LIST CP(DSN$DB8B$DB) TERM
COPYPOOL= DSN$DB8B$DB                    VER=002, VALID=Y, VTOCENQ=N,
MADE ON 2003/11/12 AT 17:58:01
TKN(C)=C'DB8B & U+       - '
TKN(H)=X'C4C2F8C2BA5053E44E472848000002426090'
SGNAME=DB8B      SOURCE=MHL125  TARGET=MHL213
SGNAME=DB8B      SOURCE=MHL126  TARGET=MHL214

ARC0140I LIST COMPLETED, 6 LINE(S) OF DATA 958
```

Table 6-2 shows its breakout for the example.

Table 6-2 DB2 token contents breakout of Example 6-1 output listing

Field contents	length	value	human readable format
DB2 SSID	4	C4C2F8C2	DB8B
TOD	8	BA5053E44E472848	Nov. 12th, 2003 at 22:58:01,939058
RBA	6	000002426090	000002426090

NOVTOCENQ keyword

DB2 uses the *NOVTOCENQ* (no VTOC enqueue) keyword to communicate to DFSMShsm that itself prevents that data sets are neither created, extended, opened, nor closed during the DFSMShsm Fast Replication.

6.3.1 Other changes to support system level point-in-time recovery

DB2 stand alone utilities have been changed to reflect the new functions.

DSNJU003 Change Log Inventory

A new option has been added to DSNJU003, Change Log Inventory, to allow you to create a new type of conditional restart control record to truncate logs for a system point-in-time recovery in preparation for running the RESTORE SYSTEM utility. The syntax of the option is:

```
CRESTART CREATE SYSPITR=log-point
```

The log-point is the RBA or LRSN (for data sharing) to which you want to recover the system. The SYSPITR option cannot be specified with any other option and is only allowed after NFM is enabled. When DB2 is started with a SYSPITR CRCR, the system starts in system recovery-pending mode and only the RESTORE SYSTEM utility can be run.

Most DB2 commands will work, with the exception of START DATABASE and TERM UTIL. A DIS UTIL command will display only the status of the RESTORE SYSTEM utility. Data is unavailable until the utility completes successfully and DB2 is recycled. DB2 must be recycled after recovery is complete to reset recovery-pending status.

If data sharing is active, each member that has been active after the log truncation point must have a SYSPITR CRCR created with the same log truncation point and must be started prior to recovery. The RESTORE SYSTEM utility cannot be run until all members have been started.

DSNJU004 Print Log Map

DSNJU004 Print Log Map has been enhanced to show SYSPITR CRCR information if it is present and system backup information. Example 6-2 illustrates the changes.

Example 6-2 DSNJU004 output.

```
23:47:48 OCTOBER 08, 2003
**** ACTIVE CRCR RECORD ****
CRCR IDENTIFIER  0001
     USE COUNT   0
     RECORD STATUS
         CRCR ACTIVE
         CRCR NOT USED
     PROCESSING STATUS
         FORWARD = YES
         BACKOUT = YES
     SYSPITR SYSTEM LEVEL RECOVERY MODE RESTART
     STARTRBA                      NOT SPECIFIED
     ENDRBA                        NOT SPECIFIED
     ENDLRSN                       BA245C6867D9  <------SYSPITR
     EARLIEST REQUESTED RBA        000000000000
     FIRST LOG RECORD RBA          000000000000
     ORIGINAL CHECKPOINT RBA       000000000000
     NEW CHECKPOINT RBA (CHKPTRBA) 00001022BE2E
     END CHECKPOINT RBA            00001023AF66
     CRCR CREATED            23:47:18 OCTOBER 08, 2003
     TIME OF CHECKPOINT      23:41:28 OCTOBER 08, 2003
     RESTART PROGRESS                STARTED     ENDED
                                     =======     =====
        CURRENT STATUS REBUILD         NO         NO
        FORWARD RECOVERY PHASE         NO         NO
        BACKOUT RECOVERY PHASE         NO         NO


                  BACKUP SYSTEM UTILITY HISTORY
                        SUBSYSTEM ID DJ1G
                    23:47:49 OCTOBER 08, 2003
            START STCK                          DATA COMPLETE
DATA/LOG
COMPLETE
     DATA              LOG             RBLP          LRSN         DATE
LTIME    LOCATION NAME
----------------  ----------------  ------------  ------------
--------------------  ----------------
BA2458B70E2AC5AE  0000000000000000  BA245635C2B2  BA245635C2B2
2003/10/08
```

SET LOG SUSPEND/RESUME

The SET LOG SUSPEND and SET LOG RESUME commands have been enhanced to quiesce 32 KB page writes and data set extends.

DSN1LOGP

DSN1LOGP has been enhanced to print a system event type log record, subtype=8, which is generated when a system goes into system recover-pending mode.

DSN1PRNT

DSN1PRNT has been enhanced to print the *recovery base log point* (RBLP) stored in the header page of DBD01.

6.4 DB2 PITR using Fast Replication

In this section we describe the DB2 Backup System utility and the invocation of the DFSMShsm Fast Replication function.

6.4.1 DB2 BACKUP SYSTEM utility

The BACKUP SYSTEM utility invokes new fast replication services in z/OS DFSMShsm V1R5 to take volume level copies of only the database portion of a system (DB2 catalog, directory and user data) or the entire DB2 subsystem which includes the database and log (active logs and BSDS) portions of a subsystem or data sharing group. These copies are taken without DB2 having to take any quiesce points and can be used to restore a subsystem or data sharing group to a prior point-in-time, even when there is uncommitted data.

Two options are available to you when running the BACKUP SYSTEM utility:

- FULL:

  This indicates that you want to copy both the database copy pool and the log copy pool. This is the default.

- DATA ONLY:

  This indicates that you want to copy the database copy pool only. This option will not copy the log copy pool.

Database only backups can be taken by using the DATA ONLY keywords. This tells the utility to copy only the *database* copy pool. Full system backups are the default and can be explicitly specified with the FULL keyword. Both types of backups can be used for point-in-time recovery with the RESTORE SYSTEM utility because it only uses the *database* copy pool to restore the data prior to applying logs.

A full system backup can also be used to restore a system to the time the backup was taken, for disaster recovery, or for cloning purposes. In a full system backup, the *database* copy pool is copied first and the *log* copy pool is copied second so that normal DB2 restart recovery processing can be used to restore data consistency when restoring to a full backup.

During backup, DB2 records a RBLP in the header page of DBD01. The RBLP is identified as the most recent system checkpoint prior to a backup log point, and the point at which DB2 starts scanning logs during a RESTORE SYSTEM recovery operation. DB2 updates its BSDS (boot strap data set) with backup version information and can keep track of up to 50 backup versions. In the case of data sharing, the submitting member records the backup version in its BSDS and also in the SCA.

In general, the BACKUP SYSTEM utility performs the following steps:

- Takes a new exclusive lock to ensure no other backup utility can execute. If data sharing, takes a global lock.
- Suspends 32 KB page writes for objects created prior to NFM. You can avoid the write suspension by REORGing these objects. If data sharing, all members are notified.
- Suspends data set creation (create table space, index, etc.), deletion (drop table space, index, etc.), renames (online REORG fast switch), and extensions. If data sharing, all members are notified.
- Suspends system checkpoints. If data sharing, all members are notified.
- Prevents data sets from pseudo close. If data sharing, all members are notified.
- Records the RBLP RBA or LRSN in the header page of DBD01 and writes the page to DASD. If data sharing, the system checkpoint prior to the lowest LRSN of all active members is used.
- Updates the BSDS with the system backup information. If data sharing, only the BSDS for the submitting member is updated.
- Invokes DFSMShsm to take a FlashCopy of the 'database' copy pool.
- Invokes DFSMShsm to take a FlashCopy of the 'log' copy pool if it is a full system backup.
- Resumes all suspend activities above. If data sharing, notifies all members.
- Releases the exclusive lock. If data sharing, notifies all members.
- Issues an informational message indicating the backup is complete.

Figure 6-2 illustrates what happen during the backup process.

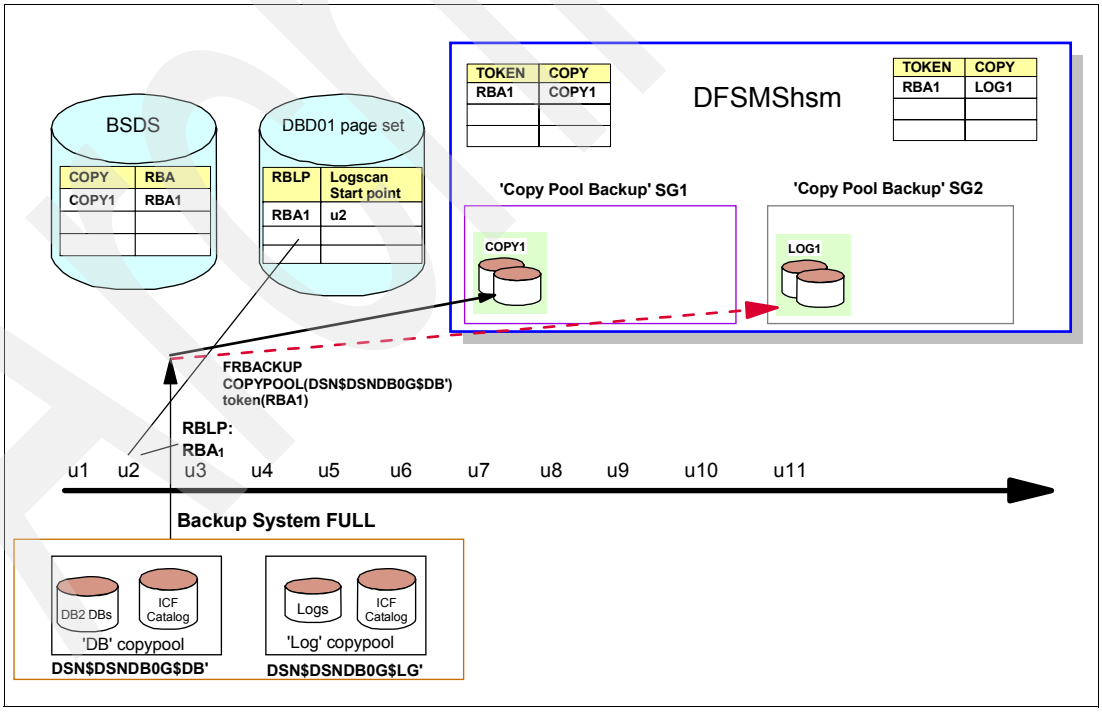


Figure 6-2 BACKUP SYSTEM utility execution for DSNDB0G

One BACKUP SYSTEM utility has been run, and a full system backup has been taken for DSNDB0G. The information is recorded in the DB2 BSDS, the header page of DBD01, and in DFSMShsm control data sets.

Figure 6-3 shows what happens during the BACKUP SYSTEM FULL process, and also illustrates what happens as more BACKUP SYSTEM copies are taken:

- The BSDS is updated with COPY1 version information at RBA1, the RBLP in DBD01 is updated with the most recent system checkpoint RBA or LRSN prior to backup RBA1 at u2. DFSMShsm records COPY1 and RBA1 and keeps track of the DB and LG copy pool copies.
- Three BACKUP SYSTEM backups have been taken for DSNDB0G: two BACKUP SYSTEM FULL and one DATA ONLY. The second backup is also a full system backup and the same sequence of events occurs, but the third backup is a data only backup. Notice that the same information is recorded in the BSDS and the header page of DBD01. DFSMShsm records the same copy version information but only takes a copy of the DB copy pool.

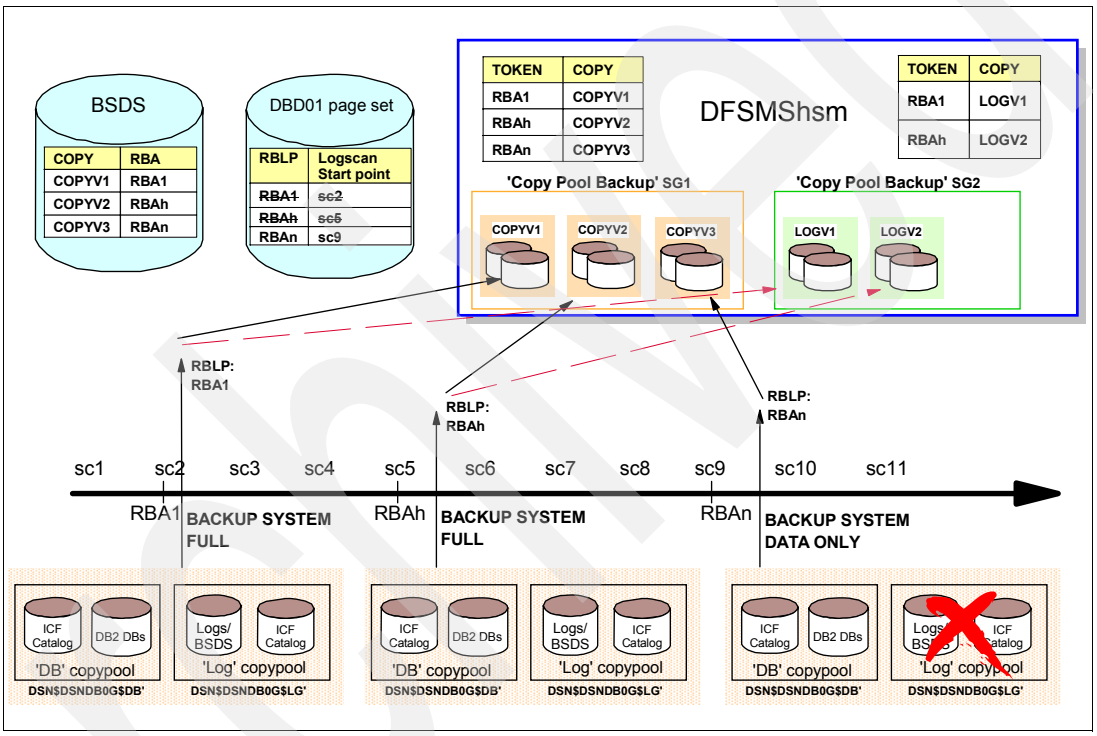


Figure 6-3 Two BACKUP SYSTEM FULL, and one DATA ONLY are taken

Running BACKUP SYSTEM

We now show samples of the JCL required and the control statements necessary to invoke DFSMShsm Fast Replication with DB2.

> **Note:** Only one BACKUP SYSTEM job can be running at one time.

Example 6-3 shows an instance of full backup.

Example 6-3 Invoking DFSMShsm Replication with DB2 utility

```
//STEP1    EXEC DSNUPROC,SYSTEM=DB8B,UID=DIAG
//*        UTSTATS=''
//SYSIN    DD  *
  BACKUP SYSTEM FULL
```

If you do not specify the FULL option in the Example 6-3, the FULL option is still implied, however, as this is the default.

In Example 6-4, the DATA ONLY option has been used. This will not copy the log copy pool.

Example 6-4 Invoking DFSMShsm Fast Replication with DB2 utility (DATA ONLY option)

```
//STEP1    EXEC DSNUPROC,SYSTEM=DB8B,UID=DIAG
//*        UTSTATS=''
//SYSIN    DD  *
  BACKUP SYSTEM DATA ONLY
```

Once complete, DFSMShsm has a record of the fast replication copy invoked by DB2 and maintains this information until it expires. You can see the copy pool token information stored in DFSMShsm by using the DFSMShsm LIST command. Example 6-5 shows the database output.

Example 6-5 DFSMShsm LIST command and output on DSN$DB8B$DB copy pool

```
F HSM,LIST CP(DSN$DB8B$DB) ODS(MHLRES3.DB8B.LIST)

-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 18:36:34 ON 03/11/12 FOR SYSTEM=SC65

COPYPOOL = DSN$DB8B$DB

VERSION  VALID VTOCENQ      DATE           TIME
  002      Y      N      2003/11/12      17:58:01
TOKEN(C)=C'DB8B[&ëU+å.ç...â-°'
TOKEN(H)=X'C4C2F8C2BA5053E44E472848000002426090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8B     MHL125 - MHL213  MHL126 - MHL214

VERSION  VALID VTOCENQ      DATE           TIME
  001      Y      N      2003/11/12      17:46:29
TOKEN(C)=C'DB8B[&é│^.Ç¢...â-°'
TOKEN(H)=X'C4C2F8C2BA50514FB03D684A000002426090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8B     MHL125 - MHL013  MHL126 - MHL014

----- END OF -- COPY POOL -- LISTING -----
```

Example 6-6 shows the log output.

Example 6-6 DFSMShsm LIST command and output on DSN$DB8B$LG copy pool

```
F HSM,LIST CP(DSN$DB8B$LG) ODS(MHLRES3.DB8B.LIST)

-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING --------- AT 18:36:43 ON 03/11/12 FOR SYSTEM=SC65

COPYPOOL = DSN$DB8B$LG

VERSION  VALID VTOCENQ      DATE           TIME
  002      Y      N      2003/11/12      17:58:04
TOKEN(C)=C'DB8B[&ëU+å.ç...â-°'
TOKEN(H)=X'C4C2F8C2BA5053E44E472848000002426090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8BLOG1 MHL037 - MHL225
```

```
DB8BLOG2 MHL147 - MHL226

VERSION  VALID VTOCENQ      DATE           TIME
  001      Y      N      2003/11/12      17:46:32
TOKEN(C)=C'DB8B[&é|^.Ç¢...â-°'
TOKEN(H)=X'C4C2F8C2BA50514FB03D684A000002426090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8BLOG1 MHL037 - MHL025
DB8BLOG2 MHL147 - MHL026

----- END OF -- COPY POOL -- LISTING -----
```

Sample DB2 messages for DB2 BACKUP

In Example 6-7 you see the output from an execution of the DB2 Backup utility specifying BACKUP SYSTEM FULL.

Example 6-7 DB2 Backup System Full

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DIAG
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  BACKUP SYSTEM FULL
DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA STARTING,
                       COPYPOOL = DSN$DB8B$DB
                       TOKEN = X'C4C2F8C2BA5053E44E472848000002426090'.
DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$DB8B$DB
                       TOKEN = X'C4C2F8C2BA5053E44E472848000002426090'
                       ELAPSED TIME = 00:00:02.
DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS STARTING,
                       COPYPOOL = DSN$DB8B$LG
                       TOKEN = X'C4C2F8C2BA5053E44E472848000002426090'.
DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$DB8B$LG
                       TOKEN = X'C4C2F8C2BA5053E44E472848000002426090'
                       ELAPSED TIME = 00:00:01.
DSNU1602I   DSNUVBBD - BACKUP SYSTEM UTILITY COMPLETED, ELAPSED TIME = 00:00:04
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Sample DFSMShsm messages for DB2 BACKUP

Example 6-8 shows the HSM output written to syslog when the FRBACKUP commands are executed during the DB2 Backup execution.

Example 6-8 FRBACKUP output in syslog

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8B$DB, AT 17:58:01
ARC1801I (CONT.) ON 2003/11/12
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION BACKUP OF COPY POOL
ARC1805I (CONT.) DSN$DB8B$DB
ARC1805I (CONT.) MHL125
ARC1805I (CONT.) MHL126
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8B$DB, AT 17:58:04
ARC1802I (CONT.) ON 2003/11/12, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8B$LG, AT 17:58:04
ARC1801I (CONT.) ON 2003/11/12
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION BACKUP OF COPY POOL
ARC1805I (CONT.) DSN$DB8B$LG
ARC1805I (CONT.) MHL037
ARC1805I (CONT.) MHL147
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8B$LG, AT 17:58:05
ARC1802I (CONT.) ON 2003/11/12, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

Example 6-9 shows the output written to DFSMShsm backup log data set from execution of the FRBACKUP command.

Example 6-9 FRBACKUP output in DFSMShsm backup log data set

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB8B$DB, AT 17:58:01 ON 2003/11/12
ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R05.0 DATA SET SERVICES     2003.316 17:58
ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC0640I ARCFRTM -  PARALLEL
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL'
ARC0640I ARCFRTM -  COPY FULL IDY(MHL125) ODY(MHL213) ALLX ALLD(*) DUMPCOND FR(REQ) PUR
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM -  COPY FULL IDY(MHL126) ODY(MHL214) ALLX ALLD(*) DUMPCOND FR(REQ) PUR
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2003.316 17:58:03 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
 2003.316 17:58:03 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR035I (002)-PRIME(86), INSTALLATION EXIT ALTERED DEBUG(FRMSG) OPTION TO UNSPECIFIED
ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR035I (003)-PRIME(86), INSTALLATION EXIT ALTERED DEBUG(FRMSG) OPTION TO UNSPECIFIED
ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2003.316 17:58:03 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (002)-TOMI (02), VOLUME MHL125 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2003.316 17:58:03 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2003.316 17:58:03 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2003.316 17:58:03 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME MHL126 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2003.316 17:58:03 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2003.316 17:58:03 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2003.316 17:58:03 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL DSN$DB8B$DB
ARC1805I (CONT.) MHL125
ARC1805I (CONT.) MHL126
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB8B$DB, AT 17:58:04 ON 2003/11/12, FUNCTION RC=0000,
 MAXIMUM VOLUME RC=0000
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB8B$LG, AT 17:58:04 ON 2003/11/12
ARC0640I ARCFRTM - PAGE 0001     5695-DF175  DFSMSDSS V1R05.0 DATA SET SERVICES     2003.316 17:58
ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC0640I ARCFRTM -  PARALLEL
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL'
ARC0640I ARCFRTM -  COPY FULL IDY(MHL037) ODY(MHL225) ALLX ALLD(*) DUMPCOND FR(REQ) PUR
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM -  COPY FULL IDY(MHL147) ODY(MHL226) ALLX ALLD(*) DUMPCOND FR(REQ) PUR
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2003.316 17:58:05 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
```

```
 2003.316 17:58:05 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
ARC0640I ARCFRTM - ADR050I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR035I (002)-PRIME(86), INSTALLATION EXIT ALTERED DEBUG(FRMSG) OPTION TO UNSPECIFIED
ARC0640I ARCFRTM - ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR050I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADR035I (003)-PRIME(86), INSTALLATION EXIT ALTERED DEBUG(FRMSG) OPTION TO UNSPECIFIED
ARC0640I ARCFRTM - ADR016I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADR006I (003)-STEND(01), 2003.316 17:58:05 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000001:0005 AND ENDING AT 000001:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME MHL147 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2003.316 17:58:05 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2003.316 17:58:05 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR006I (002)-STEND(01), 2003.316 17:58:05 EXECUTION BEGINS
ARC0640I ARCFRTM - ADR241I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000001:0005 AND ENDING AT 000001:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (002)-TOMI (02), VOLUME MHL037 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (002)-STEND(02), 2003.316 17:58:05 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (002)-CLTSK(01), 2003.316 17:58:05 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2003.316 17:58:05 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL DSN$DB8B$LG
ARC1805I (CONT.) MHL037
ARC1805I (CONT.) MHL147
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB8B$LG, AT 17:58:05 ON 2003/11/12, FUNCTION RC=0000,
 MAXIMUM VOLUME RC=0000
```

6.5 DB2 Recover using Fast Replication backups

In this section we describe how DFSMShsm Fast Replication backups are used during DB2 recovery. We describe the recovery scenarios where Fast Replication backups are used.

6.5.1 DB2 RESTORE SYSTEM utility

Use the RESTORE SYSTEM utility only when you want to recover a subsystem or data sharing group to an arbitrary point-in-time. The utility restores only the *database copy pool* of a data only or full system backup, and then applies logs until it reaches a point in the log equal to the log truncation point specified in a point-in-time *conditional restart control record (*SYSPITR CRCR) created with DSNJU003. You cannot explicitly name the backup to use for recovery. That is implicitly determined by the log truncation point used to create the SYSPITR CRCR.

Similar to the Recovery utility, Restore can optionally use the Fast Log Apply (FLA) option.

The RESTORE SYSTEM utility uses the RBLP stored in the header page of DBD01 and updated by the BACKUP SYSTEM utility as the log scan starting point. The log apply phase uses Fast Log Apply to recover objects in parallel. DB2 handles table space and index space creates, drops and extends, and marks objects that have had LOG NO events as RECP (table spaces and indices with the COPY YES attribute) or RBDP (indices with COPY NO attribute).

An informational message will be issued to let the user know if there are any objects that need additional recovery.

If you want to restore a system to the point at which a backup was taken, do not use the RESTORE SYSTEM utility. Use DFSMShsm FRRECOV COPYPOOL(cpname) GEN(gen) to restore both *database* and *log* copy pools. Then start DB2 which will use normal restart recovery processing to back out inflight URs.

One option is available to you when running the RESTORE SYSTEM utility:

- LOGONLY: This indicates that the database volumes have already been restored outside of DB2, so the RESTORE SYSTEM only applies the outstanding log changes to the database. Use the LOGONLY option to recover a DB2 subsystem or a data sharing group to a previous point-in-time.

When the utility is used with the LOGONLY keyword DB2 skips the call to DFSMShsm, assumes the data was restored by another method, and executes only the log apply phase.

In general, to restore a system to a prior point-in-time with the RESTORE SYSTEM utility, use the following steps:

- Stop DB2. If data sharing, stop all members.
- Use DSNJU003 to create a SYSPITR CRCR specifying the point to which you wish to recover the system. If data sharing, create a SYSPITR CRCR for each member.
- If data sharing, delete all coupling facility structures.
- Start DB2. If data sharing, start all members of the data sharing group.
- DB2 will enter into system recover-pending mode, ACCESS(MAINT), and will bypass recovery except for indoubt URs.
- Execute the RESTORE SYSTEM utility. The utility must be completed on the original submitting member.
- Restores the most recent *database copy pool* version that was taken prior to the log truncation point.
- Performs log apply function.
- If a method other than the BACKUP SYSTEM utility was used to copy the system, restore the data manually and use RESTORE SYSTEM LOGONLY.
  - This option can be used with z/OS V1R3 and above.
  - Backs up data with another volume dump solution and uses SET LOG SUSPEND/RESUME.
  - Performs log apply function only.
- Stop DB2 to reset system recovery-pending status. If data sharing, stop all members.
- Display and terminate any active utilities.
- Display restricted objects and recover objects in RECP status and rebuild objects in RBDP status.

The SET LOG SUSPEND/RESUME commands can be used with other volume backup solutions if BACKUP SYSTEM isn't available. DFSMShsm copy pools can be used to simplify the backup process if you are on z/OS V1R5. The SET LOG SUSPEND command is more disruptive then the BACKUP SYSTEM utility because it halts all update activity on a subsystem. Update activity is resumed only when the SET LOG RESUME command is issued. If data sharing is active both commands must be entered on all active members of the data sharing group. The SET LOG SUSPEND command updates the RBLP in DBD01 so that recovery with RESTORE SYSTEM LOGOONLY is possible.

Important: No other utilities can run while the RESTORE SYSTEM utility is running.

Figure 6-4 illustrates what occurs during a system level point-in-time recovery with the RESTORE SYSTEM utility. DB2 recognizes the log truncation point in the SYSPITR CRCR, checks the BSDS to determine which backup to use, calls DFSMShsm to restore the correct *database copy pool* version, gets the RBLP from DBD01, then scans the logs and applies log records until reaching the log truncation point.

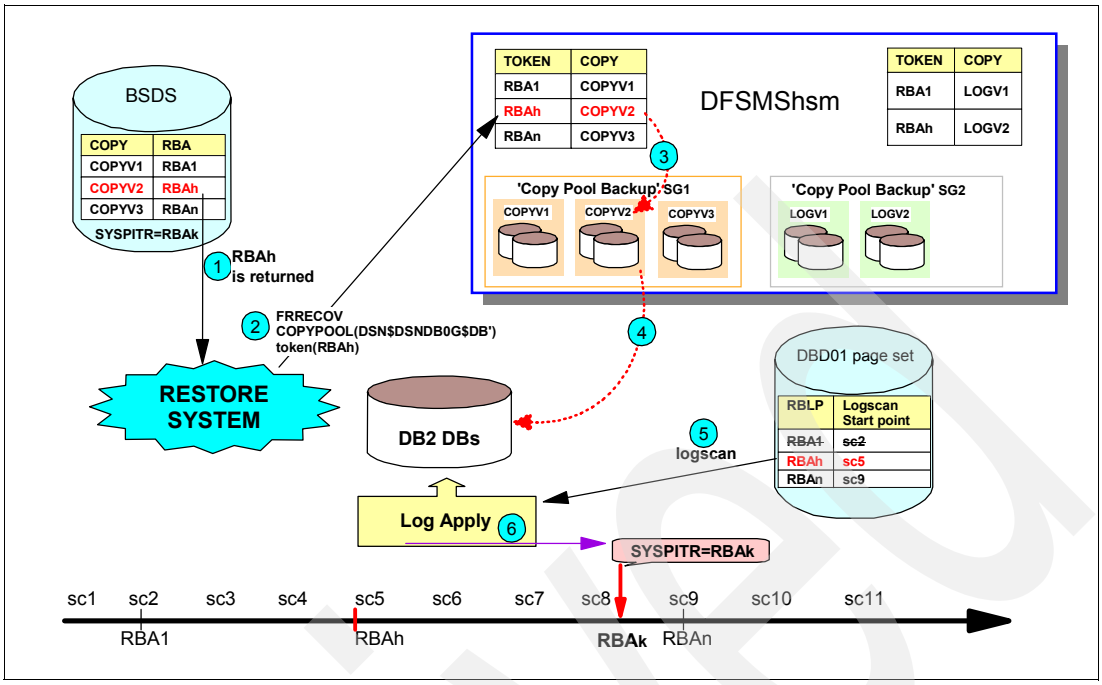


Figure 6-4 Recovering to an arbitrary point-in-time using the RESTORE SYSTEM utility

6.5.2 Running RESTORE SYSTEM

You cannot specify a backup version with the RESTORE SYSTEM utility. RESTORE SYSTEM uses the latest version before the log truncation point. You can specify the log truncation point with the CRESTART SYSPITR option of the DNJU003 stand-alone utility.

Example 6-10 shows the JCL and control statements required for the RESTORE SYSTEM.

Example 6-10 Sample JCL of RESTORE SYSTEM utility

```
//STEP1    EXEC DSNUPROC,SYSTEM=DB8B,UID=DIAG
//*        UTSTATS=''
//SYSIN    DD  *
  RESTORE SYSTEM
```

The LOGONLY option can be used. This indicates that the database volumes have already been restored outside of DB2, so the RESTORE SYSTEM only applies the outstanding log changes to the database. Use the LOGONLY option to recover a DB2 subsystem or a data sharing group to a previous point-in-time. See Example 6-11.

Example 6-11 Sample JCL of RESTORE SYSTEM with LOGONLY option

```
//STEP1    EXEC DSNUPROC,SYSTEM=DB8B,UID=DIAG
//*        UTSTATS=''
//SYSIN    DD  *
  RESTORE SYSTEM LOGONLY
```

You cannot specify a backup version with the RESTORE SYSTEM utility. RESTORE SYSTEM uses the latest version before the log truncation point. You can specify the log truncation point with the CRESTART SYSPITR option of the DNJU003 stand-alone utility.

Before running RESTORE SYSTEM

Complete the following steps prior to running the RESTORE SYSTEM utility:

- Stop DB2.
- RUN DSNJU003 (Change Log Inventory) with the CRESTART SYSPITR option. For SYSPITR, specify the log truncation point that corresponds to the previous point-in-time to which the system is being recovered.
- Start DB2.

After running RESTORE SYSTEM

Complete the following steps after running the RESTORE SYSTEM utility:

- Stop DB2 to reset system RECOVER-pending status.
- Use the DISPLAY UTILITY command to see if any utilities are running. If other utilities are running, use the TERM UTILITY command to end them.
- Use the RECOVER utility to recover all objects in RECOVER-pending or REBUILD-pending mode, or use the REBUILD INDEX utility to rebuild objects.

Refer to the redbook, *DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There*, SG24-7088, for further details on necessary steps before running RESTORE SYSTEM, and necessary steps after running RESTORE SYSTEM.

Displaying RESTORE SYSTEM

You can use the DISPLAY UTILITY command for RESTORE SYSTEM. You must issue the command from the member on which the RESTORE SYSTEM utility was invoked.

Terminating and restarting RESTORE SYSTEM

The RESTORE SYSTEM utility cannot be terminated by using the TERM UTILITY command.

You can restart the RESTORE SYSTEM utility at the beginning of a phase, or at the current system checkpoint. You can reissue a RESTORE SYSTEM job with the same options, and the utility determines which phase to restart.

Sample DB2 messages for DB2 RESTORE

Example 6-12 shows an example of the output you should see from a successful execution of the RESTORE SYSTEM utility.

Example 6-12 RESTORE SYSTEM utility output

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RESTORE
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RESTORE SYSTEM
DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BA140298CDE3D14200120CDAC090'.
DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BA140298CDE3D14200120CDAC090'
                       ELAPSED TIME = 00:00:04.
DSNU1604I -P870 DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT = X'00120CDAC090'.
DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME = 00:04:54.
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Sample DFSMShsm messages for DB2 RESTORE

The DFSMShsm messages generated for the RESTORE SYSTEM utility show the copy pools that are being restored, and the volumes involved with the restore process, see Example 6-13

Example 6-13 FRRECOVER messages in DFSMShsm log

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8B$DB, AT 16:38:50
ARC1801I (CONT.) ON 2003/07/31
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION RECOVERY OF COPY POOL
ARC1805I (CONT.) DSN£DB8B£DB
ARC1805I (CONT.) MHL125
ARC1805I (CONT.) MHL126
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8B$DB, AT 16:38:51
ARC1802I (CONT.) ON 2003/07/31, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

A sample procedure is listed in B.2, “Sample scenario: Restoring to an arbitrary PITR” on page 468. Details on executing procedures for restoring to an arbitrary PITR are included in the scenarios of Chapter 19, “Local recovery: System PITR” on page 363.

7

Peer-to-Peer Remote Copy

Peer-to-Peer Remote Copy (PPRC) is a building block of several disaster recovery solutions. PPRC is a hardware-based disaster recovery solution designed to provide real-time mirroring of logical volumes within an ESS or between two distant ESSs. PPRC is an optional feature on the ESS.

There are several forms of PPRC, which we briefly describe here. More details are available from *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04.

We describe the basic functionality and characteristics of each to provide you with enough background that you will understand the scenarios in which they are used. If you are already familiar with the general concepts, you may skip this section and go directly to the solutions described in Part 3, “General solutions for disaster recovery” on page 129.

In this chapter we provide a basic description of the following Peer-to-Peer Remote Copy technologies:

- Synchronous PPRC, now called Metro Mirror
- PPRC - Extended Distance (PPRC-XD) also called Global Copy
- Asynchronous PPRC, also called Global Mirror

7.1 Peer-to-Peer Remote Copy (PPRC)

The synchronous PPRC implementation (PPRC-SYNC) is a synchronous remote copy solution where write operations are completed on both copies (primary and secondary ESS) before they can be considered to be done. Thus, the recovery data at the remote site will be a constant real time mirror of the data at the local site as DB2 (and other subsystems and applications) perform updates.

7.1.1 How PPRC works

PPRC is a hardware based solution that enables the mirroring of application system data from one site (the application site) and its associated DASD volumes (the primary volumes) to a second system at another site (the recovery site) and its DASD volumes (the secondary volumes).

The characteristics of Synchronous PPRC (Metro Mirror) are summarized in Figure 7-1.

PPRC Synchronous is a hardware technology that can be used at Metro distances to provide a remote copy of data without data loss.

It is time consistent for dependent writes within one LSS or LCU but not across them or DASD subsystems natively.

BSDS and DB2 logs at least should be on two separate DASD subsystems.

- It can be used within Metro distance (<103 KM)
- Local DB2 write operations can have overhead, depending on distance between sites
- Critical attribute should not be enabled for DB2
- Stand alone PPRC is exposed to rolling disaster (lack of consistncy of LSS volumes at recovery site)
- Requires other technologies to guarantee data consistency at recovery site
- RPO is zero

Figure 7-1 Metro Mirror features

Updates made on the primary DASD volumes are synchronously shadowed onto the secondary DASD volumes. The DB2 write operation is only completed when the data is secure on the secondary site ESS. Figure 7-2 illustrates the sequence of a write update with PPRC (synchronous mode).

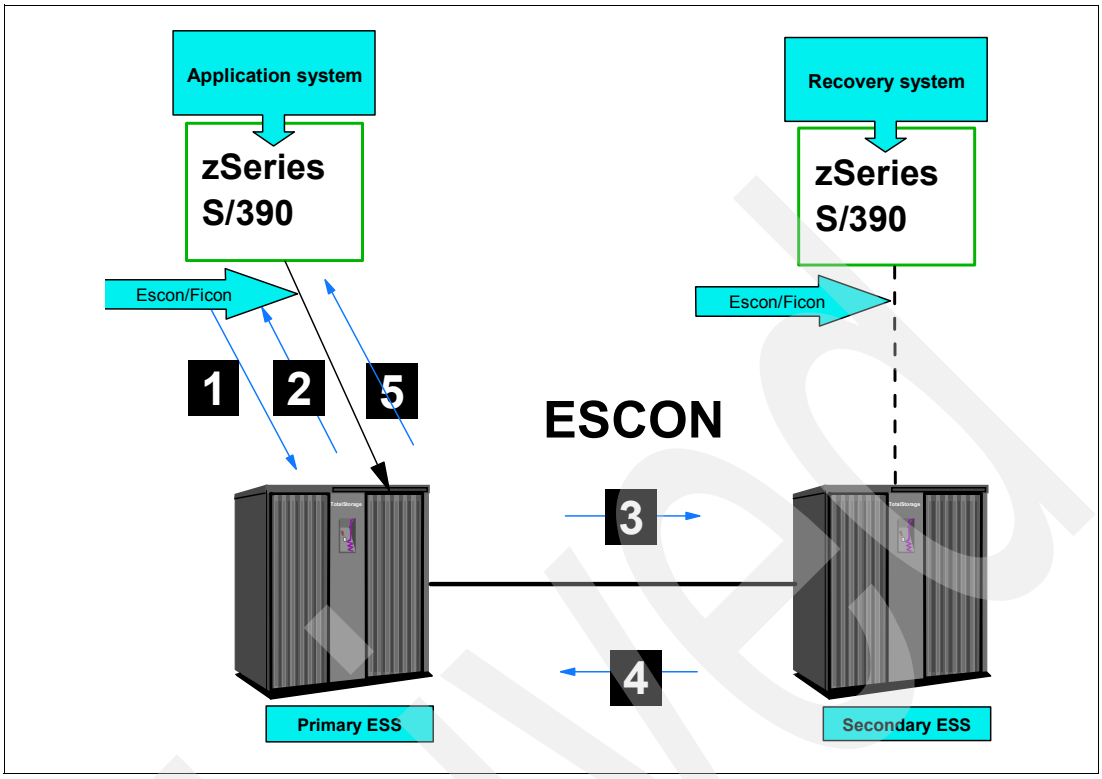


Figure 7-2 PPRC data flow

When DB2 performs a write update operation upon a primary volume, this is what happens:

1. Write to primary volume, on ESS cache and non-volatile storage (NVS):

   DB2 writes data to a primary volume on the ESS, and cache hit occurs.

2. Disconnect from channel (CE):

   Once the data has been transferred to the primary ESS's cache and NVS, the *channel end* status is issued to DB2.

3. Write to secondary (ESS cache and NVS):

   The local site ESS then initiates an I/O channel program to the recovery site ESS, to transfer the updated data to the recovery site cache and NVS. This technique ensures that performance for the synchronous write is optimized.

4. Signal write complete on the secondary:

   The recovery site ESS signals write complete to the local site ESS when the updated data is in its cache and NVS.

5. Post I/O complete (DE):

   When the local site ESS receives the *write complete* from the recovery site ESS, it returns *device end* status to DB2.

Note: The movement of data from cache to the disk drives on both the local and recovery site ESSs is performed asynchronously.

If, during the process, an error occurs that prevents the secondary volume from being updated, PPRC automatically suspends the mirroring function on that volume, and a set of accompanying actions are triggered to ensure data consistency at the recovery site. The

series of resulting actions will depend on certain critical attribute and consistency group options defined at implementation time (these recovery options are discussed later in this chapter).

7.1.2 DB2 and critical attributes

Data currency between sites can be achieved by putting all database volumes in PPRC pair relationships. It must be carefully considered what would happen if a remote copy pair fails. This option can stop applications if all primary and secondary volumes cannot be kept in sync. PPRC alone places more value on data currency than application availability.

You have a choice of two PPRC attributes that can be specified on a volume *pair* basis. There are definite implications for DB2 on either choice. You specify the desired option desired with the TSO command CESTPAIR with the parameter CRIT(Y) or CRIT(N). When using the Web User Interface, this option is chosen by selecting the right parameter in the panel Copy Options - Critical Volume Mode, shown in Figure 17-10 on page 309.

Critical attribute enabled

When the critical option is enabled for a volume pair, the primary volume returns a permanent error to all I/O writes in case the secondary PPRC volume cannot be updated. While less robust applications usually terminate, DB2's internal recovery allows processing to continue, which could lead to loss of availability locally.

DB2 continues locally and takes these actions:

- Active log: DB2 marks the current active log unavailable, and switches to the next active log.
- BSDS: DB2 marks the BSDS unavailable, and switches to single BSDS mode.
- Database (including DB2 catalog/directory): DB2 sets the pages in the Logical Page List (LPL) and makes them unavailable for access.

To recover objects in LPL, you only need to issue a -START command for the affected objects. DB2 reads the log, then reads the pages for the object and applies them. This is usually completed in a few seconds to minutes, depending on how many objects are affected and how much time has gone by since the error had occurred.

To maintain DB2 integrity at the secondary system, all volumes that are used by DB2 must be in PPRC pairings. In an SMS environment you can avoid this problem by ensuring that all volumes in a particular SMS storage group are in PPRC pairings. All data will therefore continue to be reflected at the recovery site.

If a volume has been logged as needing recovery, some tables and partitions may be affected. If the error is due to all paths failing between a ESS pair, a growing number of pages and table spaces could be flagged as subsequent writes encounter the PPRC permanent errors. As a result, the application primary site will require many LPL recoveries at the minimum. The disk storage error recovery procedure (ERP) issues an `IEA491E` message for each pair that suspends to the system initiating the I/O, and an `IEA494I message` is broadcast to any of the attached systems where the volume is online.

Since any failure to write to the secondary PPRC volume causes this error, it might not indicate that a disaster is in progress. It could occur for any damage to the communication lines. Regardless, there is now a local loss of availability at the least. Additionally, volume pairs have no guaranteed consistency across LSSs. For this reason we recommend against having the critical attribute enabled for all DB2 users.

Critical attribute not enabled

When the critical attribute is not enabled, the primary volume will not return an error condition to the I/O writes — even if the secondary volume cannot be successfully updated. PPRC will be aware of the problem, but DB2 continues to update the primary volumes.

Though it is not passed to DB2, the DASD ERP receives the write error from the storage controller. The sense information received identifies the error as a write failure to the secondary device, and a permanent error is *not* issued on the primary. The DASD ERP issues an `IEA491E` message for each pair that suspends to the system initiating the I/O, and an `IEA494I message` is broadcast to any of the attached systems to which the volume is online.

The ESS suspends the PPRC pairing. All subsequent writes to the primary should succeed as the copy to the secondary is not attempted. ERP retries the failed write operation. DB2 continues locally with minimal impact. The secondary volume no longer reflects any updates and is subsequently out of synchronization with the primary and other secondary volumes.

The rolling disaster at the secondary site occurs with CRIT(N).

7.1.3 Rolling disaster with PPRC alone

Regardless of the setting of the critical attribute, PPRC alone will experience a rolling disaster. With the critical attribute enabled, the local site will also experience a loss of availability.

Note: We recommend you specify CRIT(N) for all DB2 volume pairs for availability.

Therefore, *something else* is needed to assure consistency when PPRC is used for disaster recovery. It can be one of these:

- Geographically Dispersed Parallel Sysplex, also called Metro Mirror. It has the capability to “freeze” using PPRC commands each LSS in the entire enterprise, thus guaranteeing consistency at the secondary site. It is described in Chapter 14, “Geographically Dispersed Parallel Sysplex” on page 233.
- PPRC Consistency Groups (called CGROUP)

7.1.4 Consistency grouping and messages issued

The CGROUP parameter of the CESTPATH TSO command is used to control error situations and to maintain consistency at the secondary site.

When specifying CGROUP(YES) on the CESTPATH command (or, its equivalent with the ESS Copy Services WUI), the disk storage subsystem ERP (error recovery procedures) will put the primary LSS in an extended long busy (ELB) wait condition if PPRC cannot update a secondary volume. Also, the ELB state condition will be reported on message `IEA494I`.

These are the messages issued:

```
IEA494I 1162,LIVI62,PPRC PAIR SUSPENDING,SSID=00B0,CCA=02, EXTENDED LONG BUSY STATE
IEA494I 1162,LIVI62,PPRC PAIR SUSPENDED,SSID=00B0,CCA=02, EXTENDED LONG BUSY STATE
```

IEA491E message

When PPRC detects an error that prevents the update of a secondary volume, the pair is suspended. Message `IEA491E` is issued indicating that an error has occurred, and a possible reason is given. `IEA491E` is returned to any system attempting an I/O to the failed volume pair.

When the error that caused `IEA491I` to be issued has been corrected, the volume pair can be re-synchronized. The following is an example of the **`IEA491E`** message:

```
IEA491E LIVI62,PPRC SUSPENDED, COMMUNICATION_TO_SECONDARY_FAILURE,
(PRI)SER=0339-35570,CCA=02 (SEC)SER=0339-35416,CCA=03 CONTINUATION OF IEA491E
SNS=101010F0C20000FBF2030C69008A580C69008AF28700
```

Automation

Alerted by the PPRC messages, automation functions can detect when outage conditions are occurring. Automation procedures can then perform damage evaluation and limitation, deciding whether a major failure is in progress, or if a transient or an isolated failure has occurred, and then it will proceed to take the necessary actions.

The PPRC Consistency Grouping timer function (known also as the long busy time-out value in the ESS) gives time to the *automation* routines and functions to issue the necessary commands (when a pairing has an error) by putting the primary volume into a long busy state. This state is issued by the ERP and reflected in the **`IEA494I`** messages.

The long busy state temporarily queues I/O to the affected primary volume; thus, dependent writes are held. On the ESS, the creation of Consistency Groups is controlled through the CGROUP parameter of the TSO CESTPATH command, the ANTRQST macro and GPDS, or equivalent options from the ESS Copy Services Web User Interface.

With the Consistency Groups created and the extended long busy window active, the automation functions can do a freeze to temporarily suspend I/O to all the necessary volume pairs at an LSS level. This ensures a point of consistency at the secondary site, across all necessary volumes.

Automated procedures should be set to ensure the consistency of the data across the application volumes in whatever LSS or ESS they reside. These procedures have to be able to act on the PPRC configuration according to the error.

GDPS, which is discussed in Chapter 14, “Geographically Dispersed Parallel Sysplex” on page 233, implements all necessary functions for an automatic recovery, with PPRC integrated into this solution.

7.2 PPRC Extended Distance (PPRC-XD)

PPRC-XD, also called Global Copy, is a recommended solution for remote data copy and off-site backup without impacting local DB2 (and other subsystems) write performance, which is particularly relevant when implemented over continental distances. PPRC-XD can also be used for DB2 disaster recovery solutions based on periodic point-in-time copies of the data.

Figure 7-3 illustrates a general picture of a PPRC-XD configuration involving two sites with two sets of DASD at the secondary site.

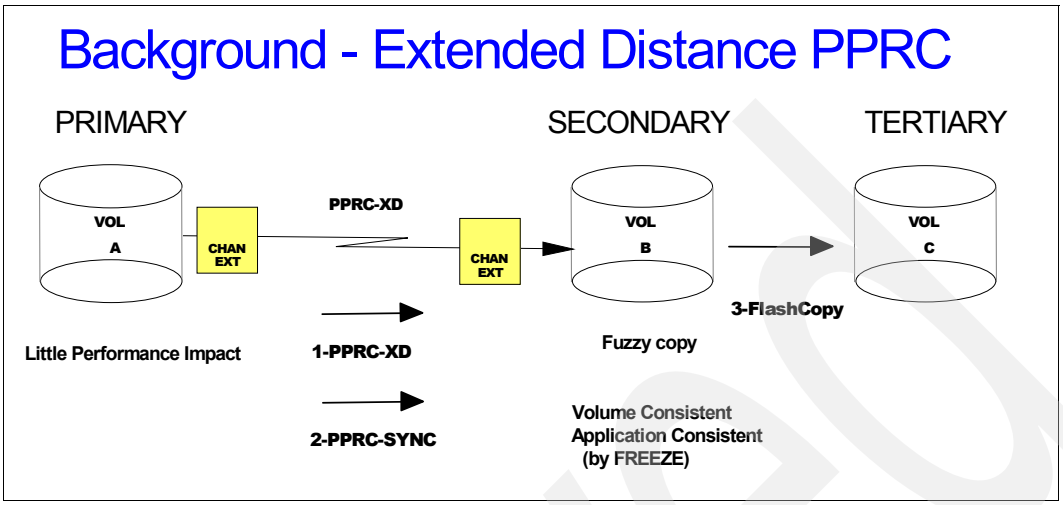


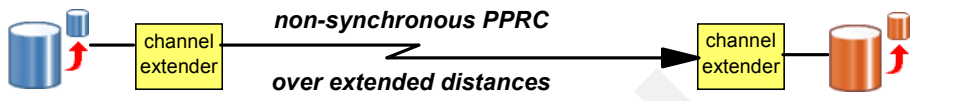
Figure 7-3 PPRC Extended Distance background

PPRC-XD provides a non-synchronous long-distance copy option whereby write operations to the primary ESS are considered complete before they are transmitted to the secondary ESS. This non-synchronous operation results in a “fuzzy copy” at the secondary site. Through operational procedures such as the DB2 -SET LOG SUSPEND command, you can create a point-in-time (PIT) consistent copy at the remote site on the secondary volumes. They can be FlashCopied to a tertiary set that is suitable for disaster recovery purposes.

PPRC-XD can operate at very long distances (distances well beyond the 103 km supported with PPRC synchronous transmissions) with the distance typically limited only by the capabilities of the network and channel extension technologies.

7.2.1 Characteristics

In this section we discuss the basic characteristics of PPRC-XD, summarized in Figure 7-4.



- XD option when establishing a volume pair
  - XD option (extended distance) in addition to the traditional synchronous option (SYNC)
  - Long distance implementations when using channel extenders
- Non-synchronous copy
  - Write updates to primary receive immediate completion
  - Primary ESS keeps record of updates to primary volumes
  - Incremental copy of primary updates periodically sent to secondary
  - Less demanding bandwidth requirements compared to SYNC option
  - No synchronous-like overhead on DB2 local response time
  - Continential distance with channel extenders
  - Maintains a fuzzy copy on secondary volumes until forced to synchronous
    - No certainty for dependent writes mirroring sequence
  - User creates consistency group with -SET LOG SUSPEND
    - Writes stop and remaining tracks arrive quickly at secondary

Figure 7-4 Characteristics of PPRC Extended Distance

Non-synchronous option of PPRC

The XD option brings more functionality into the PPRC environment, changing the paradigm that PPRC is only synchronous. PPRC-XD will be used in different implementation scenarios than those of traditional Synchronous PPRC.

When doing the initial copy (establishing a new PPRC pair), or when re-synchronizing (RESYNC) a previously suspended volume pair, there are two options you can specify:

- **SYNC** This option is used for synchronous mirroring. The initial copy of a pair is completed up to the *duplex state*, and from there on updates to the primary volumes are mirrored *synchronously* on the secondary volumes. A complete status is not returned to the application until both writes are done. As a consequence, there is an overhead which becomes progressively more significant at longer distances. At longer distances, the overhead in the I/O response time may be onerous. While some applications may need to issue this command to obtain consistency, DB2 does not need to use this capability, as it has commands that can be used to suspend and resume activity.
- **XD** This option is used for *non-synchronous* transfer of updates over extended distances. In this method of PPRC operation, when doing the initial copy of a pair, or when a suspended pair is re-synchronized, the volumes do not reach the *duplex state,* but remain in *duplex pending*-XD state. While in duplex pending-XD state, the DB2 updates to the primary volumes receive a complete status without waiting for the secondary ESS to acknowledge — as opposed to the Synchronous PPRC. The primary ESS keeps record of the updates to the primary volumes, and the updates are periodically sent in batches to the secondary volumes.

Write operations: Response time

PPRC-XD uses a non-synchronous mirroring technique, and so it does not incur synchronous-like overheads for DB2 I/O write operations. The write operations are finished as soon as the updates are secured in the primary ESS non-volatile storage (NVS) as in normal non-PPRC operation.

This characteristic makes PPRC-XD recommended for solutions needing a minimum impact on the local DB2 I/O response time, especially when extended distances are to be considered.

Distance factor

The non-synchronous technique of PPRC-XD relieves DB2 write operations from the typical synchronous mirroring overhead, which is directly proportional to the distance between the application and the recovery sites.

This characteristic makes PPRC-XD a recommended solution when planning for implementations over extended distances — continental distances, well beyond the synchronous 103 km ESCON (300 km FCP) supported distance. The distance is limited only by the capabilities of the network and the channel extenders' technologies.

Fuzzy copy: dependent writes

The PPRC-XD non-synchronous mirroring technique does not guarantee that DB2 dependent writes are copied in the same sequence as they have been applied onto the primary volumes; the secondary copy is *fuzzy*. Dependent writes are explained under 12.1.1, "Terminology" on page 166.

CRIT attribute

As XD mirroring is done non-synchronously, with the secondary being a fuzzy copy, any CRIT setting is invalid for volume pairs established with the XD option.

7.2.2 Creating a Consistency Group with PPRC-XD

Lets examine the considerations that come into play in a PPRC-XD environment, when the consistency of the secondary data is considered.

While in the *duplex-pending XD* state (the normal DB2 operational mode), and with DB2 writing to the primary volumes, the recovery site volumes are keeping a *fuzzy copy* of the data. This means that during this state, there is no certainty that the DB2 *dependent writes* will be applied to the secondary volumes in the same sequence as they are written to the primary volumes.

With PPRC-XD you can create a "consistency group" across the secondary copies of the DB2 set of volumes, if you implement the appropriate checkpoint activities in order to build this logical consistency. The *catch-up* transition is part of this checkpoint procedure.

The *catch-up* transition allows you to synchronize the volume pairs in a minimum interval of time.You issue a DB2 command -SET LOG SUSPEND which allows all outstanding tracks to be sent to the recovery site as there are no further I/O. Upon reaching duplex state, the volume pairs can be temporarily suspended, followed by a FlashCopy of the secondary volumes onto tertiary volumes (which permanently captures the point of consistency), and then resuming the PPRC-XD relationships.

When planning the recovery of DB2 using a point-in-time copy of the data, you must remember that while in an active PPRC-XD relationship, the secondary volumes always have a fuzzy copy of the primary volumes. So, for a valid recovery you must obtain the tertiary volumes where you did the last FlashCopy from secondary volumes at the point of consistency. As you can realize, this tertiary copy of the data does not reflect the updates to the primary volumes done after the last global *catch-up* operation.

Managing PPRC-XD recovery

When PPRC-XD is used for DB2 recovery solutions, automated procedures can be implemented in order to efficiently execute the steps involved in building DB2 consistency at the recovery site. You will find the state change messages that the system issues when volumes transition from one PPRC state to a different one very useful. The **IEA494I** and **IEA491E** messages can be detected by automation routines and used to suspend secondary PPRC operations. These messages occur for volumes and not on an LSS basis.

When scheduling the quiesce of your DB2 writes, you will also need to consider some lead time to build consistency of the recovery data. This planned outage can be minimized when properly automated, and the whole process can prove to be very efficient when recovering an application after its outage.

When making your secondary volumes fully synchronous with their respective primaries, the command -SET LOG SUSPEND is DB2-specific. In data sharing, you need to issue commands on all members of the group. For other applications or subsystems for which you want a point of consistency with DB2, you can use the PPRC command to go-to-SYNC or to quiesce the application in whatever manner is supported by it. Those options are not described in this manual.

The RPO is the amount of time that has passed since the last FlashCopy to the tertiary volumes.

7.3 Asynchronous PPRC: Global Mirror

PPRC Version 2 provides capabilities that can be used in a DB2 disaster recovery solution. Asynchronous PPRC, also called Global Mirror, is designed to provide a long-distance remote copy solution across two sites using asynchronous technology. It operates over high-speed, Fibre Channel communication links and is designed to provide a consistent and restartable copy of the data at the remote site, created with minimal impact to applications at the local site. It eliminates the requirement to periodically suspend DB2 at the local site in order to create a consistency group at the remote site. See Figure 7-5.

Asynchronous PPRC (Global Mirror) is a hardware technology that can be used at continental distance across multiple ESS subsystems to produce data consistent secondary volumes suitable for DB2 disaster recovery with an RPO of seconds.

DB2 Active logs should reside on separate DASD subsystems.

- It can be used at continental distance
- Local DB2 write operations can have negligible overhead, regardless of distance between sites
- Uses combination of PPRC-XD across distance and FlashCopy at the secondary site to enable consistency among all DB2 volumes
  - No rolling disaster
- Not DB2 specific
  - Supports open systems too
- Requires tertiary set of DASD
- RPO is > zero, about 5 seconds

Figure 7-5 Global Mirror features

Asynchronous PPRC is designed to provide a long-distance remote copy solution across two sites for open systems and z/OS data using asynchronous technology. It is based on existing Copy Services functions PPRC-XD and FlashCopy and additional new functionality for the periodic creation of Consistency Groups across one or more ESSs and for the management and control of the Asynchronous PPRC session.

It operates over high-speed Fibre Channel communication links and supports distance typically limited only by the capabilities of the network and channel extension technologies. This provides a wider choice of remote site locations based on business needs. It also enables site separation to add protection from localized disasters.

The data at the remote site is maintained to be a consistent and application restartable copy of the data at the local site without significant impact to applications at the local site. Compared to PPRC-XD, Asynchronous PPRC provides automation to periodically create Consistency Groups at the primary and the secondary site.

Data currency at the remote site lags behind the local site typically 3 to 5 seconds (when the Consistency Group Interval Timer is set to 0 seconds under a typical workload), minimizing the amount of data exposure in the event of an unplanned outage. Data currency is also known as the recovery point objective (RPO). The actual lag in data currency experienced will depend upon a number of factors, including specific workload characteristics and available bandwidth between the local and remote sites.

Data consistency at the remote site is internally managed across up to four (current limit) primary ESS machines located at the local site.

Given a known primary workload and the available bandwidth to the remote site, a timer variable in the Asynchronous PPRC session can dynamically be adjusted to meet the Recovery Point Objective of the customer.

7.3.1 Prerequisites for Asynchronous PPRC

- ESS model 800 or 750 for the primary and secondary subsystems.
- PPRC V2 for primary and secondary ESSs.
- FCP PPRC links.

- ESS LIC 2.4.0 minimum.
- Existing ESS machines do not require the new feature code.
- FlashCopy V2 required at the remote site.

Any site with PPRC V2 that uses FlashCopy must use FlashCopy V2.

7.3.2 Terminology

With Asynchronous PPRC the data consistency at the secondary site is guaranteed in comparison to PPRC-XD where a disruption is required to create consistency and take remote copies for each point-in-time. The critical aspect for insuring data consistency is how a consistency group can be created in any disaster recovery solution. The following terms are important so that you will understand how this occurs.

Session

An Asynchronous PPRC Session is a collection of volumes that are managed together when creating consistent copies of data. This set of volumes can reside in one or more LSSs and one or more ESSs at the customer's primary site. Open systems volumes and z/OS volumes can both be members of the same session.

When you start or resume a session, the creation of Consistency Groups is performed and the Master ESS controls the session by communicating with the Subordinate ESSs.

Master

The Master is a term for a function inside an ESS that, by communicating with Subordinates in other ESSs, controls the creation of Consistency Groups and manages the Asynchronous PPRC Session. The Master is defined in the start command for a session by addressing any LSS in the Master ESS.

The Master needs PPRC Fibre Channel communication paths to any one of the LSSs in each Subordinate ESS.

Subordinate

The Subordinate is a term for a function inside an ESS that communicates with the Master and is controlled by the Master. At least one of the LSSs of each Subordinate ESS needs PPRC Fibre Channel communication paths from the Master to the Subordinate. It is used to enable the Master to create Consistency Groups of volumes in more than one ESS.

If all the volumes of an Asynchronous PPRC Session reside in one primary ESS no Subordinate is required, because the Master can communicate to all LSSs inside the primary ESS.

Consistency Group

This is a group of volumes in one or more ESSs whose data must be kept consistent at the remote site.

> **Note:** With Asynchronous PPRC this is done periodically and automatically without disruption to the host I/O in a very short time.

7.3.3 Asynchronous PPRC: How it works

The purpose of Asynchronous PPRC is to provide recoverable copies of production data at a long distance remote site by continually forming consistent sets of data with no significant performance impact. This allows a quick restart at the remote site in case of a disaster at the primary site.

Basics of Asynchronous PPRC

In contrast to the PPRC-XD (Global Copy) technology described in 7.2, “PPRC Extended Distance (PPRC-XD)” on page 102, Asynchronous PPRC (Global Mirror) overcomes its disadvantages:

- Unacceptable currency of the remote site data for time critical applications
- DB2 outage from suspending the log while volumes are becoming consistent
- Several separate manual steps to manage

It also automates this technology, so that you can get:

- A very good data currency of only a few seconds
- Automatic creation of Consistency Groups in a very short time
- No application impact at the local site
- One central point of control

Asynchronous PPRC (Figure 7-6) is implemented at Licensed Internal Code level 2.4.x or higher of the ESSs and works as described in 7.1, “Peer-to-Peer Remote Copy (PPRC)” on page 98 for all volumes in one or more LSSs that previously have been added to a defined Session. The creation of the Consistency Group at the local site is a very quick process followed by a drain of the consistent data to the remote site, while DB2 (and other application) I/Os are continuously running.

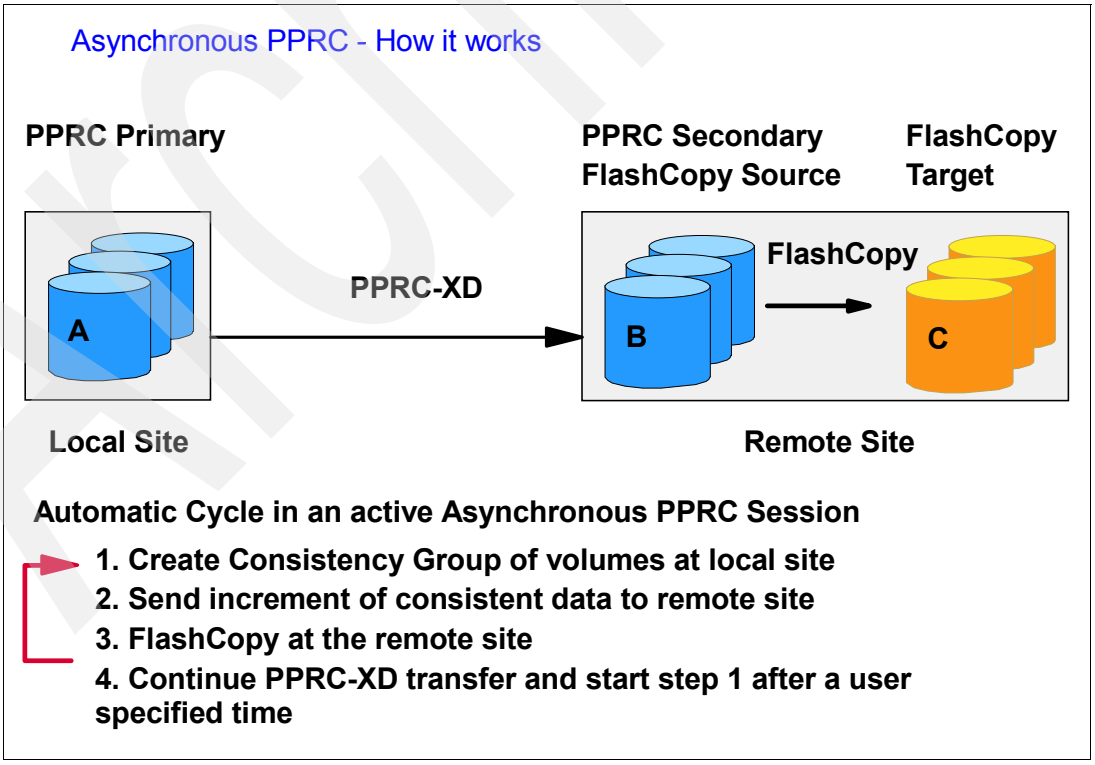


Figure 7-6 Asynchronous PPRC

A persistent FlashCopy with no background copy and active change recording continues immediately after the drain has finished. According to your Recovery Time Objectives (RTO), the available bandwidth and your application workload, you can assign a Consistency Group Interval Timer value before the process starts again at step one.

The currency of the data at the remote site can achieve typically 3 to 5 seconds with a zero second Consistency Group Interval Timer value), but this depends on the workload and bandwidth available to the remote site.

Operational commands and session setup are described in *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04.

Note: The consistent data at the remote site resides on the FlashCopy target volume and not on the PPRC secondary volume.

To enable this very fast cycle across multiple ESSs, a fast inband communication network is required. The configuration example in Figure 7-7 explains the necessary communication across LSSs and ESSs.

One of the local site ESSs is the Master ESS that communicates to Subordinate ESSs over fiber channel links to enable the creation and control of the Consistency Groups.

The volumes in the session can be open systems volumes as well as z/OS volumes. Consistency Groups are created for all volumes in the session at the same time, independently from the volume characteristics.

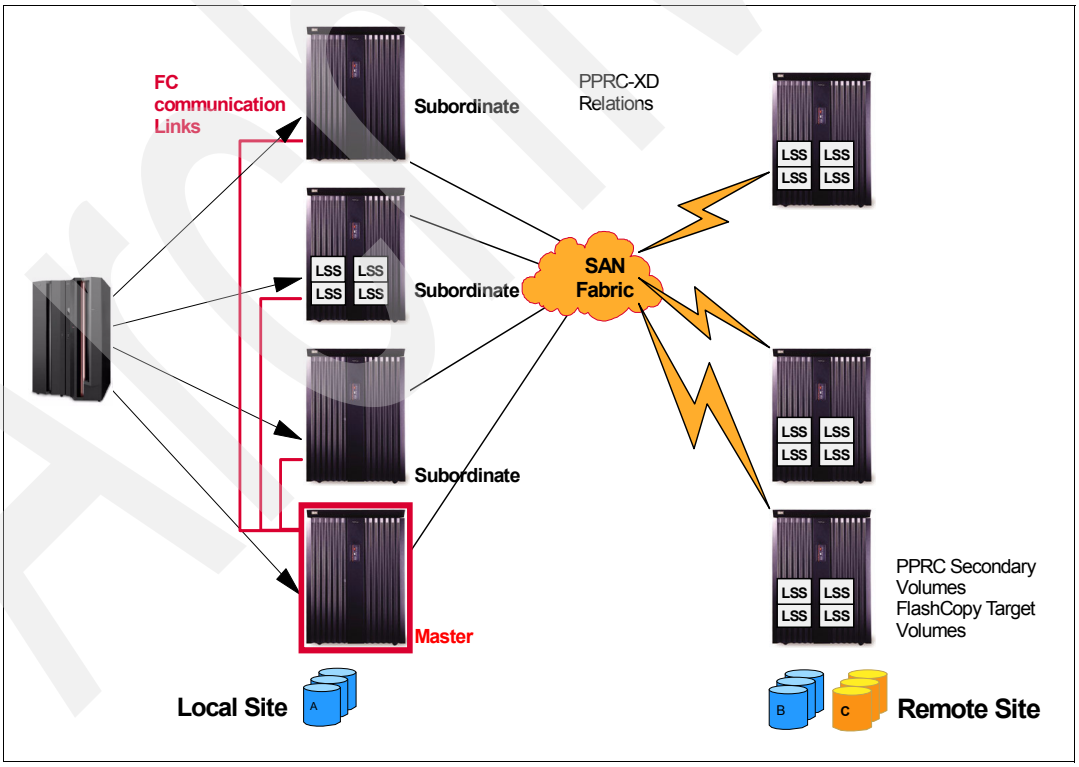


Figure 7-7 Example of an Asynchronous PPRC configuration

Asynchronous PPRC session management

A session is a collection of volumes in an ESS or in multiple ESSs that are managed together when creating consistent copies of data. When you start or resume a session, the Asynchronous PPRC configuration is specified or modified. This causes Asynchronous PPRC to form a Consistency Group that includes each volume in the session. The topology of a session is defined by its Master and Subordinate relationship. An individual Session consists some or all of the volumes in a Master and up to three primary ESS subordinates. There can be a maximum of 8 ESSs in any combination of primaries and secondaries.

It is also possible to establish an Asynchronous PPRC relationship in both directions with primary and secondary volumes at the local and remote sites.

7.3.4 Prerequisites for Asynchronous PPRC

These are the prerequisites for Asynchronous PPRC:

- ESS model 800 or 750 for the primary and secondary subsystems.
- PPRC V2 for primary and secondary ESSs.
- FCP PPRC links.
- ESS LIC 2.4.0 minimum.
- Existing ESS machines do not require the new feature code.
- FlashCopy V2 required at the remote site. Any site with PPRC V2 that uses FlashCopy requires FlashCopy V2.

7.4 PPRC feature dependencies

PPRC is an optional chargeable feature of the ESS product and is available in 2 versions. PPRC Version 1 provides real-time mirroring and extended distance:

- Synchronous PPRC
- PPRC-XD

PPRC Version 2 further supports business continuance solutions with the following additional functions:

- PPRC over Fibre Channel links
- Asynchronous PPRC
- Asynchronous Cascading PPRC

Note that PPRC Version 2 includes all PPRC Version 1 features. Only one of the PPRC features (V1 or V2) can be ordered on the same machine. Also, any other ESS that will contain the PPRC pairs must also have the minimum PPRC feature version required to support the type of Copy Services being used. Also note that FlashCopy Version 1 cannot be ordered with PPRC Version 2 on the same machine.

See Table 7-1 for the minimum allowable LIC levels for the various PPRC features.

Table 7-1 Minimum requirements for PPRC features

PPRC feature	Minimum PPRC License	Minimum ESS Licensed Internal Code		
		Fxx	800	750
Synchronous PPRC	PPRC-V1	1.3.0	2.0.0	2.3.1
PPRC-XD	PPRC-V1	1.5.2	2.0.0	2.3.1
Asynchronous Cascading PPRC	PPRC-V2	2.2.0	2.2.0	2.3.1
Asynchronous PPRC	PPRC-V2	no support	2.4.0	2.4.0
PPRC paths over ESCON	PPRC-V1	1.3.0	2.0.0	2.3.1
PPRC paths over FCP	PPRC-V2	no support	2.3.0	2.3.1

8

eXtended Remote Copy

In this chapter we provide a basic description of eXtended Remote Copy (XRC), also known as the IBM TotalStorage z/OS Global Mirror. Also discussed here are the considerations for its implementation on the IBM TotalStorage Enterprise Storage Server (ESS).

As XRC is used later in our tests, its basic functionality is described here to provide you with enough background that you will understand the testing scenario(s) in which it is used. If you are already familiar with general concepts of XRC, you may skip this chapter. Much of the material in this chapter is based on *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04.

z/OS Global Mirror is both a technology as well as a stand alone solution, as we show in Part 3, “General solutions for disaster recovery” on page 129. It can be incorporated in other solutions, as we describe in Part 4, “Implementing disaster recovery scenarios” on page 249.

When planning to implement and use XRC, the following publications should be referenced to complement the information presented in this chapter:

- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547
- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *DFSMS Extended Remote Copy Installation Planning Guide*, GC35-0481
- *DFSMS Extended Remote Copy Reference Information for Advanced Users*, GC35-0482

8.1 Introduction

XRC is a copy function available for z/OS and z/OS data only. A DFSMSdfp™ component called System Data Mover (SDM) will replicate the writes issued to primary volumes to the secondary devices. XRC maintains a copy of the data at a remote location, because data is sent asynchronously, XRC can be implemented over unlimited distances. It is a combined hardware and software solution that offers data integrity and data availability that can be used as part of the business continuance solutions.

When ESS is used as the primary storage subsystem, XRC Version 3 supports its capability to maintain a hardware bitmap of the tracks changed on primary volumes by the primary systems. If an outage occurs, only the changed tracks need to be copied to the secondary volumes when the connection between the ESS and the SDM is reestablished, and by this, a full resynchronization of the volumes is avoided.

Best connectivity performance is obtained by FICON® channels to connect the SDM to primary ESSs. Other connectivity options, such as Channel Extenders and telecommunications lines, lend themselves to cross continental solutions. Since it is an asynchronous solution, XRC allows greater distance between the primary and secondary sites without impacting primary site applications.

Figure 8-1 depicts all components in an XRC configuration.

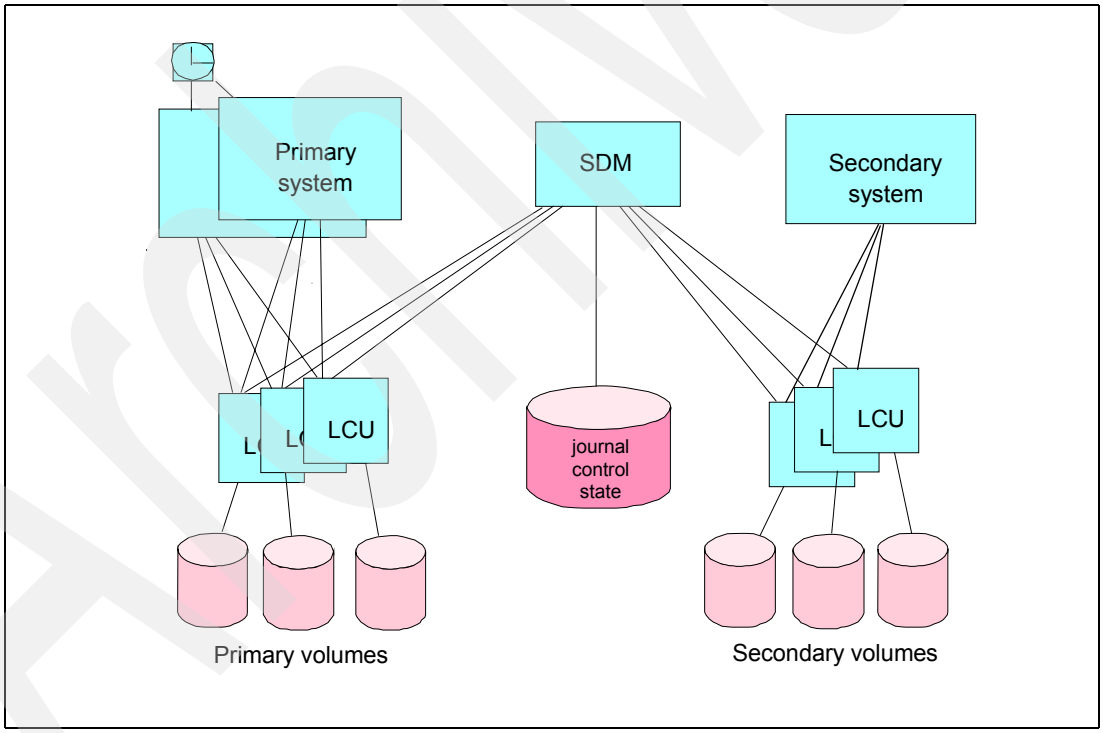


Figure 8-1 XRC configuration

The components are:

- Primary z/OS images, which have a common time reference used to mark their I/O.
- Primary DASD subsystems, which receive the I/O from primary systems and keep them to be replicated on the secondary disk subsystems.

- System Data Mover (SDM), which is part of DFSMSdfp, and is the heart and the brain of the solution. It reads the I/Os kept in the primary subsystems and replicates them in the same time sequence on the secondary disk subsystems.
- SDM Control data sets (journal, control, state, as shown in Figure 8-1 on page 114), which are used by SDM to manage the entire solution.
- Secondary DASD subsystems, which are kept updated by the SDM.
- Secondary z/OS images, which are used in an emergency to provide IT services from the primary site.

Since it is part of DFSMSdfp, SDM can run at any site where a z/OS system can run. Most XRC users choose to run the SDM at the secondary site because:

- The same processor used to run SDM can be used to restart IT services from the alternate site, once the SDM recovery process is finished.
- This configuration minimize the connectivity required. SDM needs to access remotely to primary disk only, but it uses a special protocol for the data transfer from primary disks which transfers multiple records with a single read operation.
- Data stored in SDM buffers and not yet hardened on Journals is also safe in case of a disaster, and this helps in containing the data loss.

Attention: Regardless of where the SDM runs, its Journal, Control, and State data sets must be located at the secondary site, as they are required by SDM to recover the secondary volumes.

8.1.1 XRC configurations

Figure 8-2 depicts a basic XRC configuration where there is a single SDM to manage the disk mirroring between primary and secondary ESSs. With this configuration you are able to manage up to 1500-1800 primary volumes, but larger configurations are possible using more SDMs.

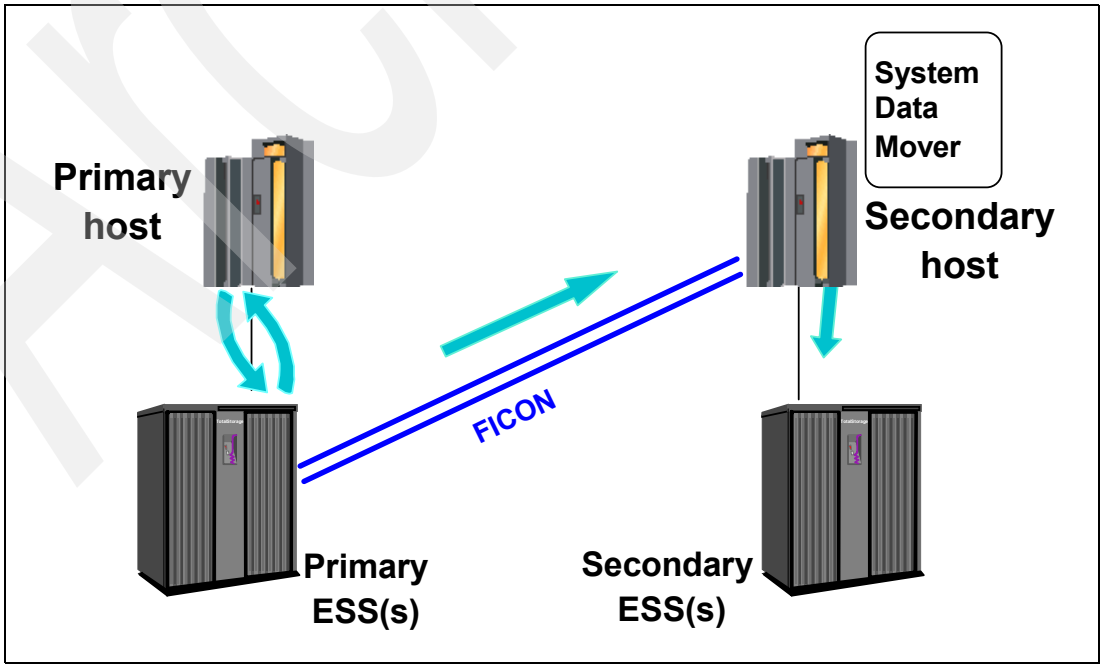


Figure 8-2 XRC basic configuration

*Multiple Extended Remote Copy (*MXRC) allows you to run up to five XRC sessions within a single LPAR; the sessions may or may not be coupled.

Coupled Extended Remote Copy (CXRC) configuration

Coupled Extended Remote Copy (CXRC) allows you to manage very large installations that have configurations consisting of thousands of primary volumes. In a *Coupled Extended Remote Copy* (CXRC), the multiple SDMs coordinate their Consistency Group processing such that the recovery of all secondary volumes can be done to a single point of consistency. Up to 14 XRC sessions can be coupled to a single master session.

Figure 8-3 shows a possible CXRC configuration where four SDMs are coupled together to provide a common consistent point-in-time for all the data across all the primary volumes that make up the application set.

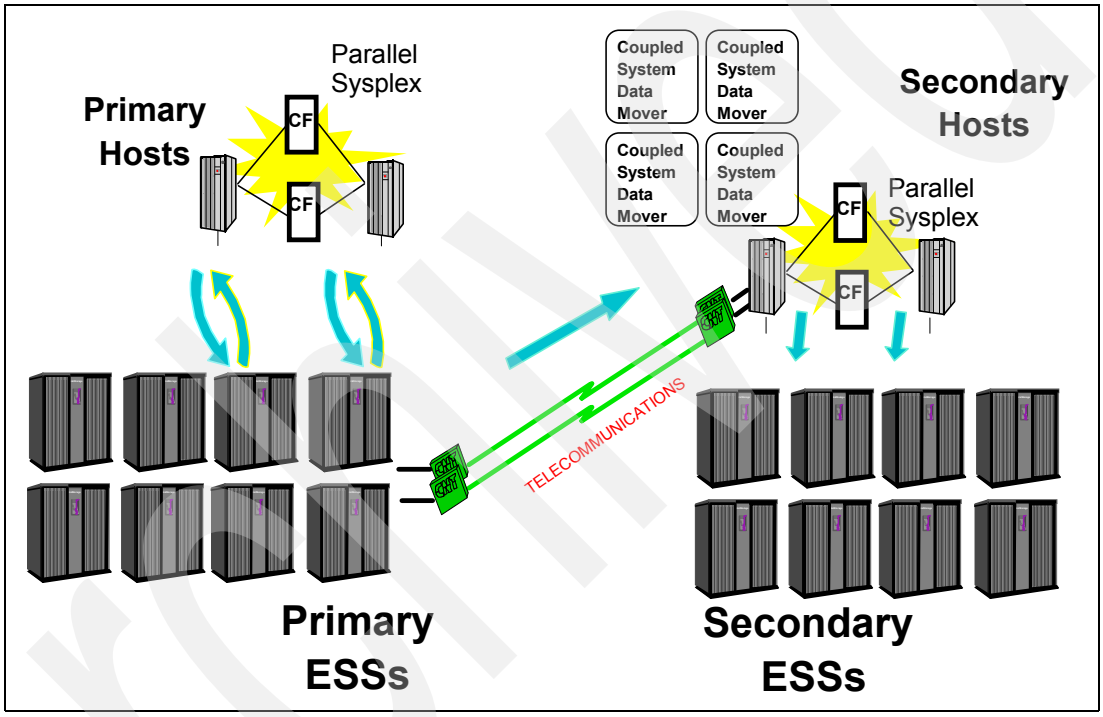


Figure 8-3 Coupled Extended Remote Copy (CXRC)

The secondary site can be located in another continent, since the connectivity between SDMs and primary disks is provided by channel-extenders devices and telecommunication lines.

8.1.2 XRC hardware requirements

XRC hardware requirements are:

- zSeries hardware must be used.
- For the primary ESSs, the XRC optional feature must be enabled. Support is provided by XRC-capable Licensed Internal Code (LIC).
- Primary systems must have a common time reference. XRC uses timestamped record information to ensure that updates are not copied out of sequence at the secondary site:
  - In environments where the primary systems are on different CECs, an IBM Sysplex Timer® is required.

– When you have only one primary system, or the primary systems are running on different LPARs on the same CEC, the system time-of-day clock is used.

- A compatible secondary volume must be available for each primary volume you want to copy. The secondary volume must have the identical track capacity and number of tracks per cylinder as the primary volume. The capacity of the secondary volume can be the same or larger than that of the primary volume.

8.1.3 Dynamic workload balancing

With XRC Version 2, a dynamic workload balancing algorithm was introduced. The objective of this mechanism is to balance the write activity from primary systems, and SDM's capability to offload cache during write peaks or temporary lack of resources to SDM, and with minimal impact to the primary systems.

In situations where the SDM offload rate falls behind the primary systems write activity, data starts to accumulate in cache. This is dynamically detected by the primary ESS microcode, and it responds by slowly but progressively reducing available write bandwidth for the primary systems, thus giving the SDM a chance to catch up.

The ESS implements device level blocking. The update rate for a volume continues unrestricted unless a volume reaches a residual count threshold waiting to be collected by the SDM. Whenever that threshold is exceeded, application updates to the single volume are paused to allow the SDM to read them from the cache of the subsystem.

Important: By using the DONOTBLOCK parameter of the XADDPAIR command, you can request XRC not to block specific devices. This option can be used for IMS WADs, DB2 logs, CICS journals, or spool data sets which use small block sizes, do numerous updates, and are critical to application response time.

8.1.4 Planned outage support

As the ESS provides XRC Version 3 *hardware bit map* support, suspension of an XRC session has no primary system impact since the ESS maintains a *hardware bit map* of updates, thus not consuming cache resources while sessions are suspended.

Suspending an XRC session

This is accomplished by using the XSUSPEND TIMEOUT command. This command is issued when you want to terminate the SDM for a planned activity such as a maintenance update, or moving the SDM to a different site or a different LPAR. The XSUSPEND TIMEOUT command will end the ANTASnnn address space and inform the involved LCUs that the XRC session is suspended. The ESSs will then use the hardware bit map to record changed tracks, and will free the write updates from the cache.

When the XRC session is restarted with the XSTART command and volumes are added back to the session with the XADDPAIR command, the hardware bit map maintained by the ESS while the session was suspended will be used to resynchronize the volumes, and thus full volume resynchronization is avoided.

Suspending volumes

This is accomplished by using the XSUSPEND VOLUME command. The XSUSPEND VOLUME command will accept either a list of up to 100 volumes pairs, or ALL. If ALL is specified, XRC will suspend all volumes in the XRC session at the same consistency time.

If a volume list is specified, then XRC will suspend all the volumes in the list at the same consistency time. If it is necessary to suspend more than 100 pairs, but not all the volumes in the session, then multiple XSUSPEND VOLUME(volser,volser,...) ATTIME() commands with the same ATTIME will cause the volumes to suspend with the same consistency time. Be sure to pick an ATTIME far enough in the future to allow all commands to be processed.

When volumes are suspended with the XSUSPEND VOLUME command, the SDM address space is not terminated. The ANTASnnn address space continues to run and with primary disk subsystems not supporting hardware bit maps the SDM continues to read updates. With the ESS as the primary disk subsystem the hardware bit map is used to track updated tracks. No updates are applied to the journals or secondary volumes.

8.1.5 Unplanned outage support for ESS

The ESS, with its capability of maintaining hardware bit maps, supports *unplanned outages*. This is the capability of avoiding full resynchronization of volumes after an unexpected failure on one or more of the components in the SDM data path.

When data starts to accumulate in cache, either for a temporary write peak from the primary systems, or for an outage in the data moving process of the SDM, the dynamic workload balancing algorithm takes place to reduce the effects for situations, but if this is an outage of a vital SDM component, this mechanism does not help.

When the data in the cache exceeds a predefined level, a time interval starts to decrease. This time interval value can be customized and can be set individually for each LSS.

The time interval is reset every time the SDM reads from the storage control session. If there is no draining of the cache, the data will continue to accumulate, and will eventually reach a predefined high threshold.

A *long busy* condition will then be presented to the primary systems, and I/Os will be queued in the primary hosts. The duration of the *long busy* condition is the remaining part of the time interval.

When the time interval expires, the SC session is suspended. The hardware bit map will be used to reflect the changed tracks, and the cache resources will be freed. The outage can be of any duration, the primary systems continue to run without any impact from the XRC environment, and the ESS keeps track of the write updates in the hardware bit map. This bit map is used at the later time when the problem is fixed and the volume pairs are eventually added back to the XRC session with the XADDPAIR command.

8.2 XRC components

In this section we introduce the main components of the XRC environment.

8.2.1 Primary storage subsystem

The primary storage subsystem is a collection of volumes that is designated to be copied to a secondary site. This collection of volumes may be all the volumes at the primary site, or a subset of them. Attaching the primary volumes are one or more host applications running on one or more z/OS system images.

Attention: to act as a primary storage subsystem, the subsystem needs to be XRC capable. For ESS, XRC capability is an optional licensed microcode feature. Other DASD vendors have licensed the XRC capability from IBM and provide XRC support in their products.

8.2.2 Secondary storage subsystem

The secondary site storage subsystem is a collection of volumes that holds copies of the primary volumes. Each XRC primary volume has a corresponding secondary volume. The XRC secondary volumes can be part of any storage subsystem supported by the secondary system, and by the z/OS system where the SDM is running. These storage volumes are offline to this image under normal operational mode. SDM brings the volumes online again at the end of the XRECOVER process, which recovers all the secondary volumes to the nearest common consistent point-in-time before the emergency situation happened at the primary site.

8.2.3 System Data Mover

The System Data Mover (SDM) is part of the DFSMSdfp software, and must have connectivity to the primary volumes and to the secondary volumes. When primary systems write to the primary volumes, the SDM manages the process of copying those updates to the secondary volumes.

A single SDM can manage between 1500 and 1800 volumes. Multiple XRC sessions can be in effect per z/OS system (up to 5 with or without coupling them). Multiple instances of SDM on separate z/OS images are also possible. Each SDM will have one XRC session that is responsible for a group of volumes. SDM maintains the updates' sequence consistency for the volumes participating in the XRC session, across LSSs in the ESS and across ESSs (as well as with other primary storage subsystems that support XRC).

8.2.4 Address spaces

The system data mover for XRC operates in specific system address spaces whose names start with "ANT". The functions that are provided by these address spaces are described in Table 8-1.

Table 8-1 Address space descriptions

Address space	Description
ANTAS000	This address space handles TSO commands and API requests that control XRC. ANTAS000 and the point-in-time address space, ANTMAIN, are automatically started during IPL. The SDM automatically re initializes the address space if it is canceled.
ANTASnnn refers to: ANTAS001 ANTAS002 ANTAS003 ANTAS004 ANTAS005	An ANTASnnn address space will start for each XSTART command issued

Notes:

1. Because they process a large number of I/O requests, ANTAS000 and ANTASnnn are unswappable. The ANTASnnn address space runs at the MASTER dispatching priority.
2. If you are running CXRC, you may run up to five XRC sessions with address spaces of ANTAS001-ANTAS005 in a single LPAR, with or without coupling them.
3. If the ANTAS000 address space is canceled, XRC automatically restarts it. Except for TSO commands and API requests, there is no impact to SDM functions that run in the ANTASnnn address space. If an event cancels the ANTASnnn address space, XRC ends all active storage control sessions with storage controls that do not support hardware bitmapping.

8.2.5 Journal, control, and state data sets

The *journal data set, control data set,* and *state data set* are used by the SDM to harden — on disk — consistent time groups of updated records received from the primary volumes, and to control the process of applying them to the secondary volumes, thus maintaining sequence consistency. SDM creates Consistency Groups and writes them to the journal data sets.

The control data set has pointers into the journal data sets indicating the last set of Consistency Groups written to the secondary volumes and the amount of data written to the journal.

The state data set maintains a list of XRC primary and secondary volumes, and is updated whenever a volume pair status for that session is changed.

Combining SDMs

A Coupled XRC configuration requires another data set to ensure recoverable consistency among all XRC subsystems handled by the CXRC system. This data set is called the *Master Data Set*. It is written at regular intervals by all SDMs. Its data includes the last consistency point applied to secondary volumes and the last consistency point written to the journal data set.

In the same I/O operation, the SDM reads the status of the last consistency group (both applied and on journal) written by all the other SDMs in the CXRC configuration, and writes its own status.

8.2.6 An XRC session

From a DB2 point of view, an XRC session must include all objects for a DB2 subsystem or a data sharing group, such as these:

- DB2 Catalog
- DB2 directory
- Active Logs
- BSDS
- All ICF catalogs that contain entries for DB2 objects above
- DB2 program libraries
- User data: table spaces and indexes

Any other Transaction Managers such as CICS or IMS that you wish to be consistent with the DB2 environment have be included, as XRC is not DB2-specific.

When an XRC session is started, it is given a session-id and volume pairs can be added to it. A volume pair can only belong to one XRC session, but a primary volume in one session can be the secondary in another, and a secondary volume can be primary in another session.

Control for starting, ending, and recovering XRC operations is at the session level. If multiple sessions are required, multiple instances of the SDM must be started. Depending on how many SDMs are required you may need multiple z/OS images, because each z/OS system supports up to five SDMs. If you are considering running multiple XRC sessions, you should be aware of the fact that to provide data consistency across XRC sessions, you will need to establish a Coupled XRC (CXRC) environment.

8.2.7 Utility devices

For any host system to read data from a disk storage control unit in z/OS, that storage control unit must provide a device address (UCB); this is a requirement of the S/390 I/O architecture. The *utility device* is a physical disk belonging to an LSS that SDM will use to address all the reading operation for that LSS

It is important that the utility device is not heavily used by the primary system, because this could prevent SDM from getting access to the device, and the offload process would slow down.

Tip: For the ESS, you have the option to define a single cylinder custom volume, and then specify this volume in the XADDPAIR command with the XRCUTL parameter.

Utility device selection method

You can specify the method of selection of the utility devices using the XSET command with the UTILITY parameter. You can choose that a specific (*fixed*) device be used, or you allow the ESS to determine which utility device (*floating*) to use.

In a channel-extender configuration, the telecommunication links tend to be the most vulnerable. If they fail, it is essential that you do not leave a production volume in the status of an XRC utility device. If an application attempts to reserve a primary volume that is currently being used as a utility device, the message `IEA482I WAITING FOR CONTROL UNITS` is written to the system log. The application program will not be able to proceed until the storage control session is suspended.

Tip: In a channel-extender configuration, it is highly recommended that you define fixed rather than floating utility devices. Also, you can define utility devices on additional paths to improve resilience and performance.

8.2.8 An XRC storage control session

An XRC storage control session (XRC SC session) is created in the ESS LSS as soon as the first volume pair is established. The XRC SC session is used by SDM to perform read operations from that LSS.

It is possible to have multiple XRC storage control sessions within an LCU per SDM.

SDM will use a separate utility device and run a separate subtask per XRC storage control session, making it possible to transfer data in parallel between one LCU and one SDM. This functionality is also referred to as *multiple reader support*.

If SDM has multiple XRC storage control sessions in one LCU, SDM needs one utility device per storage control session.

You can use the SCSESSION parameter in the XADDPAIR command to route a primary volume to a specific storage control session within an LCU. The primary volume will be part of the default XRC storage control session when no SCSESSION parameter is specified.

XRC session and XRC SC session: Configuration rules

The following configuration rules apply when defining XRC and XRC SC sessions:

- Maximum 64 combined XRC Storage Control sessions and CC sessions for each primary site ESS LCU
- Maximum 80 XRC SC sessions in total (across multiple ESS LCUs) for each SDM

8.3 XRC operation: data flow

Figure 8-4 illustrates a simplified view of the XRC components and the data flow logic. But before discussing the logic, let us understand a key issue for the characteristic consistency of this process: *timestamping*.

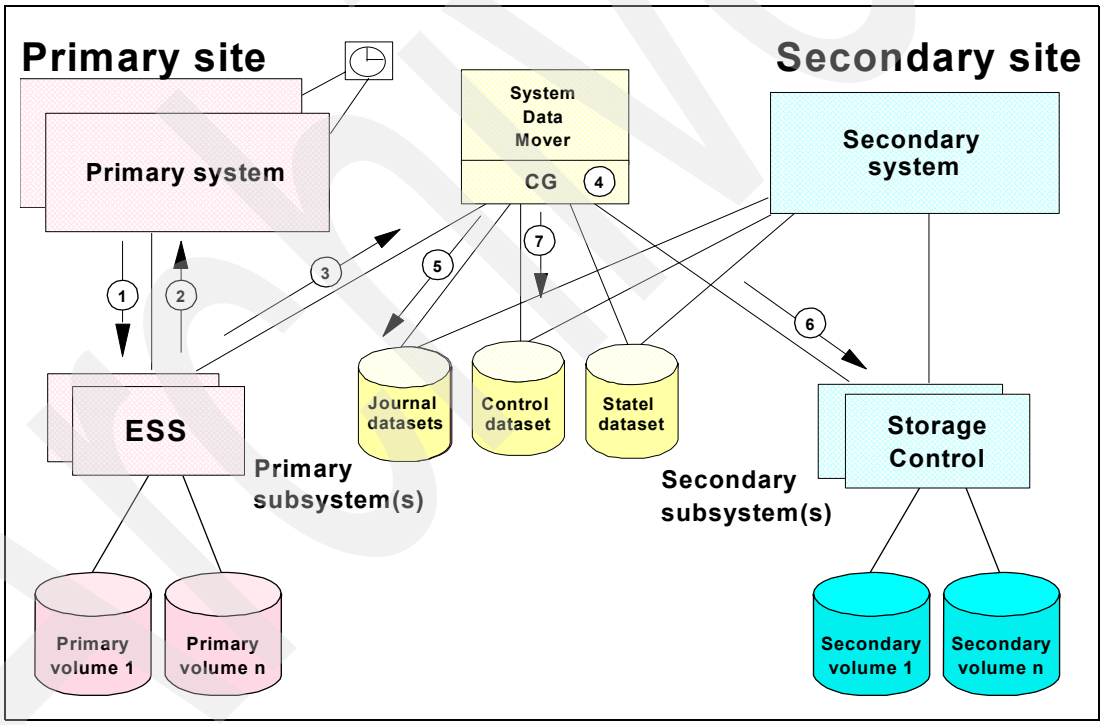


Figure 8-4 XRC data flow

Common time reference

When an XRC pair is established, this is signalled to the primary system, and the host system DFSMSdfp software starts to time stamp all write I/Os to the primary volumes. This is necessary to provide data consistency across multiple LSSs.

If those primary volumes are shared by systems running on different processors, an IBM Sysplex Timer is required to provide a common time reference. If all the primary systems are running in different LPARs on the same processor, the system time-of-day clock can be used. The primary systems use Universal Time Coordinated (UTC) time.

XRC is implemented in a cooperative way between the ESSs on the primary site and the DFSMSdfp host system software component System Data Mover (SDM). The logic for the data flow is as follows (refer to Figure 8-4 on page 122):

1. The DB2 system writes to the primary volumes.
2. DB2 I/O operation is signalled *completed* when the data is written to primary ESS cache and NVS; this is when *channel end* and *device end* are returned to the primary system. Thus, the DB2 write I/O operation has completed, and now the updated data will be mirrored asynchronously, as we describe next.
3. The ESS groups the updates into record sets that are asynchronously off-loaded from the cache to the SDM system. As XRC uses this asynchronous copy technique, there is no performance impact on the DB2 I/O operations at the primary site.
4. The record sets, perhaps from multiple primary storage subsystems, are processed into Consistency Groups (CGs) by the SDM. The CG contains records that have their order of update preserved across multiple LSSs within an ESS, across multiple ESSs and across other storage subsystems participating in the same XRC session. This preservation of order is absolutely vital for DB2's dependent write I/Os. The creation of CGs guarantees that XRC will copy data to the secondary site with update sequence integrity.
5. When a CG is formed, it is written from the SDM real storage buffers to the journal data sets.
6. Immediately after the CG has been hardened on the journal data sets, the records are written to their corresponding secondary volumes. Those records are also written from SDM's real storage buffers. Because of the data in transit between the primary and secondary sites, the currency of the data on secondary volumes lags slightly behind the currency of the data at the primary site. The lag can be less than .5 second and is a function of both bandwidth and write intensity of the workload at any given time.
7. The control data set is updated to reflect the fact that the records in the CG have been written to the secondary volumes.

8.4 Details on Consistency Groups

If you want to prove to yourself that the writes are truly applied in time sequence, you can follow the example we describe in this section.

Timestamping process

Maintaining the update sequence for applications whose data is being copied in real time is a critical requirement for DB2. If data is copied out of sequence, serious integrity exposures could render the recovery procedures useless. XRC uses special algorithms to provide update sequence consistency for all data.

The starting point for maintaining update sequence integrity is when a record is first written by DB2 to a primary volume of an XRC managed pair. When the record is written, the LSS maintains the data (including the time-stamp information) and transfers it to SDM along with other updated records. The timestamping occurs at a very low level in the z/OS operating system IOS modules and applies to all data. Deferred writes held in DB2 buffers are processed in Start Subchannel (SSCH) order. This ensures that the timestamping process delivers update sequence integrity support accurately and efficiently.

XRC update sequence consistency example

The example shown in Figure 8-5 illustrates how the Consistency Groups are created.

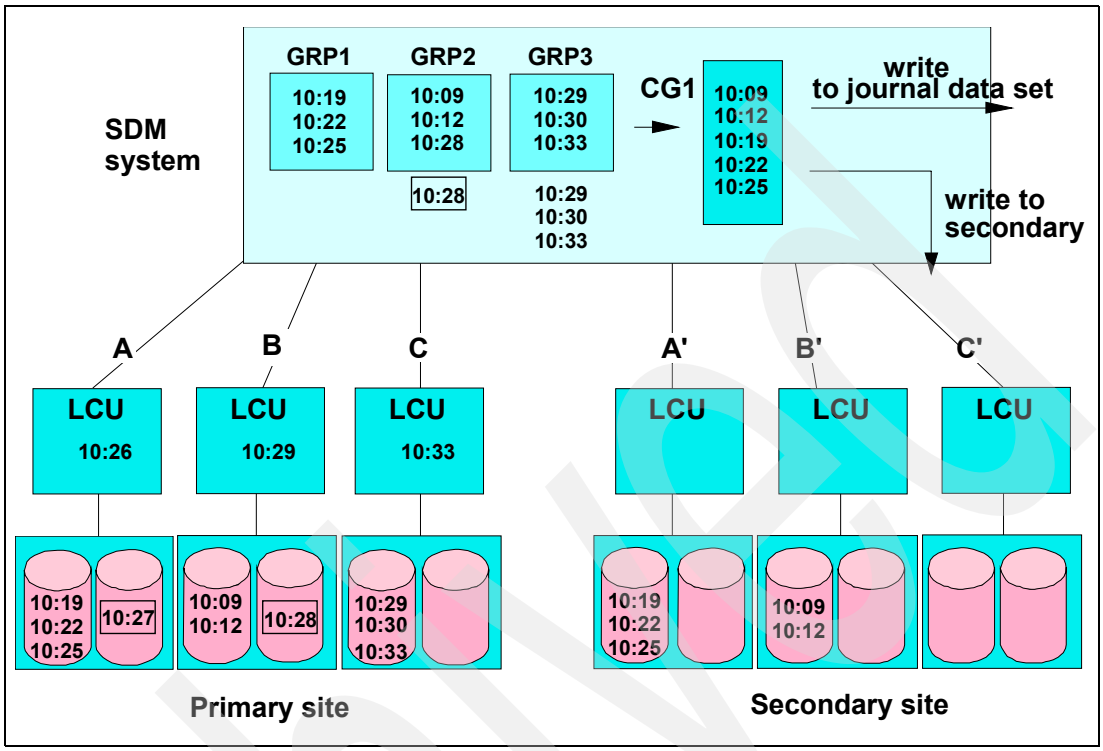


Figure 8-5 Creation of Consistency Group - CG1

In Figure 8-5 we see a configuration with six primary volumes attached to three LCUs at the primary site, and six secondary volumes attached to three LCUs at the secondary site. The SDM system has connectivity to the three LCUs at the primary site and the three LCUs at the secondary site. We assume a one-for-one relationship between the LCUs and the primary volumes at the primary site, and the LCUs and secondary volumes at the secondary site (A, B, C, copied to A', B', C', respectively).

The applications running on the primary system update records on the primary volumes at certain times. In our example, the first volume attached to LCU(A) at the application site has three records updated at 10:19, 10:22, and 10:25. Similarly, the times recorded for the other five volumes are the times when their records are updated (ignore the secondary site disk for the time being).

Two record updates are highlighted on two volumes at the primary; we will use these two records as our dependent writes. The first record is the log update on the second volume attached to LCU(A), which takes place at 10:27. The second record is the database update on the second volume attached to LCU(B), which takes place at 10:28.

Now, we define a point-in-time when SDM reads the updated records from each primary LCU. Notice the time in each of the three LCUs at the primary site. We assume that this is the time that SDM will read the records that have been updated from that LCU.

So, at 10:26, SDM will read updates from LCU(A); at 10:29, from LCU(B); and at 10:33, from LCU(C). We will look at each LCUs individually and describe the record transfer to SDM and the CG processing that takes place when all LCUs have sent their updates to SDM.

- SDM reads from LCU(A) at 10:26.

  At 10:26, SDM will read three records from the LCU(A) — the records written at 10:19, 10:22, and 10:25. SDM will not read the highlighted log record because it has not been written at this point-in-time (the log will be written at 10:27). We are deliberately stress testing SDM by creating a scenario that could easily result in copying data out of sequence. This group of records will be stored in SDM's real storage buffers and is designated as GRP1 on the diagram. SDM will not write this data yet, because it must determine whether the other LCUs have dependent record updates that are included in the session.

- SDM reads from LCU(B) at 10:29.

  At 10:29, SDM will read three records from LCU(B) — the records written at 10:09, 10:12, and 10:28. Note that the highlighted database update (10:28) is included in this group of records, which is described as GRP2 in the diagram. This poses a potential problem because SDM has read the database update, but not the log update. If SDM simply writes data as it receives it, the serious data integrity exposure described before would prevail. SDM's ability to avoid these exposures becomes clear as we proceed with this example.

- SDM reads from LCU(C) at 10:33.

  At 10:33, SDM will read three records from LCU(C) — the records written at 10:29, 10:30, and 10:33. This group of updated records is called GRP3. At this stage, SDM has received responses from all three LCUs that have volumes in the XRC session. The next phase of providing the remote copy function is to perform the update sequence integrity check for all of the updated records read from all of the primary storage subsystems in the XRC session.

- SDM creates CG.

  SDM now has three groups of records that it uses to create a CG. The CG is the unit of data transfer to the journal data sets at the secondary site. It contains records that SDM has determined can be safely written to the secondary site without risk of out-of-sequence updates. To compile the CG, SDM uses all of the record groups read from the LCUs as input. In our case, we have three record groups that SDM uses as input to produce a single CG output.

  SDM compares the maximum timestamp for each individual record group (GRP1 = 10:25, GRP2 = 10:28, GRP3 = 10:33) and selects the smallest value (10:25) to calculate which records should be included in the CG. SDM calculates that all records written at or before this smallest value time can be written together to the remote site journals as a CG.

  In our example, five records (10:09, 10:12, 10:19, 10:22, and 10:25) qualify for inclusion in the first Consistency Group of CG1. Using this calculation, SDM ensures that dependent write records are not written *ahead* of time as demonstrated by the fact that the database update at 10:28 has not been included in the CG. This is vital because the log update (10:27) is located behind a different LCU(A) and has not yet been read into SDM's real storage buffers for inclusion in the *Consistency Group creation algorithm*.

  Now, we see the importance of using the maximum timestamp in each individual record group, and selecting the minimum of all of the maximums across record groups as the CG *upper limit* timestamp.

  SDM has *retained* the database update along with other possibly dependent write sensitive records in its real storage, until it calculates that it is safe to include them in a future CG. In the diagram, these four records are listed beneath their respective original record groups (10:28 in GRP2 and 10:29, 10:30, and 10:33 in GRP3).

- SDM writes CG to journal data sets.

  Having created a group of records that can be safely written together, SDM then writes this group to the journal data sets at the secondary site. Thus SDM can harden the updates as quickly as possible in one I/O to a single data set.

- SDM writes the updates to the secondary volumes.

  After the CG has been written to the journal data set, SDM immediately writes out the individual records to their appropriate secondary volumes. This transfer takes place from SDMs real storage buffers (the journal data sets are only read during recovery using the XRECOVER command). Updating the secondary volumes could involve several I/Os, because the record updates could be directed to several volumes located behind several LCUs. In this example, the CG comprises five records directed to two volumes attached to two LCUs. The figure illustrates that the five records have been successfully copied to their corresponding secondary volumes — 10:19, 10:22, and 10:25 written to the first volume of LCU(A'), and 10:09 and 10:12 written to the first volume of LCU(B').

Figure 8-6 concludes the example on update sequence consistency by stepping forward in time and describing how the dependent write updates are copied to the secondary site.

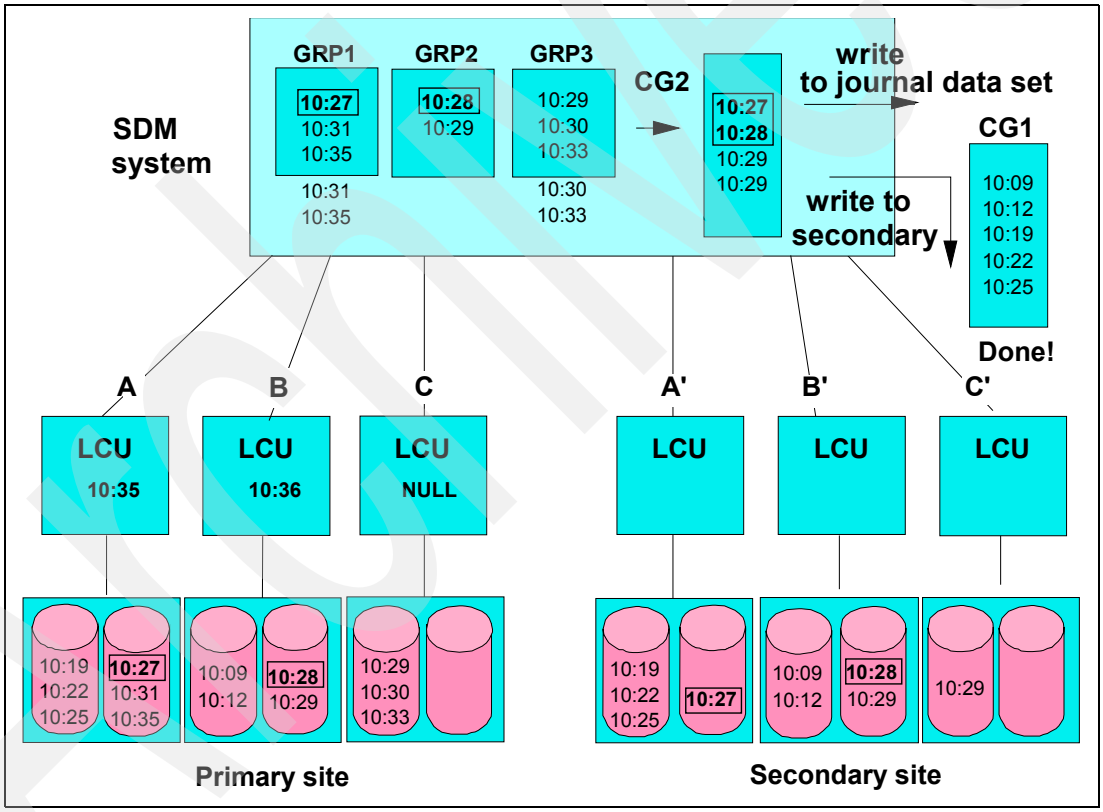


Figure 8-6 Creation of Consistency Group - CG2

SDM copies data in real time, so it constantly transfers updated records from perhaps multiple LCUs in an XRC session. In our example, we simulate moving the *clock forward* in time and tracing the steps taken by SDM in the production of its second CG.

Notice that LCU(A) and LCU(B) at the primary site now have different times indicating when SDM reads the updated records. We proceed as before:

- SDM reads from LCU(A) at 10:35.

  At 10:35, SDM reads three records from LCU(A) (10:27, 10:31, and 10:35). Note that the log update (10:27) has now been read. This group of three records has been designated GRP1 in the diagram.

- SDM reads from LCU(B) at 10:36.

  At 10:36, SDM reads one record from LCU B (10:29). This record can now keep the database update record (10:28) company in SDM real storage (these two records are in GRP2 in the diagram). Remember that the database update record (10:28) was not included in the previous CG and was kept in real storage by SDM along with other *residual records*.

- SDM detects NULL response from LCU(C).

  Finally, SDM completes its cycle of updated records collection by establishing that the third LCU(C), which also connects volumes in this XRC session, has *nothing to report*. The NULL response must be sent even if the LCU has no updates, because it indicates that the LCU has not suffered an outage, thus the record groups retrieved to date from other LCUs can be used to build a valid CG.

- SDM creates CG.

  SDM creates the second CG using the same algorithm as before. We can see that the two dependent writes (the log update at 10:27 and the database update at 10:28) have been *captured* in this CG.

  Notice that the record group transferred from LCU(B) produced a maximum timestamp, which also turned out to be the smallest of the three GRP maximums. Notice also that this value (10:29) enabled all records written before or at this time to be included in the CG — thus allowing the log update and the database update to accompany each other on the journey to the secondary site when written to the journal data sets.

  The diagram shows that when SDM writes the records to the secondary volumes, the relationship between the contents of the log and the database, which is vital for recovery, has been protected. The record residue (10:31 and 10:35 from GRP1, and 10:30 and 10:33 from GRP3) will be processed during the next SDM record group collection cycle.

8.5 XRC considerations for DB2 disaster recovery

In this section we discuss some things to keep in mind when considering XRC for disaster recovery.

The ERRORLEVEL parameter

As part of the planning process, you should evaluate your data consistency requirements in the event of a failure. To specify how XRC should process certain error conditions, use the ERRORLEVEL SESSION subparameter to suspend all volume pairs in the session. You must correct the error and resynchronize all the suspended pairs.

If a volume with error level SESSION encounters an error, all volumes in the XRC session will be suspended. If a failure occurs on a single LSS, and that LSS attaches to any duplex volume pair with error level SESSION specified, then all volume pairs in the XRC session are put into the suspended state. XRC also suspends volume pairs that are in pending state, and the pairs remain where they are in the synchronization process.

Suspending an XRC session

The SDM supports two types of suspension. One is an XSUSPEND VOLUME(ALL) suspension. This means all volumes in the session will be suspended at the same consistency time. With this, the SDM address space is not terminated. With primary disk subsystems not supporting hardware bit maps, the SDM continues to read updates. With the ESS as the primary disk subsystem the hardware bit map is used to track updated tracks. No updates are applied to the journals or secondary volumes.

The second type of suspension is a *session suspension*. This occurs when an error indication is returned to the SDM.

An interesting option is XSUSPEND VOLUME(ALL) ATTIME(). This option would not be used for disaster recovery, but could be used to regularly suspend all the DB2 volumes in a session at a certain time of day, such as 10:30pm (end of after hour trading negotiation). That would allow a FlashCopy to be performed at a business-oriented time for all the DB2 volumes and the critical XRC data sets necessary to recover the entire session for normal business processes.

Note: You can suspend an XRC session using VOLUME(ALL) or by session. If the suspension is by session, the SDM is also terminated. If it is by volume, the SDM remains active.

Single command for recovery (XRECOVER)

One of the most valuable features of XRC is its ability to do a single command recovery. After a disaster that effects the primary site, the secondary site must be made ready for takeover. This involves getting the secondary volumes to a state where their contents are usable in anticipation of application restart. XRC has a single command that performs this task, the XRECOVER command.

The TSO XRECOVER commits the last available Consistency Groups to the secondary volumes, reports the consistency time for the secondary volumes, and then clips the volume serial number of each secondary device to that of its matching primary device.

Part 3

General solutions for disaster recovery

In Part 2, “Disaster recovery major components” on page 41, we have seen the most important building blocks for DR solutions. In Part 3, we now consider how the previously described components can be combined in more general and better performing solutions.

This part of the book contains the following chapters:

- Chapter 9, “Split Mirror” on page 131
- Chapter 10, “FlashCopy Consistency Group” on page 139
- Chapter 11, “Global Copy PPRC-XD” on page 157
- Chapter 13, “XRC: Global Mirror for z/OS” on page 201
- Chapter 12, “Global Mirror PPRC” on page 165
- Chapter 14, “Geographically Dispersed Parallel Sysplex” on page 233

9

Split Mirror

In this chapter we describe the *Split Mirror* solution for local recovery and explain how aspects of it can be used for disaster recovery of SAP systems. The high availability Split Mirror backup/recovery solution delivers a *serverless* backup/recovery using advanced ESS functions with no impact to the SAP R/3 production system. This zero downtime for the R/3 production server means that SAP users continue to work while the backup takes place. Furthermore, the almost instantaneous availability of consistent copies of the whole database provides the ability to recover rapidly in the event of a disaster.

This backup/recovery solution uses ESS subsystem and its advanced copy functions PPRC and FlashCopy. The solution also uses the Parallel Access Volumes function of the ESS, which simplifies storage management while delivering high parallel-query throughput.

The solution described in this chapter can be of interest for several integrated software solutions like ERP applications, and warehousing environments.

This chapter covers the following topics:

- Split Mirror:
  - Normal operation (current logs/back-leveled data)
  - Suspend PPRC links to prepare for next step
  - Back up (FlashCopy) to preserve environment (current logs/back-leveled data)
  - Re-establish PPRC Links to both logs and data
  - Suspend log when 99% of tracks are synchronized
  - Suspend PPRC pairs. Consistency group obtained
  - Resume Logging
  - FlashCopy the consistency group
  - Re-establish PPRC links to Logs
- Considerations

A white paper, authored jointly by SAP and IBM, documents the implementation details. This paper, and others, are available in SAP's SAPNET, and at the following site:

http://www-106.ibm.com/developerworks/db2/library/techarticle/

9.1 Split Mirror

The Split Mirror, as it has come to be known, was developed by and for SAP/R3 users. Backup and recovery requirements for an SAP R/3 DB2 for z/OS installation are somewhat different from traditional software packages in that the SAP R/3 application programs reside in the SAP R/3 database, and changes to DDL and DML are not controlled. The program modules that are resident are backed up whenever the SAP R/3 database is copied, and DB2 catalog and directory need to be recovered as well.

To provide for effective recovery in the event of failure, any restoration must guarantee that every component in the SAP R/3 environment is logically consistent with the others in terms of the content and structure at a given point-in-time.

While the data was mirrored, that mirroring did not protect against application logical errors. A secondary copy of DASD would carry the same errors as the primary. This solution was implemented not only to provide backup suitable for disaster recovery, but to allow for forward processing of logs against data that did not have the application contamination present.

At a secondary site, the data is mirrored infrequently, but the logs are mirrored continuously. If an application error situation occurs since the last time the data was fully replicated, then the second site can be used to recover the data, applying the log only to the point prior to the data error. It is known as Split Mirror because the mirror is split most of the time. Occasionally (daily, four times daily) the logs and data are synchronized, and then the mirror is split until the next time.

The times of synchronization present the opportunity to capture a point of consistency for DB2 disaster recovery in the simplest form. We will step through the various stages of the scenario.

9.1.1 Normal operation

The technology used includes: PPRC-SYNC (Metro Mirror) and FlashCopy.

Figure 9-1 represents the normal mode of operation: The DB2 active logs are mirrored, but the data is not so current.

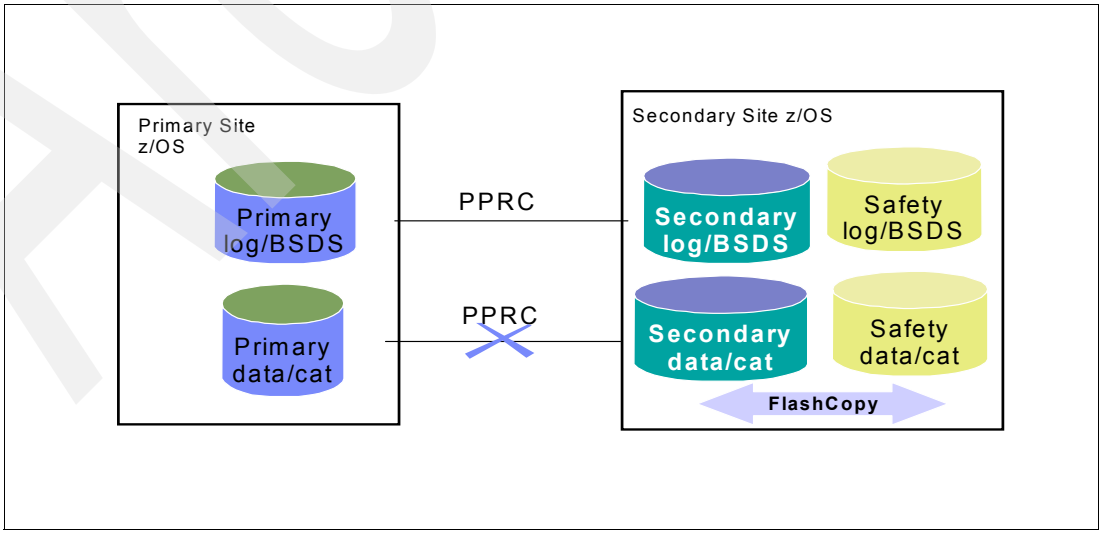


Figure 9-1 Split Mirror - Logs Current - Data not current

The primary logs/BSDS volumes include an ICF catalog on which they are cataloged. Any DASD archives to be used at the secondary site are include in the “logs” group of DASD. The Data volumes include the DB2 catalog and Directory, the application data, as well as any ICF catalogs in which they are cataloged.

The secondary site has duplicate volumes for those on the primary site. In addition they have a third set of volumes that are called the *safety set* as they are used to preserve the environment as changes are made to the secondary DASD. At any given time they guarantee a recoverable set of data.

Initially the volume serial of each volume is the same for each set respectively. As long as the volumes of the secondary set are offline to the secondary z/OS image, the volume serial numbers do not matter. The safety set is labeled with the volume serials of the primary set and are online to the secondary z/OS image.

9.1.2 Suspend PPRC links

In Figure 9-2 we show the suspension of the PPRC links.

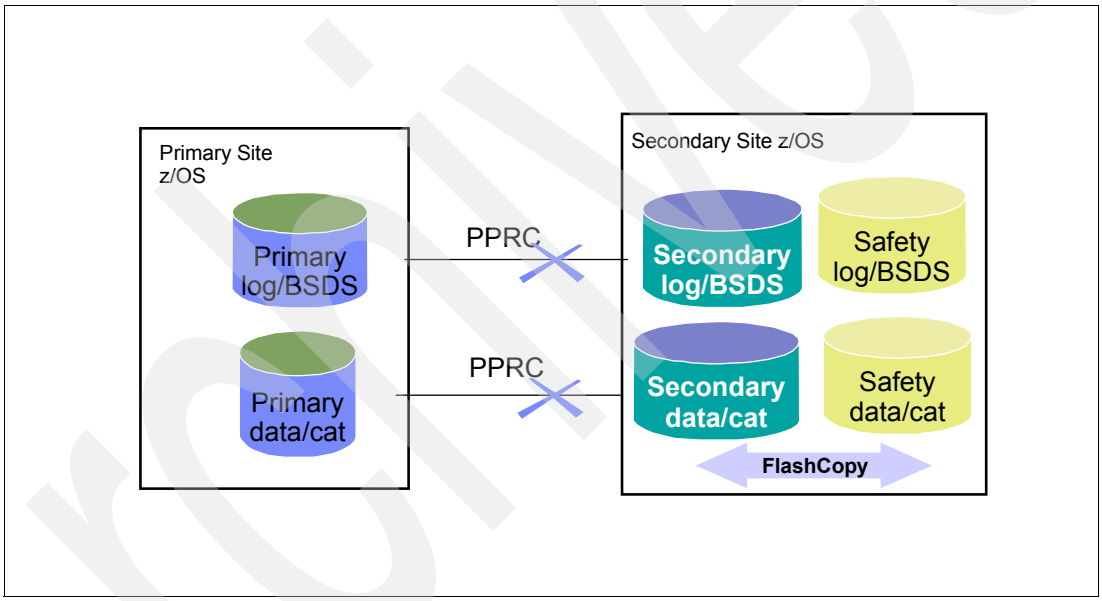


Figure 9-2 Suspend PPRC connections between the logs

Now there is no connection at all between the sites. The primary site continues to operate normally. We want to FlashCopy from the secondary set to the safety set to preserve the current status — current logs, but less than current data.

Since we cannot do any read or write I/O to a secondary volume of a PPRC pair, we cannot use them as source or target volumes for FlashCopy. However, we can bring the secondary volumes to a simplex state and then FlashCopy their contents.

The first activity is to use the PPRC CSUSPEND command to suspend the PPRC pair as we show in this step.

9.1.3 Perform backup to preserve the environment

Figure 9-3 shows that FlashCopy is being used to dump the current volumes and preserve the environment.

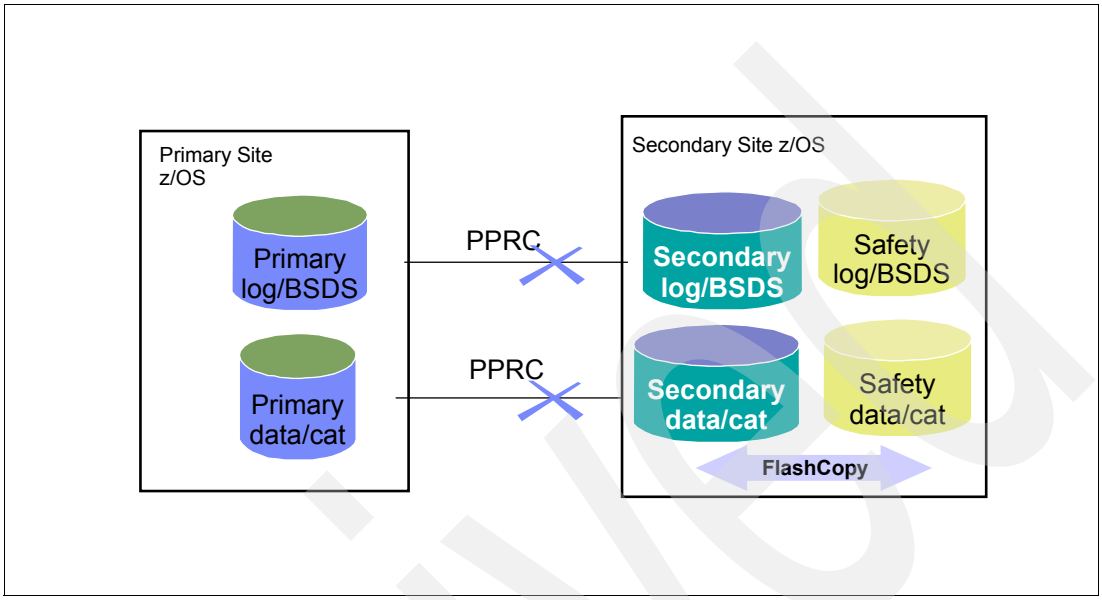


Figure 9-3 Changing the environment

In order to FlashCopy the secondary volumes to the safety set, we must bring the secondary set to simplex state and then change volume serials. Since the all three sets have the same volume serial, we use the PPRC CRECOVER command to bring the secondaries to simplex and change their volume serials. This is done on a volume by volume basis. If the safety volume is VOLA, then the secondary volume will be relabeled VOLB, for example. PPRC mirroring will still be allowed, as the volume serial does not matter for that function. This relabeling need only be done one time.

To clarify at this point, please note that the primary volume serial is VOLA, the secondary is VOLB, and the safety is VOLA respectively for the set.

We then vary the secondary volumes on line to the secondary site z/OS. We FlashCopy from the secondary to safety volumes. We then vary the secondary volumes offline again while the background FlashCopy is running.

Since the secondary PPRC has the same volser as the primary PPRC, it has to be online on an LPAR different from the one where the primary is connected.

9.1.4 Re-establish the PPRC links

Figure 9-4 illustrates the resynchronization of both sets of volumes: data and logs.

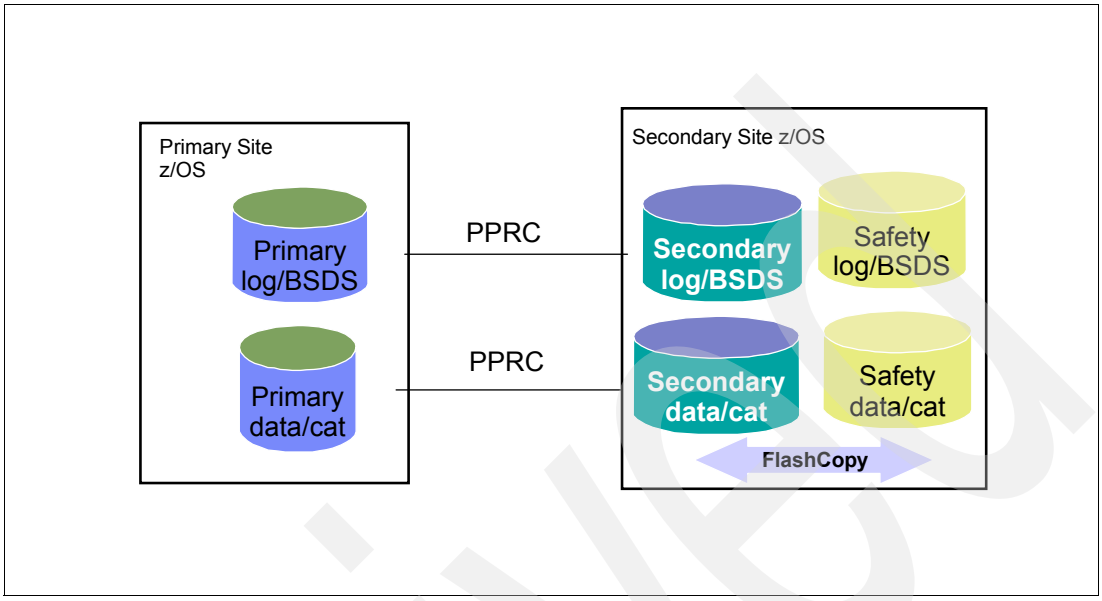


Figure 9-4 Split MIrror resynchronize all volumes

We reestablish the PPRC pairs for both sets of primary and secondary volumes. All tracks will be copied to the secondaries since the last RESYNC. The PPRC CQUERY command is used to track the progress of the RESYNC.

Figure 9-5 illustrates the activities when RESYNC is almost complete.

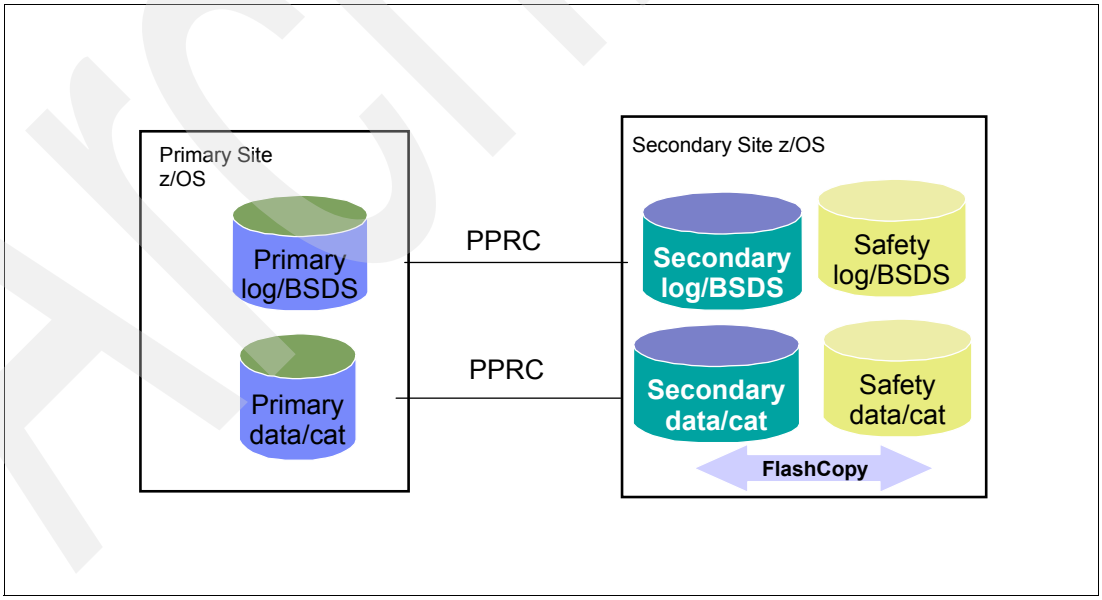


Figure 9-5 Split Mirror - 99% of tracks RESYNC and SET LOG SUSPEND

9.1.5 Suspend pairs

When 99% of the tracks have been synchronized, issue -SET LOG SUSPEND to suspend logging and let the last few tracks RESYNC.

The status of the synchronization is checked with the command CQUERY, or, from the Command Line Interface, with the script *rsQueryComplete*.

The log suspension enables a controlled suspension or “freeze” of the R/3 DB2 update activity instead of having to stop the entire DB2 system while copies are being made. This capability enables the DB2 subsystem to recover to a point-in-time without having to experience an extended recovery outage or without having to stop or quiesce the primary system. The updates are held up for a few seconds while the FlashCopy function logically completes the copy which can then be used for remote copy for disaster recovery, or data migration for non-OLTP accesses to the database.

Update activity will remain suspended until a -SET LOG RESUME command or a -STOP DB2 command is issued.

When all tracks are synchronized, we have our point of consistency.

Figure 9-6 shows the suspension of pairs to capture the *consistency group* just obtained.

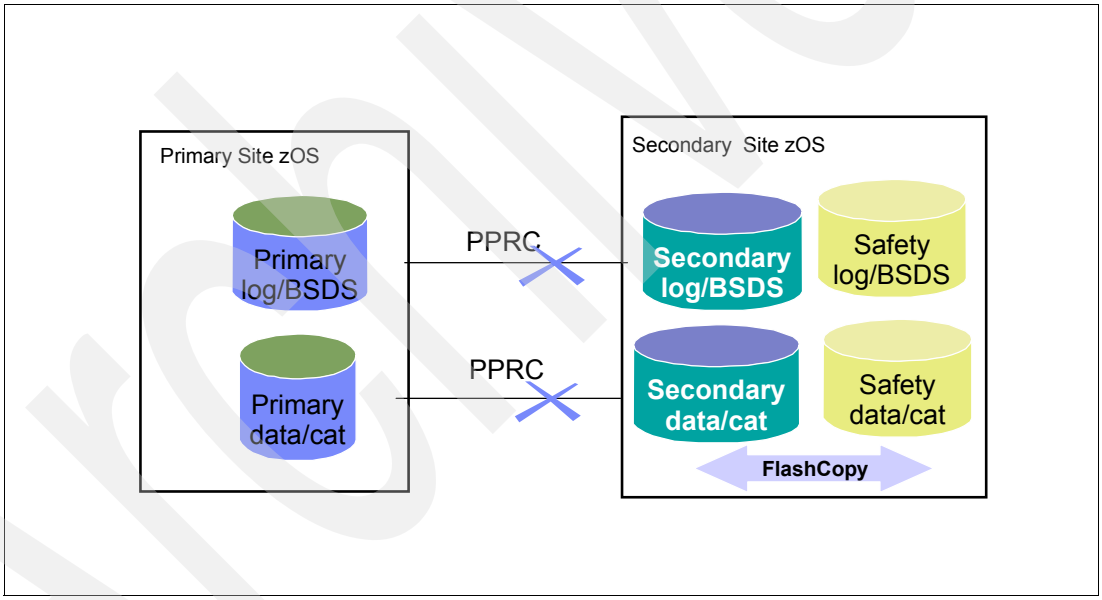


Figure 9-6 Split Mirror - Suspend all PPRC pairs and Resume DB2 logging

We split both mirrors and suspend all PPRC pairs. The secondary volumes are now consistent as the mirror is split.

9.1.6 Resume logging

Now we can issue -SET LOG RESUME and the primary site DB2 begins writing again to the primary DASD.

The amount of time that DB2 has been suspended has been around 10 seconds. Contrast the outage to that of the scenario where FlashCopy is invoked. In that case, a full volume copy must be made. In this scenario only a few tracks are sent to the secondary site.

The logs/BSDS volumes are current with the data/cat volumes. To capture the consistency group just created, we must FlashCopy the Secondary set to the Safety set for both types of volumes. Vary the secondary set online to the Secondary Site z/OS and begin the FlashCopy. We do not need to change volume serial numbers of the secondary set, as they still remain as they were when we relabeled them earlier in the scenario.

DB2 has been actively writing to the primary DASD ever since we resumed logging and the updated tracks are being held for an incremental transmission when the PPRC pairs are re-established. We need the Secondary log/BSDS as fast as possible to perform PPRC RESYNC from Primary logs/BSDS set.

As soon as the FlashCopy relationship is established (Logical Complete), we vary the secondary set of volumes offline. The source of the FlashCopy is now available for read/write access even thought the Safety copies are not complete.

9.1.7 Dump to safety volumes

We can start dumping the safety volumes to tape for potential disaster recovery use.

Figure 9-7 shows that the environment has been returned to normal Split Mirror operation: logs mirrored; data not mirrored. The operational configuration has returned.

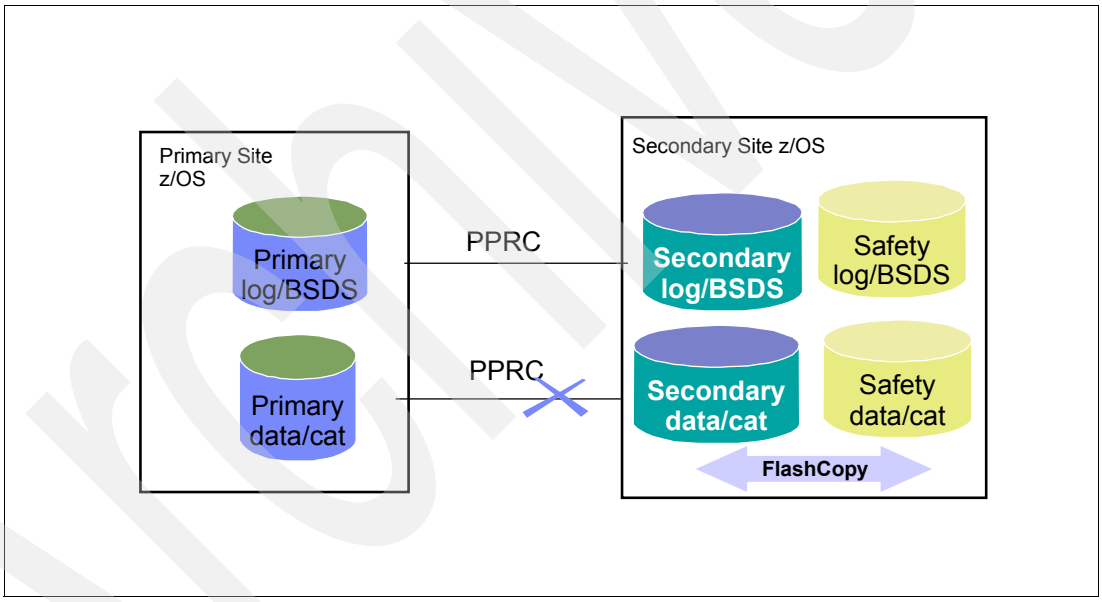


Figure 9-7 Normal operational configuration returned

The PPRC relationship between primary and secondary can be reestablished and the Primary tracks re-synchronized to the secondary while the physical copy is going on from secondary to safety in the ESS hardware.

When FlashCopy relationship ends, safety volumes are all consistent with each other as of the time when the mirror was split.

Tape dumps of safety copies can be used for a normal, fast restart, similar to FlashCopy Recovery. Data is current as of the Split Mirror. All data since then is lost.

The safety volumes can be used for disaster recovery.

Data on the Secondary volumes is less current than the logs, so RESTORE SYSTEM LOGONLY must be used. The log apply phase starts at the point of the latest -SET LOG SUSPEND, as that log point was written to DBD01 as the recovery base log point (RBLP).

The logs must be truncated to a common RBA or LRSN. That point can be identified by using Print Log Map.

9.2 Considerations

In this section we list the advantages, disadvantages, and issues you need to keep in mind.

Advantages

- Easy to implement for DB2 staff (storage staff establish)
- No DB2 complex DR recovery procedures (restore DF/DSS dumps)
- RTO within a couple of hours
- Normal restart of DB2
- No infrequent and/or incremental image copy application issues at recovery site
- Minimal outage for Log Suspend/Resume (10-30 seconds)
- Application errors do not propagate to second copy of data

Disadvantages

- Need duplicate DASD for PPRC/ tertiary copy of data for FlashCopy at second site
- Primary use for DB2 system point-in-time recovery
- Need sufficient tape drive to dump secondaries for transport if used for disaster recovery
- RPO is the amount of time between each RESYNC.

Issues

- Applying logs means RPO is current, while elongating RTO (more log to pass).
- The RESTORE SYSTEM LOGONLY utility makes this a one-step process.
- RESTORE SYSTEM LOGONLY means SET LOG SUSPEND must have been issued to properly set the recovery base log point (RBLP).

10

FlashCopy Consistency Group

This chapter describes how to use ESS FlashCopy Version 2 to provide a Disaster Recovery solution by using the Consistency Group function.

Basically, this solution is a *traditional dump-restore scenario* where FlashCopy V2 is used on a production site to create a self-consistent set of disk information to be used at a remote site (either it is on-property or on-service). This backup information can be used to restore the whole production environment should the primary site became unavailable, for whatever reason.

For further information on planning, implementing, and using FlashCopy, refer to the following publications to complement the information presented in this chapter:

- *IBM TotalStorage Solutions for Disaster Recovery*, SG24-6547
- *z/OS DFSMS Advanced Copy Services*, SC35-0428
- *z/OS DFSMSdss Storage Administration Guide*, SC35-0423
- *IBM TotalStorage Enterprise Storage Server Web Interface User's Guide*, SC26-7448
- *IBM TotalStorage Enterprise Storage Server User's Guide*, SC26-7445
- *IBM TotalStorage Enterprise Storage Server Model 800 DFSMSdss FlashCopy Version 2 Performance Evaluation and Analysis* - 11/11/2003 - Xavier Velasquez, white paper

Also refer to the following Web sites:

- IBM TotalStorage Business Continuance Solutions:

  http://www-1.ibm.com/servers/storage/support/solutions/bc.html

- IBM Global Services - Business continuity and recovery:

  http://www-1.ibm.com/services/us/index.wss/it/bcrs/a1000411

10.1 Solution overview

With FlashCopy V2 it is now possible to gather a *unique* consistency point for all the disk data in a typical IT center, without the need to stop or terminate any running application or subsystem. This *unique* consistency point can then be saved on tapes and transferred off-site to be used in case of any emergency.

Without stopping any running process, it is now possible to obtain multiple copies of the production environment, thus reducing the RPO of a traditional disaster recovery solution.

For example, if it would be possible to take a copy of the production disk data four times a day. Then, in case of an emergency, it would be possible to restart the IT operations from the alternate site with a maximum RPO of 12 hours, supposing that the emergency on the primary site happened just before the execution of the off-site new backup version tapes.

This solution works for DB2 and non-DB2 data as well, thus it is possible to cover the entire z/OS production environment using a single solution. Other platforms can be included in the scenario as well, although requirements and techniques to generate a complete set of backup tapes are platform dependent and might impact RTO and RPO of the whole recovery solution.

Attention: The FlashCopy Consistency Group function can only be activated through the ESS Copy Services Web Interface, or defined through the ESS Copy Services Web Interface and executed/controlled from the Command Line Interface (CLI).

10.1.1 How it works

Consistency groups are an implementation of technology which assists with the consistency of application data capable of dependent writes. If any volume within a consistency group is unable to complete a write to its counterpart in the PPRC relationship, an Extended Long Busy (ELB) will be issued, preventing further writes to any of the volumes within the consistency group. (This ELB period is the perfect time to issue a freeze to all volumes involved to maintain consistency.)

To generate the unique consistency point across multiple ESSs, FlashCopy V2 raises the ELB condition on each primary disk involved in the consistency group creation, which holds all the I/O operations directed to the primary disk until the ELB condition is released. This condition holds also all the dependent I/O operations on other disks not already processed by the FlashCopy function, thus only non-dependent I/O can be executed by other applications on other disks. This ensures that all dependent I/O is not executed, and generates a Global Consistency Point for all the disks in the FlashCopy task.

With all the I/O held (suspended), FlashCopy V2 initiates the source-target relationships and, if requested by the task parameters, it starts the background copy process.

ELB condition is released either by a specific ESS Web Interface task or when the ELB time-out window expires (the default is 2 minutes, but it can be customized from 1 to 600.000 seconds — which means up to +166 hours). To release the ELB condition before the time-out window expires, you can execute a task (*FlashCopy Consistency Group Created*) through the ESS Web Interface to indicate that all the FlashCopy source-target relationships have been established (*Logical Completion*).

As soon as the ELB condition is released, the I/O operations restart the normal processing, and the workload in the system proceeds as normal. Meanwhile, in the ESSs, the FlashCopy background copy operations are executed concurrently with the I/O workload from the host systems.

How long is the production workload suspended?

Improvements in FlashCopy V2 are designed to help reduce the establish time for volume level FlashCopy operations. These establish time improvements are critical, as all host application I/O must cease while the FlashCopy relationship is being created to obtain a consistent point-in-time copy. As a result, an application only needs to be down for a small amount of time to be completely backed up. In addition, this reduction may shorten overall backup window times, as DFSMSdss tasks can complete faster than before.

In particular, DFSMSdss FlashCopy V2 establish time on the ESS 800 performed up to 74% better than FlashCopy V1 on the ESS 800 and up to 83% better than on the ESS F20 in tests.

Important: For optimal performance, source and target pairs should not be established on the same rank within a single LSS, as establish time and background copy rate performance may suffer.

Rank is a unit of 8 physical disks connected to the same SSA channels.

Figure 10-1 shows measured performance of FlashCopy V2 establish pair time.

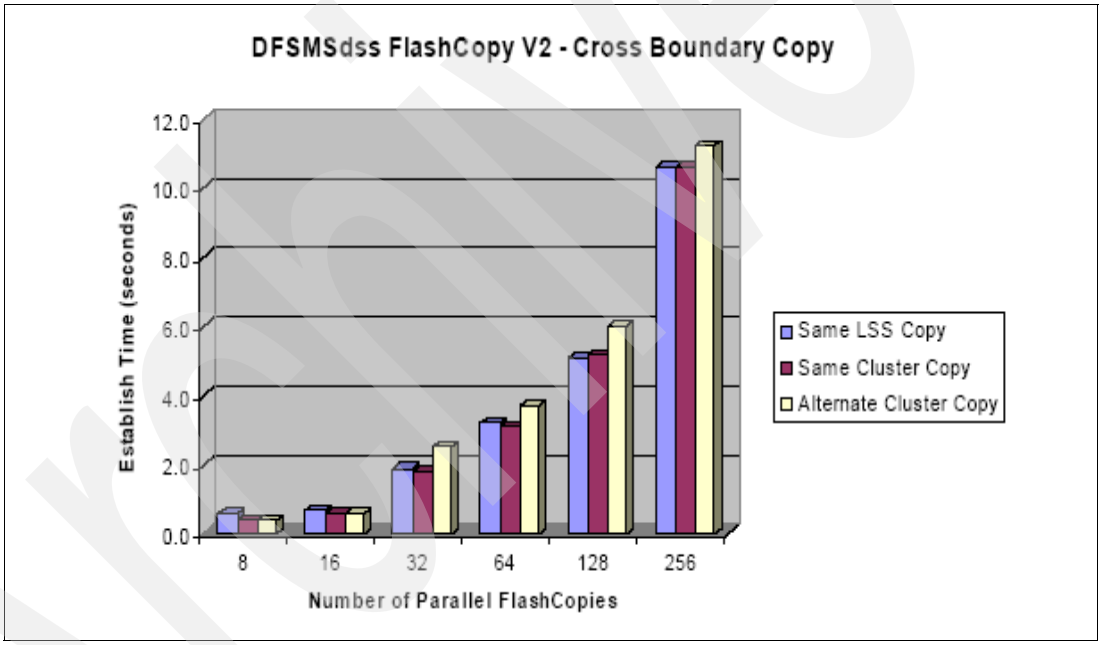


Figure 10-1 FlashCopy V2 establish pairs performance test

Based on these measurements, we can estimate the time required to establish FlashCopy relationships:

Assuming that our DASD environment is on an ESS 800 storage subsystem with 16 ranks of 36 GB 15K rpm disks, this means we have 3.4 TB of space in the ESS, which means about 1230 3390-3 volumes.

Now since FlashCopy requires the primary and secondary to be of the same geometry and to have a target disk available (in terms of unit and space) per each primary disks (source volume), it means that in our ESS, half of the available space has to be dedicated to FlashCopy target.

This brings our ESS to have about 1.7 TB of space available as primary disk (FlashCopy source) and 1.7 TB available as FlashCopy Target space, which means about 617 3390-3 volumes as primary and 617 3390-3 volumes as FlashCopy target.

According to the performance measure, the time required to establish the FlashCopy relationships for all those 617 3390-3 volumes is about 30 seconds.

In an environment where z/OS production environment is larger than 1,7 TB, all required volumes are hosted on multiple ESSs, and FlashCopy operations can be executed in parallel on all those ESSs. This means that, while still maintaining the 256 concurrent relationships per ESS, the time required to establish all the FlashCopy relationships in such an environment is still about 30 seconds.

10.2 Design the solution

To design the solution, we basically have to define its components and how they relate to each other. The following are the essential macro components of a traditional dump-restore solution that must be defined:

1. Backup environment requirements
2. Recovery environment requirements
3. Testing environment requirements
4. Backup procedure
5. Off-site procedure
6. Restore procedure
7. Automation requirements

The information provided in the following paragraphs is not exhaustive, but it provides guidelines on the whole design phase for a traditional dump-restore disaster recovery procedure.

10.2.1 Define backup environment requirements

A Consistency Group is a group of volumes participating in FlashCopy relationships that need to keep the consistency of the logically related data that spans across them. In a production environment, a Consistency Group can be the entire z/OS production disk volumes, or self-contained subsets of it that can be restored independently one from the other.

The definition of Consistency Groups is an activity that is normally executed by the Disaster Recovery Administrator. This person interfaces with the different user-groups among an organization (z/OS administrator, database administrator, application developers, storage administrator, etc.) to define the isolation of processes (and their data) and tries to determine which data needs to be saved and which does not.

Sometimes it is more effective to approach the problem from the opposite point, that is, instead of defining which data needs to be saved, let's try to define what to exclude.

Now let's start thinking at the procedure we are going to implement: it is a traditional backup and recovery, which means that we would have a couple of consistency points per day. Having this in mind, it is clear that we do not need to save any *transient data*, since in case of a restore at the alternate site, all the processes will be restarted from the nearest consistency point, and the workload since then will be re-executed from the new site.

A second point is that the process will be volume based, that is it is not necessary to get inside each volume at data set level to determine what is needed.

Now, if data set naming conventions and DFSMS rules in your environment do not strictly separates data, the Consistency Group will include all disks in your z/OS environment; while if naming conventions and DFSMS rules are applied and followed, the Consistency Group can be restricted to z/OS production environment disks only.

Whatever the case, as a subset of the above, our Consistency Group could contain just the DB2 production environment data only. The following is a non-exhaustive list of what has to be included in the DB2 Consistency Group:

- Databases
- Catalog and Directory
- BSDS
- Logs and Archive Logs on disk
- Image Copies on disk
- System Load Libraries
- Application Load Libraries (and source libraries possibly)
- ICF catalogs where all the previous data is cataloged
- Any other data needed by DB2 production environment, for example: sequential and VSAM data set needed by batch applications

Attention: If you decide to follow this last option, remember that all data within this consistency group will be saved and restored as a single entity, thus it is your responsibility to ensure that a DB2 recovery procedure is “in sync” with the recovery procedure for the other components of the z/OS system that are required to restart DB2 at the recovery site.

For the backup phase, we need at least to define:

- Number of z/OS disk volumes per host to be saved
- Number of z/OS tapes drives required to run the backup procedure
- Backup stacking factor, since we use high-capacity tapes that holds more than a z/OS disk

Backup operation parallelism

Backup parallelism is determined by the number of tape drives we have to hold backup data and by the impact of performance operation on the production environment, when the backup process is performed concurrently with the production applications. The first is normally the most limiting factor.

Based on the parallelism we can estimate the time required by the backup process to run. If this is greater than the available backup window, we have to increase the number of tape drives and thus the backup parallelism to have the completion time within the backup window available.

Backup stacking factor

To limit the number of tapes to be handled during the process, particularly during the off-site operations, we use high-capacity tapes to hold backups. Those tapes have enough capacity to hold more than one z/OS disk, therefore we need to define how many z/OS disk backups fit in a single tape.

This is called the Backup Stacking Factor. It impacts the parallelism recovery operations, since a two-disk restore from the same backup tape must be performed sequentially even when more tape drives would be available to accommodate another restore operation.

10.2.2 Define recovery environment requirements

Once we have defined what must be saved, we have all the information required to define the recovery environment. The recovery environment must be defined in all its extent — that is, not only the resources needed for the recovery phase, but also all the resources required to run the z/OS production environment from the alternate site, including workplaces and network communications.

For the recovery phase, we need at least to define:

- The number of z/OS disk volumes required to host the restored environment
- The number of z/OS tapes drives required to run the recovery procedure with the required parallelism of operations to match RTO of the recovery procedure

Restore operation parallelism

Restore parallelism is determined by the number of tape drives available to hold the restore operations in parallel data and by the stacking factor used during the backup operation.

Based on the parallelism we can estimate the time required by the restore process to run. If this is greater than the required RTO, we have to increase the number of tape drives, or we might change the backup stacking factor if all backup tapes are mounted and we still have tape drives available at the recovery site.

10.2.3 Define testing environment requirements

Depending on the recovery strategy you want to adopt, the testing environment could be a subset of the recovery environment defined above, or exactly the same, in case your recovery strategy doesn't plan to run recovery tests for only self-contained subsets of the production environment.

Anyway, as before, at least we have to define:

- Number of z/OS disk volumes required to host the restored environment or subset of the environment
- Number of z/OS tapes drives required to run the recovery procedure with the required parallelism of operations to match RTO of the recovery procedure

10.2.4 Define backup procedure

Once we have identified what to save, we need to define how to do it, that is, the steps that have to be performed to produce a set of backup tapes, and the timing and frequency of the backup operation instance.

One thing to remember is that the FlashCopy Consistency Group works only from the ESS Copy Services Web Interface and produces an exact copy of the source disk, including the disk label. This means that it will not be possible to bring the target volumes online from the same z/OS system in order to dump them on tape.

To do this, we must first change the label of the FlashCopy target disk to be different from the source one, using the ICKDSF REFORMAT command.

One of the optional parameters of the REFORMAT command can be used to define the volume as a *conditional volume*. When DFSMSdss dumps a conditional volume, it produces a copy of the volume as it would have with the original label, instead of using the current label the disk has.

This procedure could be used to allow a DUMP of the complete FlashCopy target environment without having to start another z/OS partition.

Backup procedure

A possible backup procedure could consist of the following steps:

1. Execute the *FlashCopy Consistency Group* task, which also sets the ELB condition.
2. Wait for all FlashCopy relationships to be established.
3. Execute the *FlashCopy Consistency Group Created* task, which cleans the ELB condition.
4. Execute ICKDSF REFORMAT DUMPCOND(SET) jobs to change the label of FlashCopy target disks.
5. Vary all FlashCopy target disks online to the z/OS system.
6. Execute DFSMSdss DUMP jobs to produce a set of recovery tapes.
7. Monitor, control, and wait for all DFSMSdss DUMP jobs to complete.
8. Prepare a full set of JCL to drive the recovery operation at the alternate site, which at least includes:
   - ICKDSF INIT jobs to prepare the recovery environment to receive data
   - DFSMSdss RESTORE jobs to reload backups from tapes
9. Prepare the documentation needed to drive the recovery process.
10. Vary all FlashCopy target disks offline to be ready to run the next instance of the backup procedure to the z/OS system.

Timing and frequency

After having defined the procedure steps, we have to determine when, during the day, this procedure should be executed. Let's say that there is no general rule that applies here, since each environment has it own peculiarities.

For choosing the *right* time, we have to consider those peculiarities in terms of:

- Processing windows:

  These are existing and defined times that can be associated with the start or the end of applications, subsystems, processing phases, etc.

- Low activity windows:

  The backup procedure will introduce an additional workload; we wouldn't want to execute it during the most critical hours.

- Resource contentions:

  The backup procedure will use resources that might be used by others, and might not be sufficiently available to accommodate all requests.

- Backup windows:

  Our procedure requires time to be completed. Also, it might require controls and human intervention; if it lasts for 18 hours, it is quite impossible to run more than one instance per day.

- Recovery strategy:

  In order to minimize the activity and time required to restore processing at the alternate site, a strategy is necessary. For example, a backup taken before the batch processing might require the batch to be executed at the recovery site before providing IT services back to the end users. This requires more time (RTO) compared to a solution where the backup is taken after the batch phase.

10.2.5 Define off-site procedure and timing

After having executed the backup procedure, we obtained a full set of tapes that must be transferred to a safe vault location and kept there for a determined amount of time (retention period), which depends on the frequency of the backup process and on the number of backup versions we would like to maintain in the safe vault.

The activities we have to plan for include:

- Collecting backup tapes
- Preparing tapes for shipping
- Receiving expired backup tapes from the vault
- Recycling expired backup tapes
- Preparing a new set of backup tapes to be available for the next backup procedure instance

Along with the tapes, we have to collect, prepare, and ship all the documentation that might be required at the alternate site during the recovery process. This documentation must be stored in the vault along with the backup tapes it belongs to.

10.2.6 Define the restore procedure

Once we have identified what to save and how to save it, we need to define how to restore it. We must define the steps that have to be performed to bring the saved production environment back on disk at the alternate site, as well as the parallelism of restore operations that can be sustained with the resources we have available in the recovery environment.

Restore procedure

Suppose that we have a minimal z/OS system up and running at the alternate site. A possible Restore Procedure could consist of the following steps:

1. Execute ICKDSF INIT jobs to prepare z/OS disks at the recovery environment.
2. Execute DFSMSdss RESTORE jobs as documented on the information accompanying the tape backups.
3. Monitor, control, and wait for all DFSMSdss RESTORE jobs to complete.
4. Shut down the z/OS system used for driving the recovery process.
5. IPL the production z/OS system from the restored environment.

10.2.7 Define automation requirements

Automation is one essential component of a working disaster recovery procedure, since it provides a way to automatically perform tasks that would normally require personnel with critical skills. Such skilled staff might not be available in case of a disaster.

In our scenario, all the JCL to implement the solution could be created manually, but in a real scenario where thousands volumes are involved, manual operations are time consuming and prone to errors.

A basic automation procedure is to have a unique table that links a production volume with the corresponding FlashCopy target volume with the corresponding unit address and volume serial in the recovery environment, with the backup data set name and sequence number on tape.

Starting from such a table, is quite easy to implement REXX execs that generate all the required JCL automatically, and keep the table up-to-date as long as the procedure runs.

10.3 Implement the solution

To implement the solution, we have to follow two main steps:

1. Pre-define and generate all the tasks and JCL required.
2. Build a procedure that links all these tasks and JCL in a working disaster recovery procedure.

The information provided in the following paragraphs is not exhaustive, but it does provide some guidelines on the whole implementation phase for a traditional dump-restore disaster recovery procedure.

10.3.1 Create FlashCopy Consistency Group task

Note: We will refer to this task as the *FC_Freeze_CG task*.

To define the FlashCopy Consistency Group creation task, follow these steps:

1. In the ESS Copy Services menu, select **Volumes**, then select source and target LSSs (see Figure 10-2).
2. Left-click to select the FlashCopy source volume.
3. Right-click to select the corresponding FlashCopy target volume.

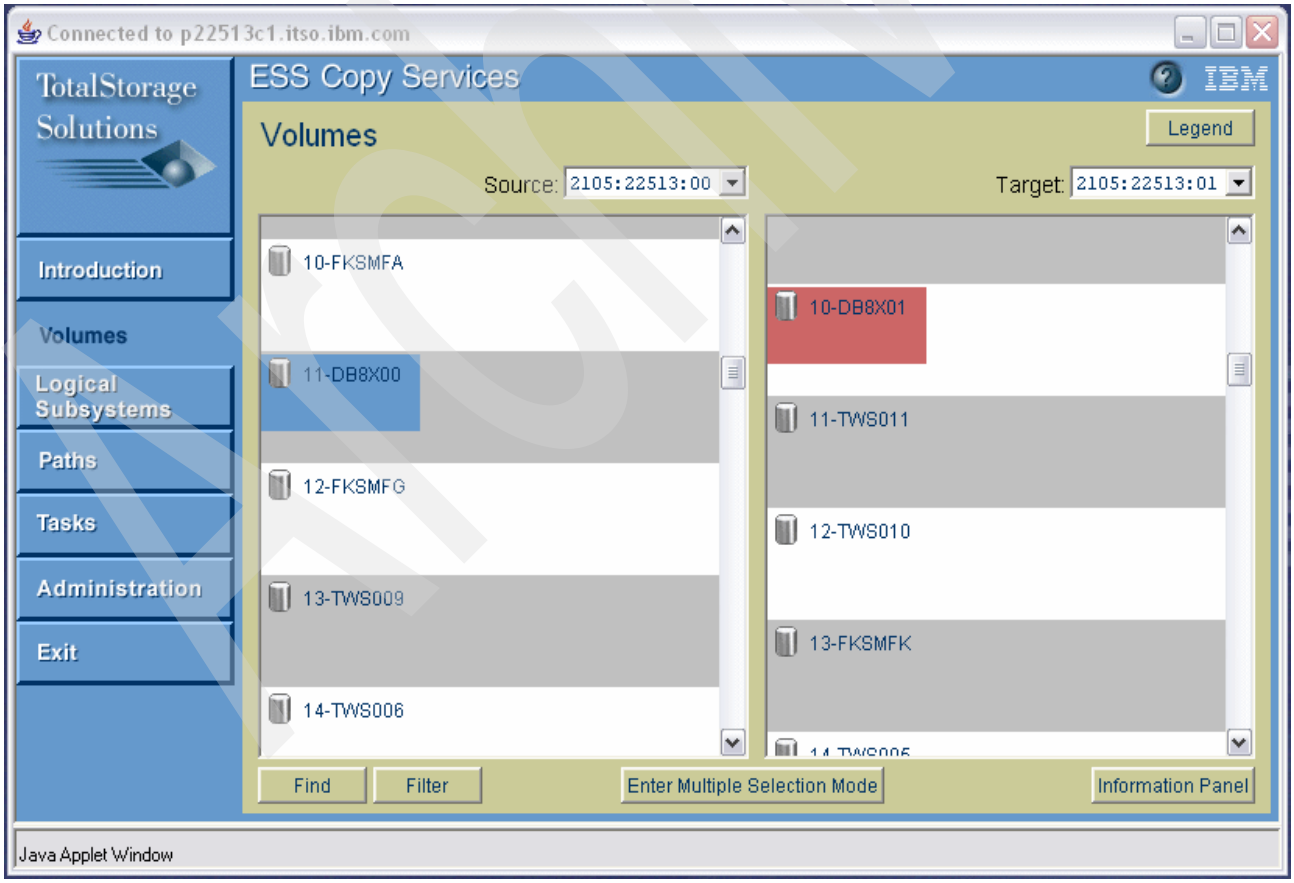


Figure 10-2 ESS Copy Services Web Interface - Volumes panel

4. Right-click again to bring up the Select Task Type panel (see Figure 10-3).
5. Select **Establish FlashCopy pair** option and click **Next**.

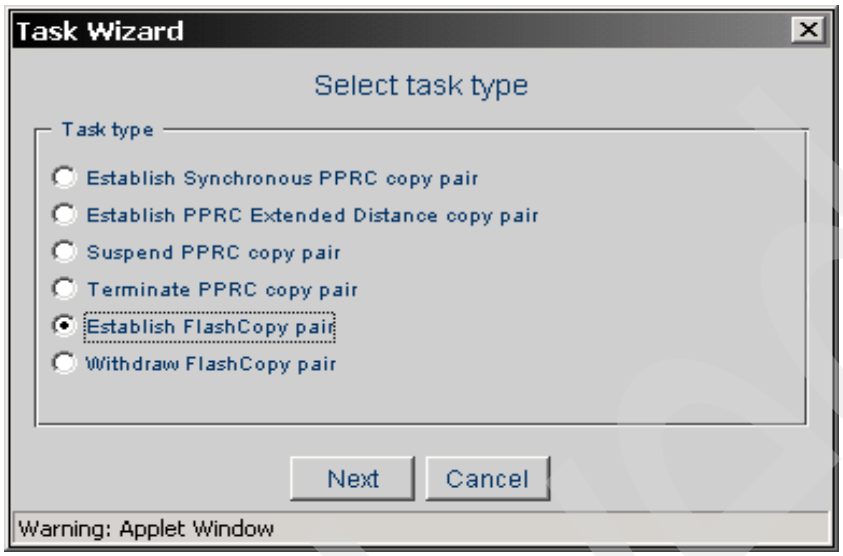


Figure 10-3 ESS Copy Services Web Interface - Select task type panel

6. At the Select copy options panel, shown in Figure 10-4, select **Freeze FlashCopy Consistency Group,** and **Permit establish if target is online** option, and click **Next**.

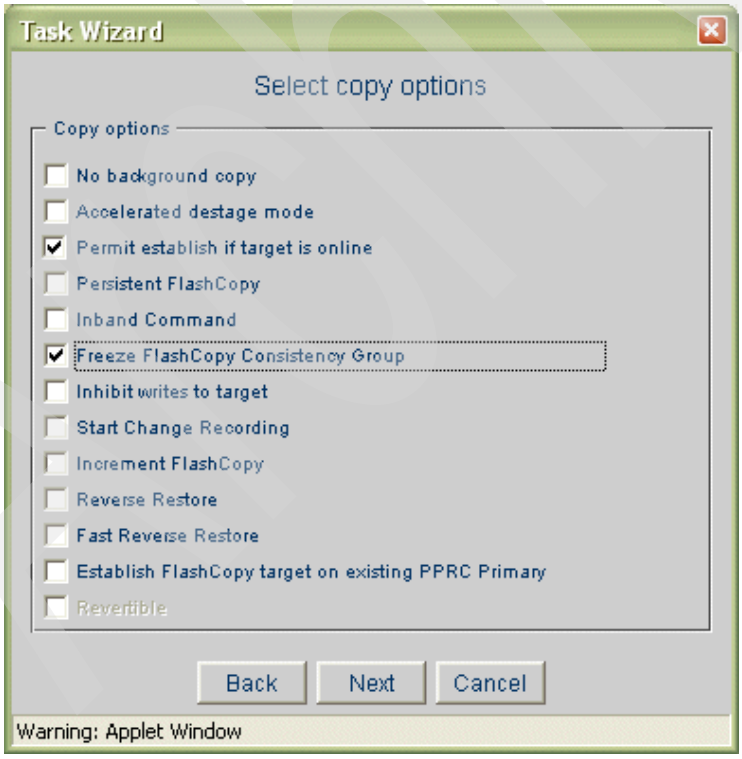


Figure 10-4 ESS Copy Services Web Interface - Select Copy Option panel

7. Enter the name and description of the task at the Define Task panel, then select **Save** to save the defined task.

8. Repeat steps 1 through 6 for each source volume that will be part of the Consistency Group.
9. When the establish tasks have been created for all volumes in the Consistency Group, click **Tasks** in the navigation frame to go the ESS Copy Services Tasks panel (see Figure 10-5).
10. Click to highlight the first *Establish FlashCopy pair* task defined in step 6.
11. Press the Ctrl key and click to highlight all the other tasks defined in step 7.

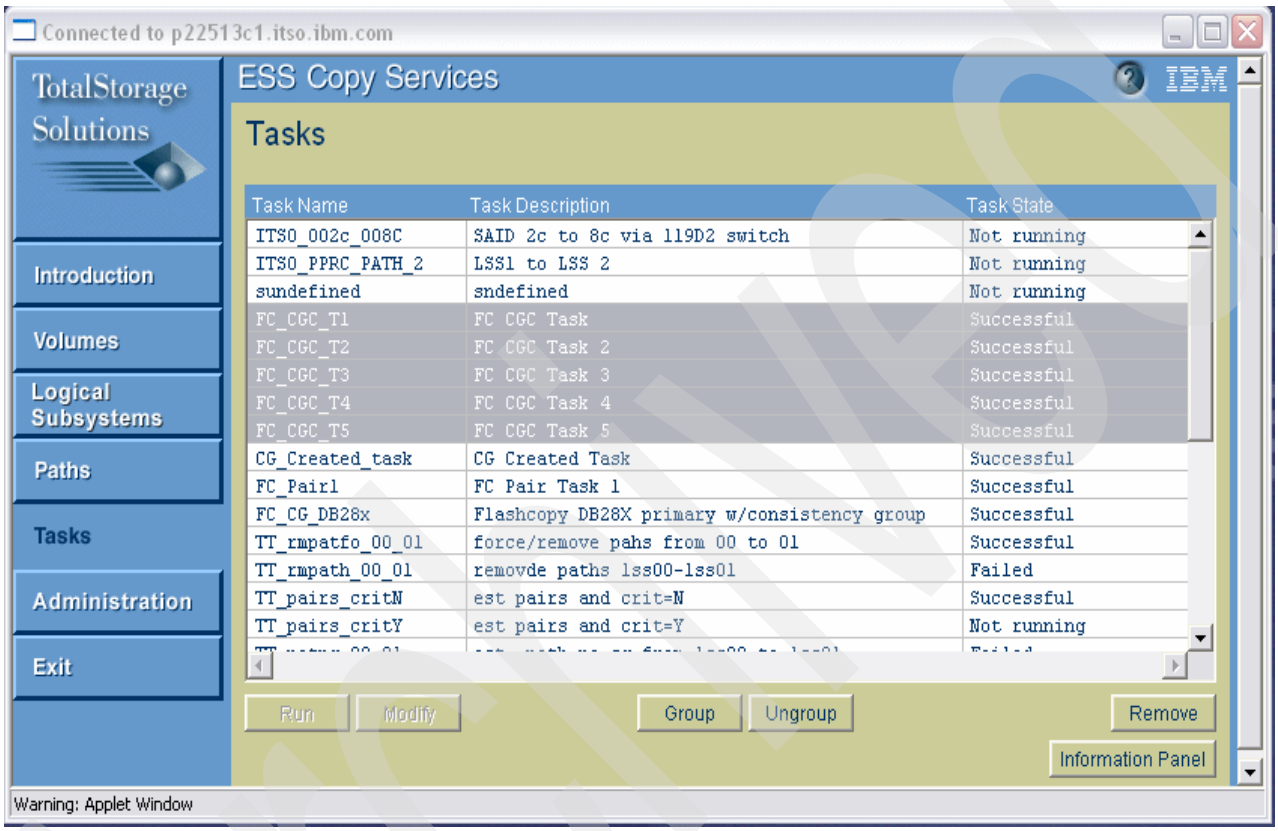


Figure 10-5 ESS Copy Services Web Interface - Tasks panel

12. When all required tasks have been highlighted, click **Group** to define a composite task that comprises all highlighted tasks.

> **Tip:** In the previous procedures, if you are going to select several volume pairs, you have the alternative of initially clicking the **Enter Multiple Selection Mode** button and then iterating through steps 2 and 3 until you select all the volumes. Once all the volumes have been selected, you then proceed with steps 4 through 7, with the procedure ending at step 7.

10.3.2 Create FlashCopy Consistency Group Created task

> **Note:** We will refer to this task as the *CG_Created_task*.

To define the FlashCopy Consistency Group created task:

1. From the ESS Copy Services Web Interface, select Logical Subsystems.

2. From Logical Subsystems panel in Figure 10-6, click to select the **source** LSS (the one which contains source volumes from the FlashCopy Consistency Group defined).

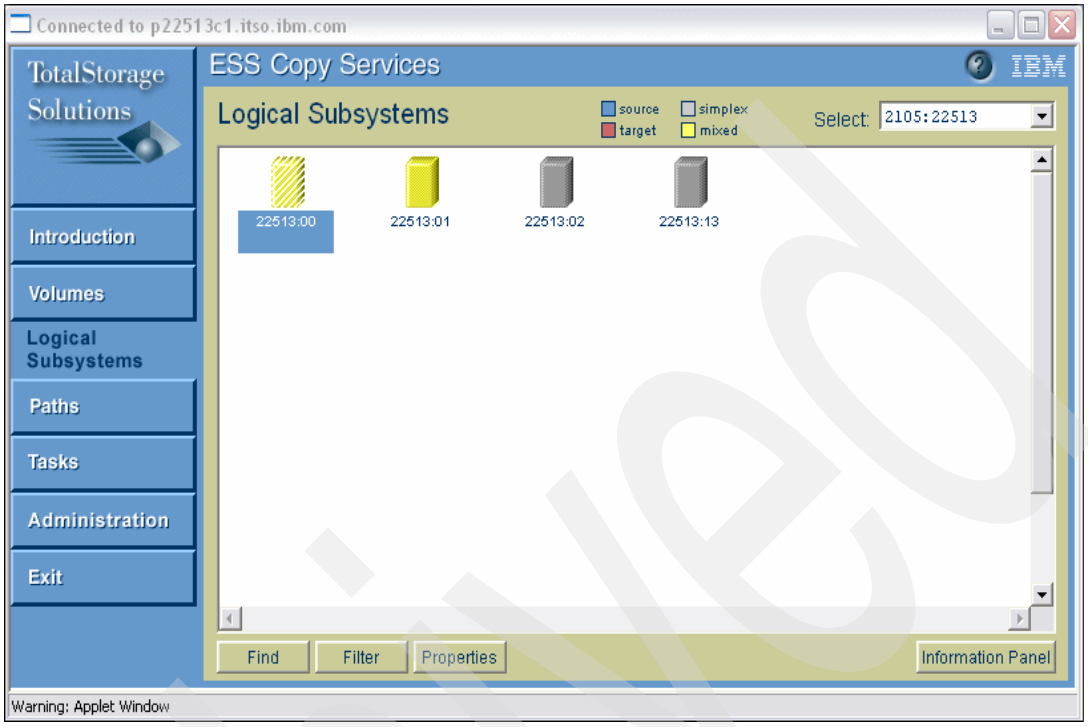


Figure 10-6 ESS Copy Services Web Interface - Logical Subsystems panels

3. Right-click *twice* to bring up the Select Task Type panel wizard (Figure 10-7).

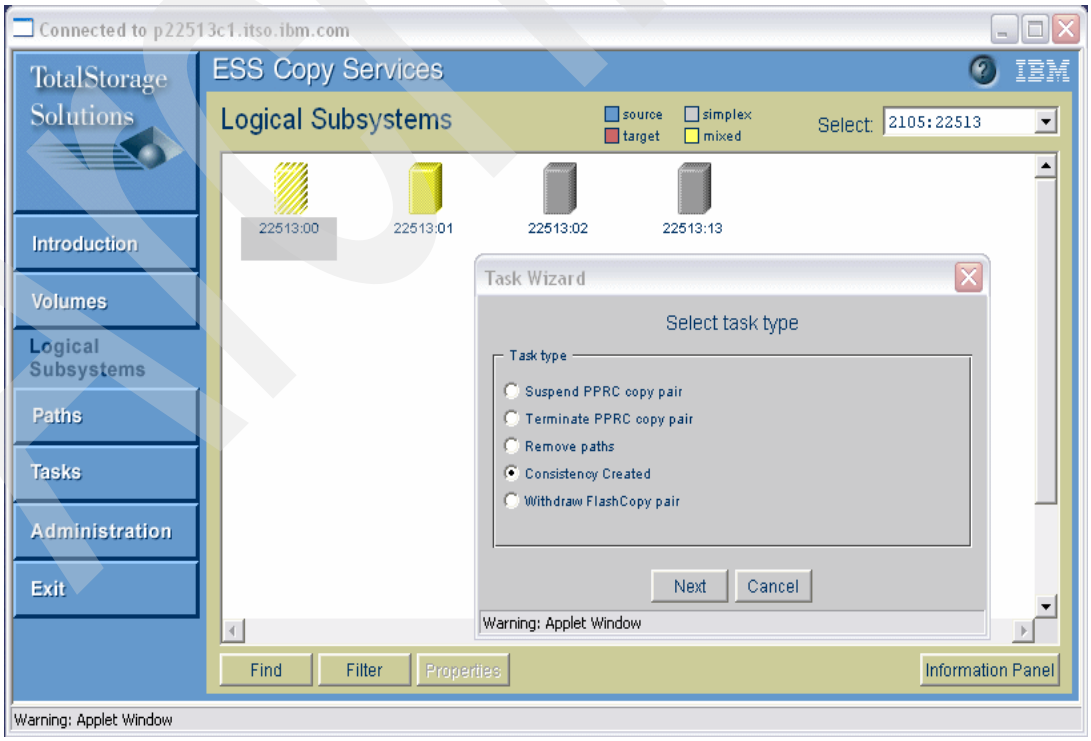


Figure 10-7 ESS Copy Services Web Interface - LSS panel - Select task type

4. Select **Consistency Created** and click **Next**
5. This brings you to the Select Copy Options wizard panel (Figure 10-8).

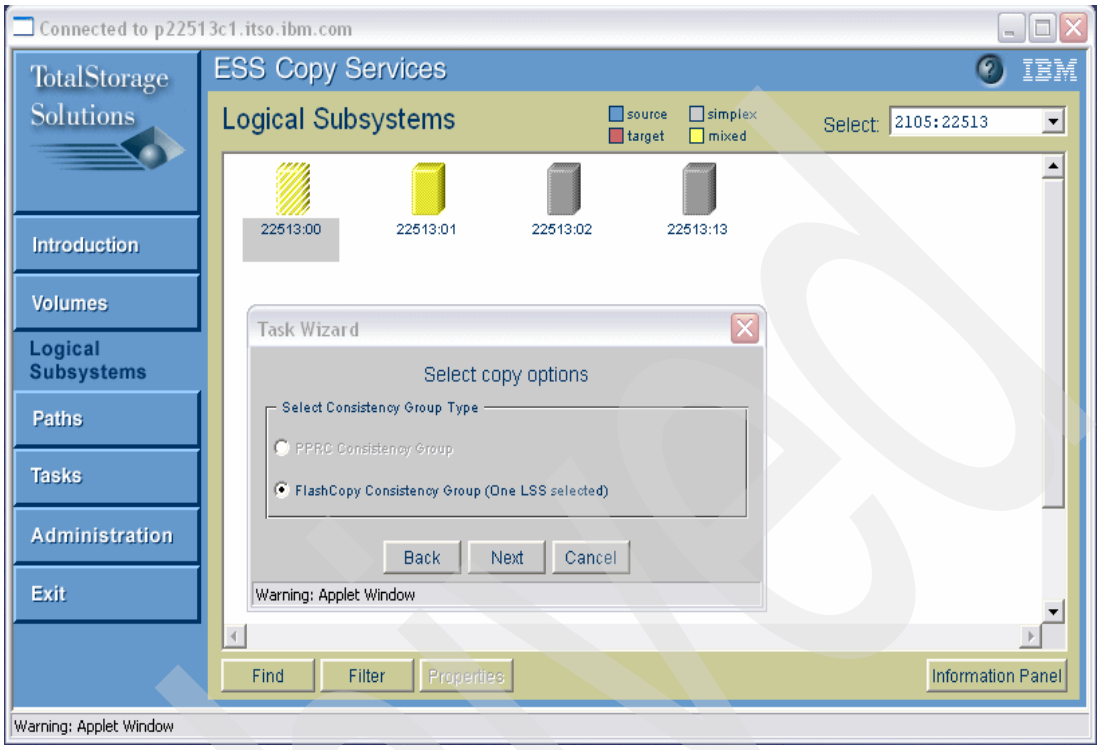


Figure 10-8 ESS Copy Services Web Interface - LSS panel - Select copy option

6. Select **FlashCopy Consistency Group (One LSS selected)** and click **Next**.
7. Enter the name and description of the task at the Define Task panel, and select **Save** to save the defined task.
8. Repeat steps 1 through 6 for each LSS that contains source volumes in the Consistency Group.
9. When tasks have been created for all volumes in the Consistency Group, click **Tasks** in the navigation frame to go the ESS Copy Services Tasks panel.
10. Click to highlight the task defined in step 6.
11. Press the Ctrl key and click to highlight each task defined in step 7.
12. When all required tasks have been highlighted, click **Group** to define a composite task that comprises all highlighted tasks.

10.3.3 Create ICKDSF REFORMAT jobs

After the FlashCopy task is executed, all the target volumes have the same label as source volumes and we won't be able to have them online in the z/OS system for the DUMP procedure.

To vary them online, we have first to reformat all the target volumes label using ICKSDF. We must also specify DUMPCOND(SET) parameter to indicate the target volumes as conditional volumes that will be dumped by DFSMSdss with the source volume label.

The JCL in Example 10-1 shows the job required to reformat all the target volumes.

Example 10-1 ICKDSF job to reformat a z/OS volume and set DUMPCONDITIONING option

```
//CLIPVOL  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//        NOTIFY=&SYSUID,REGION=0M,TIME=30
//*****************************************************
//**   VERIFY : ORIGINAL VOLUME SERIAL               **
//**   VOLID  : TARGET   VOLUME SERIAL               **
//*****************************************************
//CLIPVOL  EXEC PGM=ICKDSF,REGION=4096K,PARM=NOREPLYU
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
   REFORMAT UNIT(822D) VERIFY(DB8X01) VOLID(DMPX01) DUMPCOND(SET)
/*
```

10.3.4 Create DFSMSdss DUMP jobs

At this time we need to prepare the JCL required to save all the reformatted FlashCopy target volumes on tapes.

The JCL in Example 10-2 is an example for creating the jobs required to dump the target volumes on tape.

Example 10-2 DFSMSdss DUMP job

```
//DUMPCPB1 JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//        NOTIFY=&SYSUID,REGION=0M,TIME=1440
//*
//DUMP00    EXEC PGM=ADRDSSU,REGION=3000K
//IN        DD VOL=SER=DMPX01,UNIT=3390,DISP=SHR
//OUT       DD DSN=PAOLOR1.DB8X01.DUMP,
//             DISP=(,CATLG),UNIT=TAPE,LABEL=(1,SL)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  DUMP  FULL -
        INDDNAME(IN) -
        OUTDDNAME(OUT) -
        ALLDATA(*) -
        ALLEXCP -
        CANCELERROR -
        ADMIN
//*
```

If you are dumping to tape and you have a tape drive with the compaction feature, you do not need to specify the COMPRESS keyword to use hardware data compaction.

10.3.5 Create ICKDSF INIT jobs

All the volumes at the recovery site must be prepared to receive data.

Although it is not required for the target volume in a DFSMSdss restore operation to be empty, since DFSMSdss simply replace its contents during the RESTORE process, as a best practice, we suggest to initialize all the restore DASD environment.

This prevents you from having problems due to old data on recovery volumes, should the restore operations fail for any reason. It also simplifies the DFSMSdss RESTORE procedure and its maintenance, since the relationship of source volume to recovery volume is always pre-defined.

Use the JCL in Example 10-3 for creating the jobs required to INITIALIZE the recovery environment volumes.

Example 10-3 ICKDFS job to INIT a volume

```
//DSFINIT JOB (999,POK),'PAOLOR1',CLASS=W,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*******************************
//*  VFY : SOURCE
//*  VOLID :  TARGET
//*******************************
//INIT      EXEC PGM=ICKDSF,PARM='NOREPLYU'
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
 INIT UNIT(822C) VOLID(DB8X00) VFY(RECX00) -
      PURGE VTOC(0003,0,90) INDEX(0001,0,30) SG
//*
```

10.3.6 Create DFSMSdss RESTORE jobs

Once we have determined the relationships between source volumes and recovery volumes, we can build all jobs to restore all the backup data on tape.

Use the JCL in Example 10-4 for creating the jobs required to RESTORE the volumes in the recovery environment.

Example 10-4 DFSMSdss RESTORE job

```
//RESTCPB2  JOB (999,POK),'RESTORE',CLASS=A,MSGCLASS=T,
//             NOTIFY=PAOLOR1,TIME=1440
//*
//REST00    EXEC PGM=ADRDSSU,REGION=3000K
//OUT       DD VOL=SER=DB8X00,UNIT=3390,DISP=OLD
//IN        DD DSN=PAOLOR1.DB8X00.DUMP,DISP=SHR,
// VOL=SER=(Z91492),UNIT=TAPE,LABEL=(1,SL)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  RESTORE FULL -
        INDDNAME(IN) -
        OUTDDNAME(OUT) -
        CANCELERROR -
        ADMIN -
        PURGE
//*
```

10.3.7 Create a Disaster Recovery Procedure (DRP)

Once we have prepared all the tasks and JCL required, we need to build and document the Disaster Recovery Procedure, which instructs the users on the activities that have to be performed to produce backups, maintain backups, and recover the IT operations at the alternate site.

Execute backup of the production environment

Here we provide a procedure for backing up the production environment:

- Create consistent copy:
  - Run the *FC_Freeze_CG_task*. The freeze option in the defined task causes volumes to remain in an ELB condition until the condition is reset or its time-out value expires (default is two minutes).
  - Wait for all the source-target FlashCopy relationships to be established.
  - Run the *CG_Created task*. This task resets the ELB condition, making the volumes available for normal I/O activity.
- Prepare to DUMP FlashCopy target volumes:
  - Execute ICKDSF REFORMAT DUMPCOND(SET) jobs to change the label of FlashCopy target disks.
  - Monitor, control, and wait for all ICKDSF REFORMAT jobs to complete.
  - Vary all FlashCopy target disks online to the z/OS system.
- DUMP FlashCopy target volume:
  - Execute DFSMSdss DUMP jobs to produce a set of recovery tapes.
  - Monitor, control, and wait for all DFSMSdss DUMP jobs to complete.
- Prepare tape backup accompanying information:
  - Generate ICKDSF INIT jobs to prepare the recovery environment.
  - Generate DFSMSdss RESTORE jobs to reload backups from tapes.
  - Prepare the documentation needed to drive the recovery process.
- Prepare for the next backup instance:
  - Vary all FlashCopy target disks offline to be ready for the run of the next instance of the backup procedure to the z/OS system.
  - Verify that you have all the resources required by the new backup instance.

Execute off-site procedure

- Prepare to ship backup tapes:
  - Collect backup tapes.
  - Prepare tapes for shipping (packaging and documentation).
  - Ship backup tapes.
- Recycle an expired set of backup tapes:
  - Receive expired backup tapes from the vault.
  - Execute Tape Management System procedures to recycle expired backup tapes.

Execute restore of the production environment

- Execute ICKDSF INIT jobs, as documented on the tape backup accompanying information, to prepare z/OS disks at the recovery environment.
- Monitor, control, and wait for all ICKDSF INIT jobs to complete.
- Execute of DFSMSdss RESTORE jobs as documented on tape backup accompanying information, to restore the production environment back on disk.
- Monitor, control, and wait for all DFSMSdss RESTORE jobs to complete.

IPL production z/OS at the recovery site

Here are the steps for the recovery site:

- Shut down the z/OS system used for driving the recovery process.
- IPL your production z/OS system at the recovery site.

> **Important:** Document in detail each step that has to be performed to IPL a copy of your production z/OS system at the recovery site. Write it assuming that whoever will perform that operation might be unfamiliar with your system.

10.4 Run the solution

Now that the solution is built, we have to run its backup and offsite part with the pre-defined frequency (daily in most cases).

We hope you will be lucky enough that you never have to execute the recovery part. However, you cannot wait for a disaster to see if your recovery procedure works. You need to be prepared to execute it.

10.4.1 Execute periodic disaster recovery testing

Periodic testing is part of best practice. Periodic means that the recovery procedure has to be exercised with a frequency such that it generates confidence in performing all the required activities, procedures, etc.

Most customers with a proven recovery procedure execute at least two recovery tests per year.

10.4.2 Maintain the solution

Time changes all things, and this includes your production environment (new applications, new DASD spaces, new requirements, etc.) as well as your recovery environment.

Once the disaster recovery procedure has been implemented, it has to be maintained as if it were one of the production procedures in your center — and it is, really. It is the most important one, since it is all you will have to rely on when the time comes.

For maintaining the solution, you need to implement a change management program on at least the following levels:

1. Daily: Verify that what has to run hasn't generated any errors; and if it has, fix them.
2. Weekly: Verify that the off-site runs properly and that you have the required level of backup versions in the vault.
3. Monthly: Simulate a recovery exercise, through a walkthrough methodology, which means to read the recovery procedure and try to simulate without really executing any activity. Verify that all the required resources are available and try to simulate some "what if" scenarios, such as "what if few critical resources are not available", or "what if one of the backup tapes is missing or unreadable", etc.
4. Periodic tests: Verify test results and fix any errors.

10.5 Additional considerations

In this section we mention some more things for you to keep in mind regarding recovery.

DB2 is only a part of the “shop”

This IBM Redbook deals with DB2, but DB2 is only one part of a complex production IT environment. If you are going to build a disaster recovery procedure for the DB2 environment only, be sure to build on technologies and solutions that can be easily adopted by the other parties of your IT environment.

Do not forget tapes

One of the most common errors in a disaster recovery procedure is that the procedure tends to concentrate on disk data only. Remember that your IT environment probably needs tape data also.

If you use tapes for archive logs and image copies, and your procedure is for disk data only, you lose the capability to do a rollback or a forward recovery from an image copy, because the information you need is lost.

In case you do not want to deal with tape data, we suggest that you keep on disk at least one week of dual archive logs and the latest two versions of each image copy.

Make sure that both copies are not allocated the same DASD (or loaded on the same tape volume) and that the ICF catalog correctly includes them. You can exploit DFSMS features to control all the above operations.

11

Global Copy PPRC-XD

PPRC-XD, also called Global Copy, is a recommended solution for remote data copy and off-site backup without impacting the write performance of local DB2 (and other subsystems), which is particularly relevant when backup is implemented over continental distances. PPRC-XD can also be used for DB2 disaster recovery solutions based on periodic point-in-time copies of the data.

We have already introduced PPRC-XD in 7.2, “PPRC Extended Distance (PPRC-XD)” on page 102. In the current chapter we describe in more detail how, together with FlashCopy, it can provide a general disaster recovery solution.

This chapter covers the following topics:

- General PPRC-XD Global Copy Configuration
- Creating your Consistency Group
- Applying additional log to a Consistency Group
- Considerations

11.1 General PPRC-XD Global Copy Configuration

Figure 11-1 illustrates the general PPRC - XD configuration where there is a local PRIMARY DB2 subsystem at the primary site, and a recovery site where you see “SECONDARY” and “TERTIARY”. The two sites are connected through channel extenders if the distance is greater than metro.

See Chapter 7, “Peer-to-Peer Remote Copy” on page 97 for a description of the general technology.

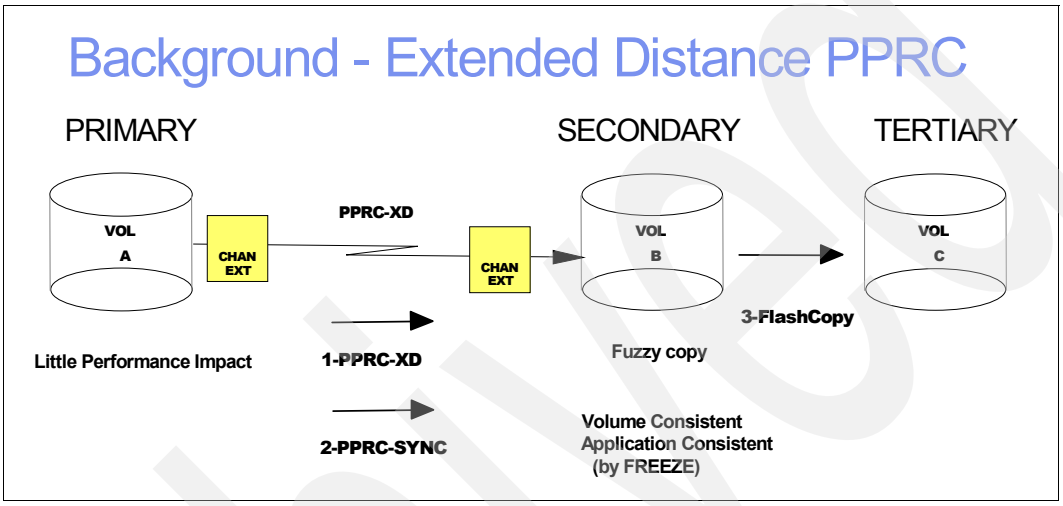


Figure 11-1 PPRC Extended Distance background

PPRC-XD provides a non-synchronous long-distance copy option whereby write operations to the primary ESS are considered complete before they are transmitted to the secondary ESS. This non-synchronous operation results in a “fuzzy copy” at the secondary site; however, through operational procedures such as the DB2 -SET LOG SUSPEND command, a point-in-time (PIT) consistent copy at the remote site can be created that is suitable for disaster recovery purposes.

Since the critical issue for DB2 disaster recovery is the consistency of data at the recovery site, the interesting aspect is how a Consistency Group is obtained.

11.2 Creating your Consistency Group

In PPRC-XD terms, catchup methodology is used. For DB2, we call it Consistency Group creation, as shown in Figure 11-2.

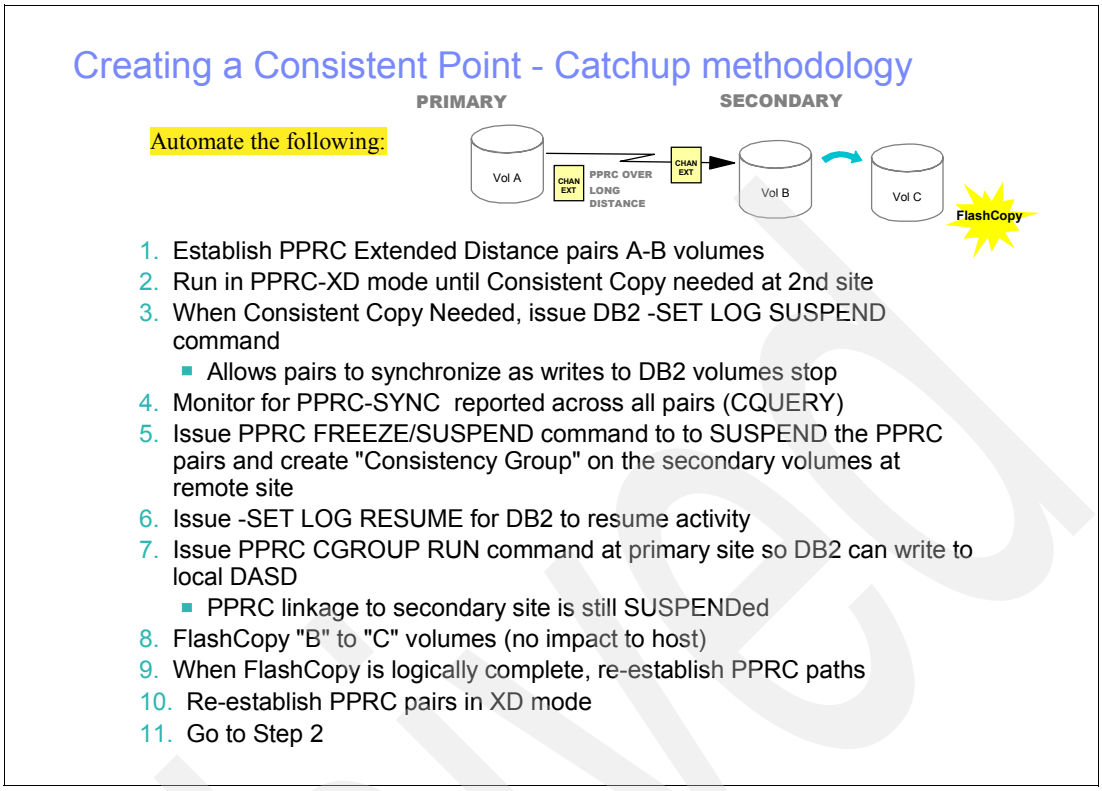


Figure 11-2 The DB2 Consistency Group Procedure for Global Copy Scenario

Refer to Figure 11-2 to follow the explanation presented in this section.

1. You can establish the PPRC pairs between the primary and secondary sites as is shown in *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04.

2. Run your DB2 subsystems in XD mode, remembering that tracks may be sent in any order and there is no consistency at the secondary site during this time. The status of the volume pairs is called duplex rebuild, as the volumes are in a constant state of rebuilding in duplex mode as writes at the primary site occur.

3. When it is necessary to create your Consistency Group, you begin the catch-up process by first issuing the DB2 -SET LOG SUSPEND command. This step should be done during a very low DB2 write activity window.

   Other subsystems have to issue a PPRC Go-to-SYNC command and incur local delays as the volumes become synchronized for as long as takes to synchronize them. DB2 does not have to suffer the synchronous write penalty, as its log suspension externalizes all the log buffers, updates the BSDS with the highest written RBA, and then holds the log write latch. Without the ability to write to DASD, writes dry up and all tracks will be sent to the recovery site.

   Since the log write latch is taken, DB2 writes will trickle to a stop. Note that this command also establishes the starting point for later DB2 log apply. This point is applicable only if you apply additional log after you have FlashCopied the point of consistency.

4. During this time you should issue the PPRC CQUERY to determine when all the volumes are completely synchronized. CQUERY operates at the volume pair level, not the LSS level.

5. When all volumes are synchronized (duplexed), you can issue either of the PPRC commands: CGROUP FREEZE or SUSPEND. This action will create the “Consistency Group” across all DASD for this subsystem or data sharing group.

   FREEZE occurs on an LSS basis while SUSPEND applies to volume pairs, thus FREEZE is easier to implement. The CGROUP FREEZE simultaneously suspends all the volume pairs between the LSS pairs and will also terminate the paths between LSSs. The FREEZE presents an extended long busy (ELB) for any writes to the volumes for a default time of 2 minutes. Since there are no writes occurring to any of the volumes in the DB2 subsystem at either site, the CGROUP FREEZE should be easily obtained.

   This leaves the secondary volumes with a consistent point-in-time copy of the primary volumes, and with the volumes suspended. This suspended state enables you to restart the DB2 write activity on the primary volumes, while you FlashCopy the secondary volumes that now have a consistent point-in-time copy.

6. Issue the PPRC command CGROUP RUN at the primary site. DB2 can then write to the primary DASD. Because of the CGROUP FREEZE, the PPRC linkage from the primary to the secondary site is still SUSPENDed.

7. Now you can resume DB2 write activity by issuing -SET LOG RESUME.

   Issue this command to resume update activity as soon as the freeze templates. These updates will not be propagated to the secondary volumes — as volume pairs are suspended and LSS paths are discontinued. In the meanwhile, PPRC will keep track of the updated tracks on the primary volumes to be able to do an incremental re-synchronization when the PPRC pairs are later reestablished.

8. As soon as the FREEZE occurs in Step 5, you must capture the Consistency Group by taking a FlashCopy of the consistent “B” volumes to the “C” set of volumes.

   Once the FlashCopy relationship is established, you can bring the primary and secondary volumes out of the suspended state. Bringing them into the duplex-pending XD state, done with an incremental re-synchronization (copy out-of-sync only), will resume the mirroring on the secondary volumes.

9. As soon as the FlashCopy relationships between all pairs are logically complete, you can re-establish the PPRC paths according to instructions in *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04.

   Before reestablishing the volume pairs’ XD relationships, you must redefine the LSS paths that were terminated by the freeze operation.

10. Re-establish the PPRC pairs in XD mode.

11. Now you are back in the same mode as exists in Step 2. The procedure is complete, the consistent point-in-time copy is on the tertiary volumes, and you are ready for the next checkpoint.

The tertiary volumes can be used in case of disaster to simply restart DB2 as of the point of consistency. The RPO is the amount of time between FlashCopies.

Before restarting DB2, since this will be the only valid copy of your subsystem, we strongly recommend that you take a FlashCopy of all the disks. To do this, you can FlashCopy using the PPRC secondary disks as FlashCopy target volumes, once PPRC has been terminated.

11.3 Applying additional log to a Consistency Group

Since you have a Consistency Group as of the last FlashCopy, it would be nice to be able to apply additional log to the DB2 objects to reduce the RPO in the event that the amount of time between FlashCopies is significant.

In order to accomplish that, you need to send all the logs created since the last FlashCopy. If the amount of active log is enough to avoid “wrapping” the log between FlashCopies, then it is not necessary to include DASD archives.

If archives are on tape, then an automated tape library is needed if it is desired to apply them.

The most likely candidate for recovery currency would be to the end of the “-1” active log. DB2 has switched logs, has archived this log, and is now writing on a new active log. The log writes to the former active log have ceased and now those tracks have no activity and should arrive quickly at the recovery site. That theory does not take into account the variable nature of PPRC-XD.

If the “-1” active log were on isolated DASD, it might arrive quickly. However, if it is on the same volume with other active data, for example, the current active log, there will continue to be heavy activity to that volume. The same is true within an LSS. In short, since tracks for all volumes are sent in any order across a variable bandwidth, you cannot predict in what order any of them will arrive. Even if you had a current BSDS sent, you could not tell that relative track 98 for any given active log did not arrive before relative track 53.

Note: We assume that the user does not know where any given log is placed relative to any other. With the current generation of DASD technology, users today do not try to track data set placement, as it is left to SMS to manage. That is indeed a good thing.

Note: If there were a situation where of a 100 track data set, relative track 53 had not been received at the secondary site, DB2 would interpret this as a hole in the log. It would detect upon reading the first CI of track 53 that the RBA was lower than that of the previous CI. DB2 would stop at that point and return an error because the log is always ascending.

If 12 hours separated the Consistency Group FlashCopies, it would be reasonable to assume that many logs were received at the recovery site. If we want to apply them, we need a way to verify that the entire data set is complete. See 11.3.1, “How to verify a log” on page 162, which describes how you do that.

If the “-1” active log verification process is successful, you can use its ENDRBA or ENDLRSn as the SYSPITR value in the CRESTART record. Following the conditional restart you submit RESTORE SYSTEM LOGONLY.

Alternately, you might reasonably say in the above example that 10 hours of log had reached the recovery site and were intact. You could decide to bypass the verification process and pick a restart point at current time minus two hours (or another arbitrary time). Using Print Log Map and your current BSDS, you could identify an archive created at that time (another good reason for timestamping your archive logs!) and perform a Conditional Restart using the ENDRBA or ENDLRSN as your SYSPITR. Following restart, you submit RESTORE SYSTEM LOGONLY.

If you have an automated tape library with archive tapes, you can use those without going through the verification process described previously. Since it is not associated with PPRC-XD, the last one you have at the recovery site is the one you use. You prepare the Conditional Restart using archive end ENDRBA or ENDLSRN as the SYSPITR and, following restart, submit the RESTORE SYSTEM LOGONLY utility.

Tip: Before doing any restart, you should dump the FlashCopied volumes to tape in order to preserve your Consistency Group, so that you retain a copy of the point of consistency, in case there is an error somewhere in the restart and recovery process. You can FlashCopy the volumes back to their original ones instead.

11.3.1 How to verify a log

The DSN1LOGP stand alone utility can process each log and identify its start and end RBAs (or LRSNs). See Example 11-1. Those RBAs can be matched against the Print Log Map and you can determine if the "-1" active log, or any given DASD archive log can be used.

Example 11-1 DSN1LOGP Sample JCL for an active log

```
//STEPLIB DD DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSABEND DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSSUMRY DD SYSOUT=A
//ACTIVE1 DD DSN=DSNCAT.LOGCOPY1.DSxx,DISP=SHR
//SYSIN DD *
SUMMARY (ONLY)
/*
```

There are many ways to allocate the input logs, either active or archive. You can specify beginning and ending RBAs or LRSNs using the BSDS to allocate logs. This example showed just a single active log, presumably the "-1" active log. Since all we want is a check for beginning and ending RBAs on the log, we specify that we only want a SUMMARY. Since we did not enter any RBAs, the defaults are used. For more information, refer to Chapter 41 of the *DB2 UDB for z/OS Version 8 Utility Guide and Reference,* SC18-7427, under DSN1LOGP.

The RBA values received from DSN1LOGP should be matched against the BSDS to verify that the ENDRBA matched, or didn't match. If it didn't match, then try for the "-2" active log and repeat the process.

There is no guarantee that the BSDS at the secondary site is also complete. You can try the second BSDS to determine if the timestamps and data match. If they don't, a BSDS on the most current DASD archive log is the next one to check.

11.4 Considerations

In this section we list the advantages, disadvantages, and issues for you to keep in mind.

Advantages

- Local site response time is negligible while in XD mode.
- There are no distance limitations.
- -SET LOG SUSPEND makes *go to Sync* and distance response time unnecessary
- This method is similar to XRC but has lower cost and flexible bandwidth.

- Outage while DB2 is suspended is much shorter than the –Set Log Suspend/FlashCopy scenario. One is an incremental track catch-up versus full volume FlashCopies.
- -SET LOG RESUME can be issued as soon as tracks are resynchronized because pairs are split. No wait is needed even for the FlashCopy to start.

Disadvantages

- This affects DB2 only (and other applications which can be quiesced at same time).

Issues

- Data Currency (RPT) = Time between FlashCopies:
  - If there is a loss of the local site, you may be able to apply some logs to minimize RPO. There are many considerations for this approach.
- If you decide to apply logs to the Consistency Group obtained through FlashCopy:
  - V8 RESTORE SYSTEM LOGONLY utility makes this easy.
- A third set of DASD is needed for Consistency Group copy.
- It requires the -SET LOG SUSPEND to catch up.
- How many other applications must be placed in go-to-SYNC mode at the same time DB2 has logs suspended? Can they be quiesced? Will their go-t0-SYNC affect the amount of time DB2 must be suspended?
  - All these considerations must be resolved before this solution can be used as “general purpose”.
- Unlike Split Mirror, PPRC-XD does propagate application errors to the secondary site, but this aspect is not a disaster recovery issue.

Brief comparison to XRC

In summary, the distance is unlimited and the solution is less costly:

- No running of the extra address space SDM is required; fewer MIPS are consumed.
- It saves at least one z/OS license (since SDM would normally run on a dedicated LPAR).
- You will need to IPL and bring up z/OS only for tests.
- The bandwidth can be less than that for XRC, keeping in mind that it might impact your RTO.
- A point of synchronization will occur with -SET LOG SUSPEND, with the alignment completed as writes trickle to a stop.
- You create your own Consistency Group via -SET LOG SUSPEND.

12

Global Mirror PPRC

Global Mirror (Asynchronous) PPRC is the latest remote copy solution for long-distance support across two sites. It applies to both open systems and z/OS data. It is based on existing Copy Services functions PPRC-XD and FlashCopy and additional new functionality for the periodic creation of Consistency Groups across one or more ESSs and for the management and control of the Asynchronous PPRC session.

Note: This function was announced in June 2004, and was not available for testing in our scenarios.

The following publications should be referenced to complement the information presented in this chapter:

- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *z/OS DFSMSdfp Advanced Services,* SC26-7400
- *Device Support Facility Users Guide and Reference,* GC35-0033
- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547
- *IBM TotalStorage Enterprise Storage Server Web Interface User's Guide,* SC26-7448
- *IBM TotalStorage Enterprise Storage Server PPRC Extended Distance*, SG24-6568
- *IBM TotalStorage Enterprise Storage Server Command-Line Interfaces User's Guide,* SC26-7494
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04

In this chapter we describe this function and show how to use it, based on information available from manuals and redbooks. We cover the following topics:

- Overview
- Design the solution
- Implement the solution
- Run the solution
- Monitor Asynchronous PPRC
- Recovery procedures

12.1 Overview

More and more customers are using data centers separated by larger distances for disaster recovery purposes. The two main reasons are that they do have existing data centers at longer distances, and they want to provide protection against regional disasters such as terrorist attacks, earthquakes, and hurricanes. Synchronous copy solutions such as PPRC over Fibre Channel using Dense Wavelength Division Multiplexer (DWDM) have a limitation of 300 km and, as the distance increases, the associated response time delay may be unacceptable for some customer' environments. Extended Remote Copy (XRC) goes well beyond such a distance, but it is available only for z/OS installations.

Asynchronous PPRC is designed to provide a long-distance remote copy solution across two sites for open systems and z/OS data using asynchronous technology. It is based on existing Copy Services functions PPRC-XD and FlashCopy and additional new functionality for the periodic creation of Consistency Groups across one or more ESSs and for the management and control of the Asynchronous PPRC session.

Asynchronous PPRC provides automation to periodically create Consistency Groups at the primary and the secondary site, without significant impact to production applications.

Data at the remote site allows for an application restart, and lags behind the local site typically 3 to 5 seconds, minimizing the amount of data exposure in the event of an unplanned outage. The actual lag in data currency experienced will depend upon a number of factors, including specific workload characteristics and available bandwidth between the local and remote sites.

Data consistency at the remote site is internally managed across up to four (current limit) primary ESS machines located at the local site.

12.1.1 Terminology

With Asynchronous PPRC, the data consistency at the secondary site is automatically and periodically guaranteed. This contrasts with PPRC-XD, where a disruption is required to create consistency and take remote copies for each point-in-time.

The following terms are used in further explanations in the following sections.

Dependent writes

If the start of one write operation is dependent upon the completion of a previous write, the writes are dependent. Application examples for dependent writes are databases with associated log files, catalogs and VTOCs as well as VSAM indexes and VSAM data components. For instance, the data base log will be updated after a new entry was successfully written to a table space, or the VTOC will be updated if you created a new data set.

The chronological order of dependent writes to primary volumes is the basis for providing consistent data for the copy operations.

Consistency

The consistency of data is ensured if the order of dependent writes to disks or disk groups is maintained. With copy services solutions the data consistency at the secondary site is important for the usability of the data at the remote site. Consistent data for instance provides the ability to perform a data base restart rather than a data base recovery that could be hours or even days.

Data consistency across all secondary volumes spread across any number of LSSs is essential for logical data integrity.

Data Currency

This term describes the difference between the time the last data was written at the primary site and the time the same data was written to the secondary site. This difference determines the maximum amount of data you have to recover at the remote site after a disaster. Only synchronous copy solutions like Synchronous PPRC have a currency of zero. All asynchronous copy solutions have a data currency greater than zero. With the new Asynchronous PPRC solution, a data currency of a few seconds can be achieved, while data consistency is maintained by the Asynchronous PPRC process.

For Copy Services, this means that the data is not copied at the same time as the local I/O. The data is written with a time difference to the remote site. Examples of asynchronous Copy Services are:

- PPRC-XD with non-consistent data at the remote site
- Asynchronous Cascading PPRC
- XRC with consistent data at the remote site
- Asynchronous PPRC with consistent data at the remote site

Session

An Asynchronous PPRC session is a collection of volumes that are managed together when creating consistent copies of data. This set of volumes can reside in one or more LSSs and one or more ESSs at the customer's primary site. Open systems volumes and z/OS volumes can both be members of the same session; this can be helpful for multi-platform solutions.

When you start or resume a session, the creation of Consistency Groups is performed and the Master ESS controls the session by communicating with the Subordinate ESSs.

Master

The master is a term for a function inside an ESS that, by communicating with Subordinates in other ESSs, controls the creation of Consistency Groups and manages the Asynchronous PPRC session. The Master is defined in the start command for a session by addressing any LSS in the Master ESS.

The Master needs PPRC Fibre Channel communication paths to any one of the LSSs in each Subordinate ESS.

Subordinate

The subordinate is a term for a function inside an ESS that communicates with the Master and is controlled by the Master. At least one of the LSSs of each Subordinate ESS needs PPRC Fibre Channel communication paths from the Master to the Subordinate. It is used to enable the Master to create Consistency Groups of volumes in more than one ESS.

If all the volumes of an Asynchronous PPRC session reside in one primary ESS, no Subordinate is required, because the Master can communicate to all LSSs inside the primary ESS.

Consistency Group

A Consistency Group is a group of volumes in one or more ESSs whose data must be kept consistent at the remote site.

Note: Asynchronous PPRC creates Consistency Groups periodically and automatically without disruption to the host I/O in a very short time.

12.1.2 How it works

The purpose of Asynchronous PPRC is to provide recoverable copies of production data at a long distance remote site by continually forming consistent sets of data with no significant performance impact. This allows a quick restart at the remote site in case of a disaster at the primary site.

Asynchronous PPRC overcomes disadvantages of PPRC-XD and automates this type of solution so that you can get:

- A very good data currency of only a few seconds
- Automatic creation of Consistency Groups in a very short time
- No application impact at the local site
- One central point of control

The new process of Asynchronous PPRC is implemented at Licensed Internal Code level 2.4.x or higher of the ESSs and works as described in Figure 12-1 for all volumes in one or more LSSs that previously have been added to a defined session.

The creation of the Consistency Group at the local site is a quick process followed by a drain of the consistent data to the remote site, while application I/Os are continuously running.

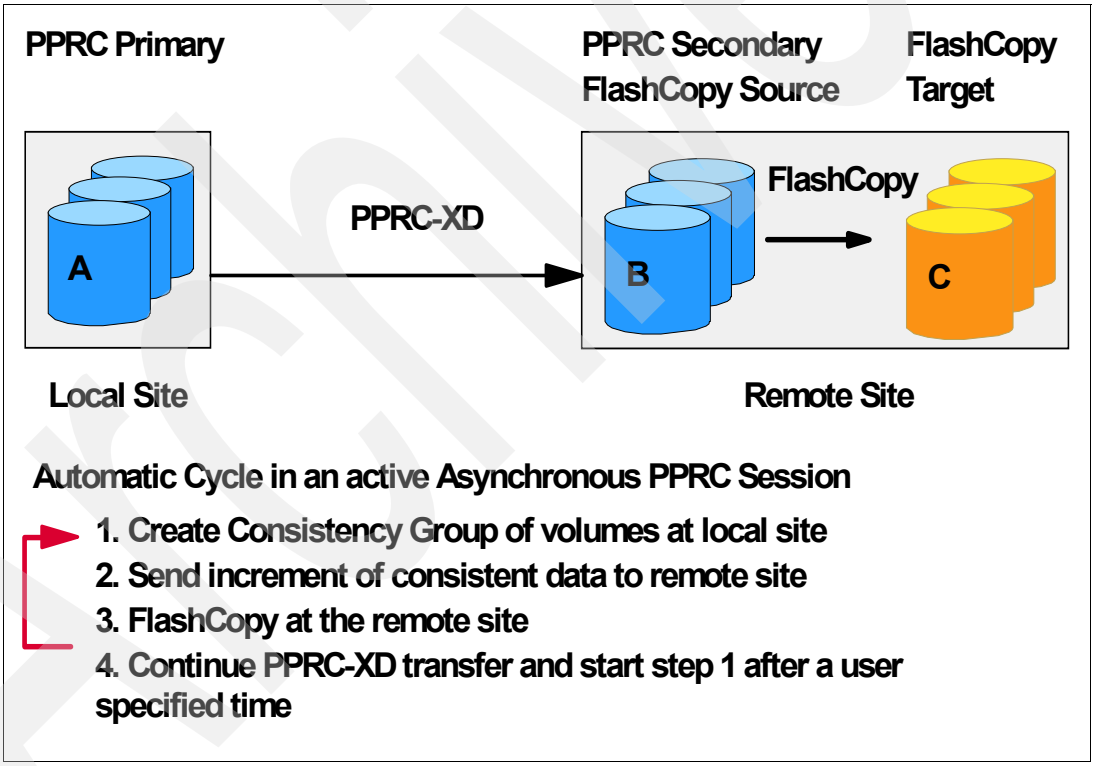


Figure 12-1 Asynchronous PPRC

A persistent FlashCopy with no background copy and active change recording continues immediately after the drain has finished. According to your Recovery Time Objectives (RTO), the available bandwidth, and your application workload, you can assign a Consistency Group Interval Timer value before the process starts again at step 1.

The currency of the data at the remote site can achieve typically 3 to 5 seconds (with a zero second Consistency Group Interval Timer value), but this depends on the workload and bandwidth available to the remote site.

Important: The consistent data at the remote site resides on the FlashCopy target volume and not on the PPRC secondary volume.

To enable this very fast cycle across multiple ESSs, a fast inband communication network is required. The configuration in Figure 12-2 depicts the necessary communication across LSSs and ESSs.

One of the local site ESSs is the Master ESS that communicates to Subordinate ESSs over Fiber Channel links to enable the creation and control of the Consistency Groups.

The volumes in the session can be open systems volumes as well as z/OS volumes. Consistency Groups are created for all volumes in the session at the same time, independently from the volume taxonomy.

.

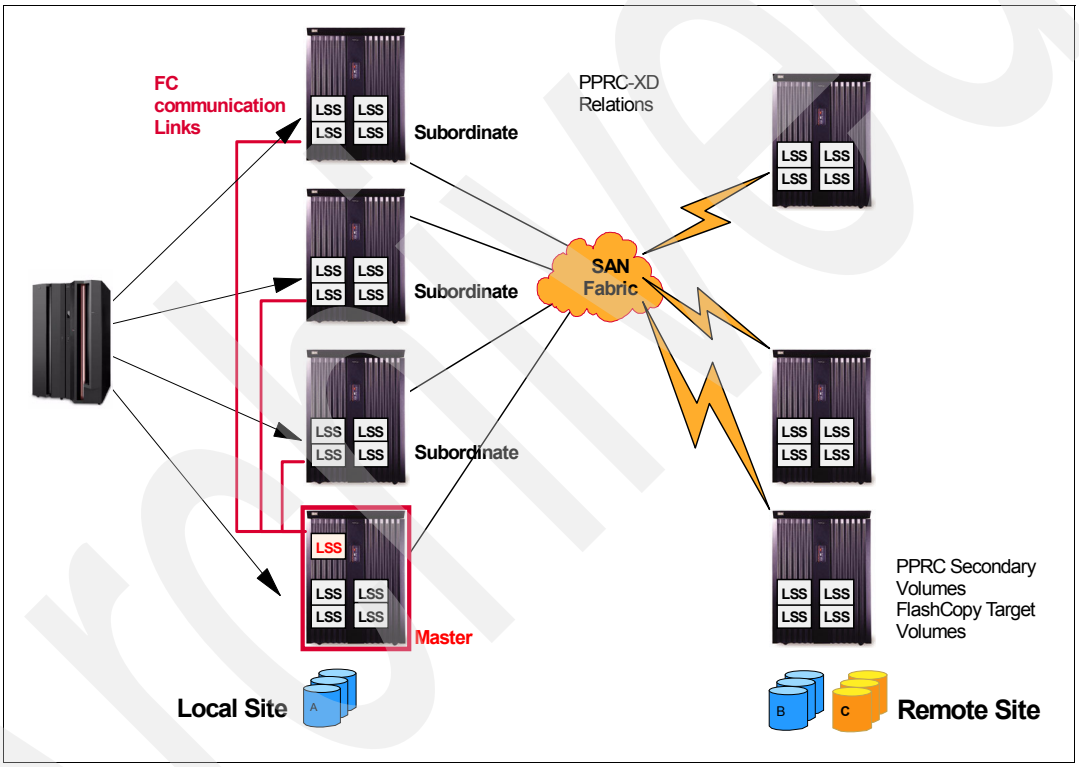


Figure 12-2 Example of an Asynchronous PPRC configuration

12.1.3 Restrictions

The architecture of this solution allows for growth, but the current restrictions are as follows:

- Only one Master can reside in an ESS and therefore only one session can be managed by one ESS.
- An LSS can be assigned only to one session.
- There can be a maximum of 3 subordinates with LIC 2.4.0 using Web Copy Services.
- An individual session consists some or all of the volumes in a Master and up to three primary ESS subordinates.

- A maximum of 8 ESSs (in any combination of primaries and secondaries) can participate in an Asynchronous PPRC environment using ICKDSF and ESS Copy Services Web User Interface (WUI).

Note: ICKDSF is architected to supports 16 Subordinates. Currently only one Master and up to seven Subordinate ESSs are supported for use. If you use ICKDSF *and* the Web User Interface, you should be aware that the ESS Specialist is only able to manage a total of eight ESSs, including primary and secondary ESSs in a single Copy Services Domain.

Important: The Asynchronous PPRC mechanism will only act upon volumes which have been added to a session, other PPRC volumes may still operate in any Copy Services mode and not be included in the definition of Consistency Groups.

12.2 Design the solution

In this chapter we present general disaster recovery solution design concepts adapted for an Asynchronous PPRC solution.

12.2.1 Define primary environment

As the first step in designing the solution, we must identify what data to mirror and indicate where it is located. After this process, we identify how many ESSs containing the production data have to be part of an Asynchronous PPRC session.

The following steps provide a logical guideline in defining your primary environment, and we assume that both ICKDFS and ESS Copy Services will be used to manage/control the solution.

Define Master and Subordinate ESS

Although the architecture of Asynchronous PPRC allows for growth, in the current implementation, a maximum of 8 ESSs (in any combination of primaries and secondaries) can participate in an Asynchronous PPRC environment.

Assuming that they are evenly distributed among primary and secondary site, it means that a maximum of 4 ESSs can act as Primary storage subsystems, where 1 will be the Master and the other 3 being the Subordinates.

You have to define which of your ESSs containing the production data to be mirrored will act as the Master ESS in the PPRC session. Remember that there can be only one Master ESS in a PPRC session and that it is not required for the Master ESS to have volumes participating in a PPRC session.

Once defined the Master ESS, all the other ESSs containing your production data act as Subordinate in the PPRC session.

Define Master/Subordinate communication paths

The Master ESS communicates with the Subordinate ESSs through a Fibre Channel communication path, which links one or more LSSs inside the Master ESS to at least one LSS in each Subordinate ESS.

The paths will be established with the same commands as PPRC data paths. They can be defined on separate physical links or on links combined with PPRC data paths, usually in a SAN.

Attention: For redundancy purposes, at least two paths from the Master to each Subordinate should be defined.

Define Master/Subordinate relationship

A Master is associated with one or more Subordinates using the TOPOLOGY keyword of the PPRCOPY STARTASYNCCOPY command.

This association is required to be explicit with the exception of the case where all the PPRCOPY primary volumes participating in Asynchronous PPRCOPY grouping are all on one ESS. What this means is that the same session ID can be defined to different ESSs, but unless they are associated with each other by the TOPOLOGY keyword, they will not participate together in the formation of Consistency Groups.

Define PPRC session and session ID

A session is a collection of volumes in an LSS, ESS or in multiple ESSs that are managed together to create consistent copies of data (Consistency Group). A session is identified with a session ID, which is a number from 1 to 255.

When the Asynchronous PPRC process is started, the Master, which is associated with a specific session ID, will communicate with all its Subordinates. Each of the Subordinates, which are also associated with a specific session ID, knows about all the sessions and their associated volumes on their ESS and will direct the processing dictated by the Master.

Only one session can be opened to a given LSS.

12.2.2 Define recovery environment

Once you have identified the primary environment, if primary and secondary sites are kept identical in the configuration (and you might do so to have the same performances in emergency situation), you have also defined the recovery environment.

Define the Target ESS

You define how many ESSs you need in the recovery site to handle both the Asynchronous PPRC environment and your whole production in case of an emergency.

The number of ESSs at the recovery site could be different from primary sites because of the following considerations:

- Space on ESS targets must be sufficient to contain PPRC secondary as well as FlashCopy target disks.
- Number of addresses (UCBs) must be sufficient to fit requirements for: PPRC, FlashCopy, PAV.
- Testing might require additional space/addresses. See 12.2.3, “Define testing environment requirements” on page 172 for more detail on this subject.
- A maximum of 8 ESSs (in any combination of primaries and secondaries) can participate in an Asynchronous PPRC environment. Since we already identified how many ESSs act as a primary, we also have identified how many can be secondary ESSs.

Define Primary/Secondary PPRC paths

Primary and Secondary ESS needs to be connected by PPRC paths (PPRC links in case of ESCON connections). We need to define how many paths we need in our configuration to sustain all the communication and data transfer for the Asynchronous PPRC session.

The Asynchronous PPRC Bandwidth Estimator tool is used by IBM ITS to support clients in the planning and sizing needed for Asynchronous PPRC. The customer requirements have to be analyzed to obtain the input data for the tool. This input data would include the required data currency, the distance to the remote location, the number of ESS and HBAs, the compressibility of the data, and the current workload characteristics of all the Open Systems and z/OS data that should be part of the Asynchronous PPRC session.

The tool calculates the expected bandwidth required for the solution and provides a recommendation for the number of required remote links for several available link types, such as OC-3, OC-12 T3, and many others.

You can contact your IBM representative for this bandwidth analysis.

Define PPRC-XD pairs and Source/Target FlashCopy relationships

PPRC and FlashCopy works on “pairs”, that is an association of two volumes. You need to identify these associations both for the PPRC-XD environment and for the FlashCopy environment on the secondary site only.

As a general suggestion, try to combine PPRC-XD pairs among the same (channel connection address) CCA/LSS in primary and secondary ESSs, and define FlashCopy source/target pairs among the same CCA in different LSSs at the secondary site.

As an example:

- The primary environment is disk ‘00’ in LSS ‘00’/ESS “A”.
- The secondary environment is ESS “B” with double space.

We define the PPRC-XD pair between ESS”A”/LSS0/CCA”00” and ESS”B”/LSS0/CCA”00”, while the FlashCopy source/target will be between ESS”B”/LSS0/CCA”00” and ESS”B”/LSS1/CCA”00”.

12.2.3 Define testing environment requirements

As we have anticipated, Asynchronous PPRC requires three sets of disks, one for primaries, one for secondaries, and one for FlashCopy targets. Valid consistent data is available only from the FlashCopy target set, but these disks are constantly updated by the Asynchronous PPRC process, and to be used by other systems, the Asynchronous PPRC process must be stopped.

If you are planning to use the FlashCopy target set for a periodic testing of the solution, you have to suspend the Asynchronous PPRC and leave it suspended while your testing is running. This means that you do not have any valid DR protection until your testing is completed and the Asynchronous PPRC processing is re-established.

Worse than that, once you have started using the FlashCopy target set, you have modified its contents, and the data you have on those disks at the end of the test might not be identical to the data you had at the beginning.

If you want to maintain your DR protection as well as doing a periodic testing of the solution, you will require additional space in the target ESSs to hold another FC target environment. In such an environment, you can define another set of ICKDSF commands to create your testing environment on another set of FlashCopy target (see Figure 12-3).

The PPRC target volumes now have two FlashCopy targets. Only the first relationship belongs to Asynchronous PPRC. The second relationship is not persistent, it cannot have change recording on, but it can be used by other systems for testing purposes.

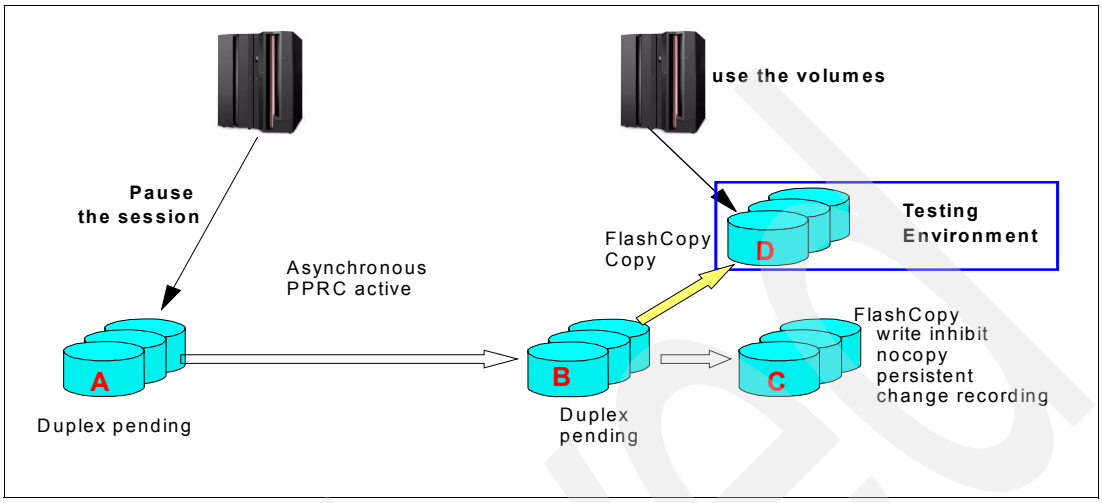


Figure 12-3 Create a set of test volumes at the remote site

12.2.4 Select the interface to PPRC commands

At the time of General Availability of LIC V2.4.0 support, only the Copy Services Web User Interface, the Command Line Interface and ICKDSF can be used to manage Asynchronous PPRC. It is planned to enable the Multiple Device Manager, TSO, ANTQRST Macro, and GDPS for managing Asynchronous PPRC. REXX based scripted tools for ICKDSF are also planned to be available with Asynchronous PPRC.

For Asynchronous PPRC management on z/OS, VM, and VSE operating systems, ICKDSF is currently the only zSeries based interface.

There are different command names for ICKDSF and for the Web User Interface that are summarized in Table 12-1.

Table 12-1 Web Interface and ICKDSF commands

Web Interface		**ICKDSF**	
Manage Session	Open session Close session	DEFINESESSION	Open Close
Start/Resume session	Start Resume	STARTSESSION	Start Modify
Manage session members	Add Remove	POPULATESESSION	Join Remove
Pause/Terminate session	Pause Terminate	TERMINATESESSION	Pause Terminate

In this redbook we focus on the management with ICKDSF. The detailed description of the ICKDSF commands can be found the *Device Support Facility Users Guide and Reference,* GC35-0033.

12.2.5 Define session management commands

In this section we provide definitions of the various session management commands.

Open/close a session

A session must be opened using a specific session ID and a session must be opened using the same session ID for each LSS that participates in the Asynchronous PPRC session. This step must be done prior to adding volumes to a session. Before closing a session you need to remove the volumes. Closing a session in an LSS deletes all of these definitions in that LSS. Close a session when you no longer want the volumes in a particular LSS to participate in Asynchronous PPRC.

Add/remove volumes

To add volumes to the session you have to specify all PPRC primary volumes. The volumes are set to the *join pending* state until the first PPRC synchronization pass has occurred. They can be dynamically removed and added to a running session.

Start/resume a session

The session will be started by addressing any LSS in the ESS that you wish to become the Master. All subsequent Asynchronous PPRC commands must be addressed to this same LSS. This LSS must have PPRC communication to all Subordinate ESSs. A Start command starts the Asynchronous PPRC mechanism for the first time or resumes the Asynchronous PPRC mechanism after a Pause command has been issued.

> **Note:** Once the initial Asynchronous PPRC configuration is created and for as long as it exists, you can start or resume a session for the Master only on the same LSS where the initial session was created.

At the start of the session, you can also modify its three timer values:

- *Consistency Group Interval Timer* (in seconds, default 0, maximum 64K sec), determines the time from last successful FlashCopy to the start of the creation of the next Consistency Group.
- *Maximum Coordination Timer* (in milliseconds, default 50 ms, maximum 64K ms), determines the maximum time to coordinate the ESSs at the local site in order to form a Consistency Group.
- *Maximum Consistency Group Drain Time* (in seconds, default 30 sec, maximum 64K sec), determines the maximum time to transfer the consistent data to the remote site.

Pause/terminate a session

This command can be used to terminate or pause the Asynchronous PPRC session. Both commands stop the Asynchronous PPRC session. Pause will ensure the current Consistency Group forms before stopping the Asynchronous PPRC session. Terminate will interrupt the formation of a Consistency Group if one is running and clean up by reverting to the previous Consistency Group. If no group is being formed it will terminate immediately. Use Terminate when you need to change the session topology such as defining a new Master and adding or removing a Subordinate. The command removes Master/Subordinate relationship information. Force Terminate will terminate immediately and will interrupt the creation of the Consistency Group without recovery if one is being processed.

> **Note:** Force should only be done when instructed to do so by IBM support. It requires that the Consistency Group be manually verified before the data can be used.

Query a session or volumes

There are several different queries available for the Web User Interface and ICKDSF. The ICKDSF queries are described in more detail in 12.5, “Monitor Asynchronous PPRC” on page 189

12.3 Implement the solution

In this section we provide you with guidelines on how to implement an Asynchronous PPRC solution for the environment depicted in Figure 12-4.

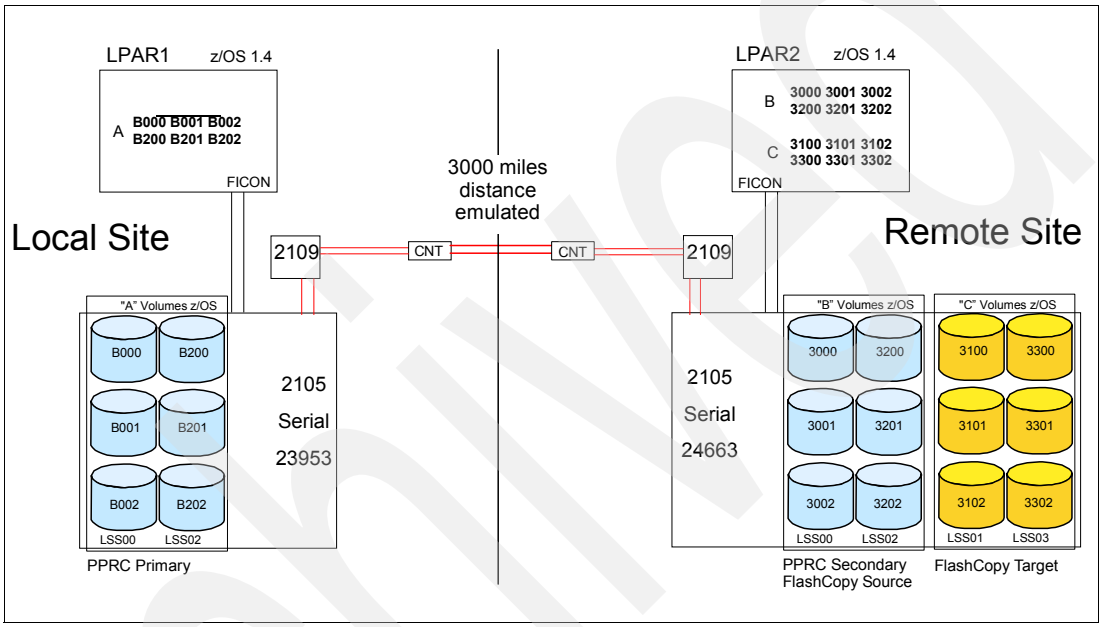


Figure 12-4 Implementation environment

Three volumes in each of two logical subsystems are used in the primary ESS. We use one ESS pair only, and therefore the primary ESS is also the Master and no Subordinate is used. The distance of 3000 miles is generated by a device with the corresponding signal delay in the line between the Channel Extenders.

12.3.1 Establish Async-PPRC data paths

In our configuration no Subordinate is used, so we do not need to define data paths among Master and each Subordinate.

In configurations where Subordinates are present, these data paths must be established to enable Master to Subordinate communication. To define a data path, use the same command for establishing a PPRC path.

Note: ICKDSF is architected to supports 16 Subordinates. Currently only one Master and up to seven Subordinate ESSs are supported for use. If you use ICKDSF *and* the Web User Interface, you should be aware that ESS Specialist is only able to manage a total of eight ESSs, including primary and secondary ESSs in a single Copy Services Domain.

PPRC Paths between the Master ESS and all subordinate ESSs have to be established prior to the start of the session. The paths can be established between any LSS in the Master and Subordinate LSS.

> **Important:** For redundancy purposes, at least two paths from the Master to each Subordinate should be defined.

12.3.2 Establish PPRC paths to the remote site

Establish two Fibre Channel PPRC Paths from the local site to the remote site ESS. See Example 12-1 for the ICKDSF commands. These commands have to be used for every LSS.

Example 12-1 Establish PPRC paths

```
PPRCOPY ESTPATH                    -
         DDNAME(VLB000)              -
         PRIMARY(X'1340',23953)      -
         SECONDARY(X'1070',24663)    -
         LSS(X'00',X'00')            -
         CGROUP(YES)                 -
         WWNN(5005076300C09B29,5005076300C09DEF) -
         FCPPATHS(X'002C002C',X'008C008C')

  PPRCOPY ESTPATH                    -
         DDNAME(VLB200)              -
         PRIMARY(X'1342',23953)      -
         SECONDARY(X'1072',24663)    -
         LSS(X'02',X'02')            -
         CGROUP(YES)                 -
         WWNN(5005076300C09B29,5005076300C09DEF) -
         FCPPATHS(X'002C002C',X'008C008C')
```

12.3.3 Establish PPRC-XD pairs

Only existing PPRC primary devices can be added to an Asynchronous PPRC session, therefore we need to establish six PPRC Pairs in Extended Distance mode. Example 12-2 shows the ICKDSF commands.

Example 12-2 Establish PPRC-XD pairs

```
  PPRCOPY ESTPAIR                       -
         DDNAME(VLB001)                 -
         PRIMARY(X'1340',23953,X'00')   -
         SECONDARY(X'1070',24663,X'00') -
         LSS(X'00',X'00')               -
         OPTION(XD)                     -
         MODE(COPY)

  PPRCOPY ESTPAIR                       -
         DDNAME(VLB001)                 -
         PRIMARY(X'1340',23953,X'01')   -
         SECONDARY(X'1070',24663,X'01') -
         LSS(X'00',X'00')               -
         OPTION(XD)                     -
         MODE(COPY)

  PPRCOPY ESTPAIR                       -
         DDNAME(VLB002)                 -
```

```
         PRIMARY(X'1340',23953,X'02')    -
         SECONDARY(X'1070',24663,X'02') -
         LSS(X'00',X'00')                -
         OPTION(XD)                      -
         MODE(COPY)

PPRCOPY  ESTPAIR                         -
         DDNAME(VLB200)                  -
         PRIMARY(X'1342',23953,X'00')    -
         SECONDARY(X'1072',24663,X'00') -
         LSS(X'02',X'02')                -
         OPTION(XD)                      -
         MODE(COPY)

PPRCOPY  ESTPAIR                         -
         DDNAME(VLB201)                  -
         PRIMARY(X'1342',23953,X'01')    -
         SECONDARY(X'1072',24663,X'01') -
         LSS(X'02',X'02')                -
         OPTION(XD)                      -
         MODE(COPY)

PPRCOPY  ESTPAIR                         -
         DDNAME(VLB202)                  -
         PRIMARY(X'1342',23953,X'02')    -
         SECONDARY(X'1072',24663,X'02') -
         LSS(X'02',X'02')                -
         OPTION(XD)                      -
         MODE(COPY)
```

Figure 12-5 shows the result in the Web User Interface.

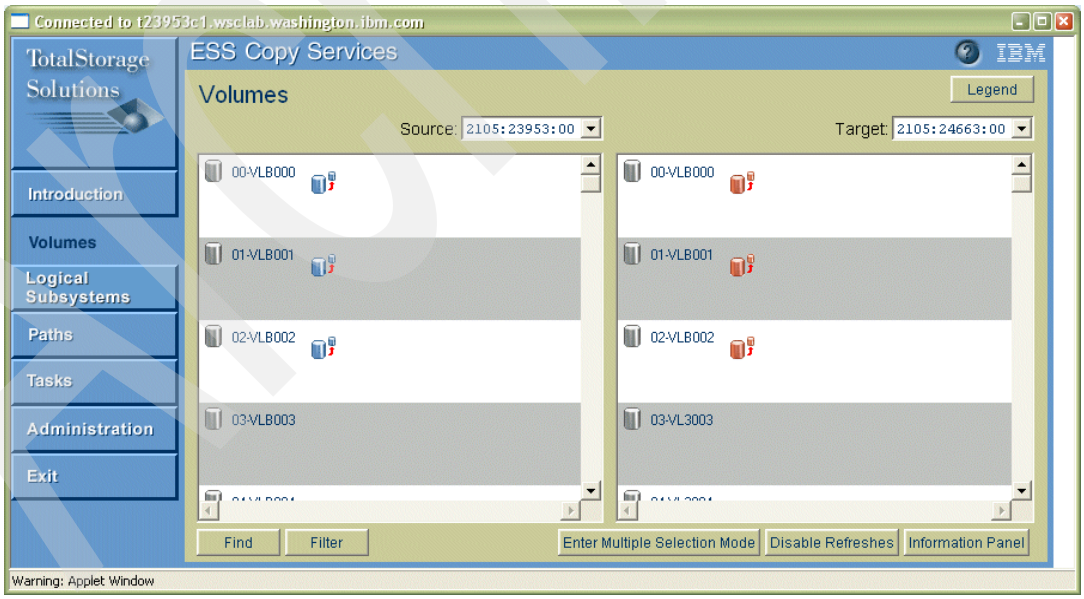


Figure 12-5 PPRC-XD pairs

12.3.4 Establish FlashCopy pairs at the remote site

At the remote site, we have to define the last component of an Asynchronous PPRC, which are the FlashCopy Target devices. FlashCopy pairs will be established using Inband FlashCopy at the remote site with the options: *nocopy*, *change recording*, *inhibit target writes* and *persistent*, which can be addressed to the PPRC secondary volumes. See the ICKDSF commands in Example 12-3. The SOURCEVOL Parameter is used to address the Inband FlashCopy source volume on a PPRC secondary ESS.

Example 12-3 Establish FlashCopy Pairs

```
FLASH   ESTAB                                -
        DDNAME(VLB000)                       -
        SOURCEVOL(X'00',X'00',X'1070',24663) -
        TARGETVOL(X'01',X'00',3100)          -
        INHIBITTARGETWRITES(YES)             -
        CHANGERECORDING(YES)                 -
        MODE(NOCOPY)

FLASH   ESTAB                                -
        DDNAME(VLB001)                       -
        SOURCEVOL(X'00',X'01',X'1070',24663) -
        TARGETVOL(X'01',X'01',3101)          -
        INHIBITTARGETWRITES(YES)             -
        CHANGERECORDING(YES)                 -
        MODE(NOCOPY)

FLASH   ESTAB                                -
        DDNAME(VLB002)                       -
        SOURCEVOL(X'00',X'02',X'1070',24663) -
        TARGETVOL(X'01',X'02',3102)          -
        INHIBITTARGETWRITES(YES)             -
        CHANGERECORDING(YES)                 -
        MODE(NOCOPY)

FLASH   ESTAB                                -
        DDNAME(VLB200)                       -
        SOURCEVOL(X'02',X'00',X'1072',24663) -
        TARGETVOL(X'03',X'00',3300)          -
        INHIBITTARGETWRITES(YES)             -
        CHANGERECORDING(YES)                 -
        MODE(NOCOPY)

FLASH   ESTAB                                -
        DDNAME(VLB201)                       -
        SOURCEVOL(X'02',X'01',X'1072',24663) -
        TARGETVOL(X'03',X'01',3301)          -
        INHIBITTARGETWRITES(YES)             -
        CHANGERECORDING(YES)                 -
        MODE(NOCOPY)

FLASH   ESTAB                                -
        DDNAME(VLB202)                       -
        SOURCEVOL(X'02',X'02',X'1072',24663) -
        TARGETVOL(X'03',X'02',3302)          -
        INHIBITTARGETWRITES(YES)             -
        CHANGERECORDING(YES)                 -
        MODE(NOCOPY)
```

Figure 12-6 shows the result in the WUI. You can see the icon indicating that *change recording* is set on at the FlashCopy Source.

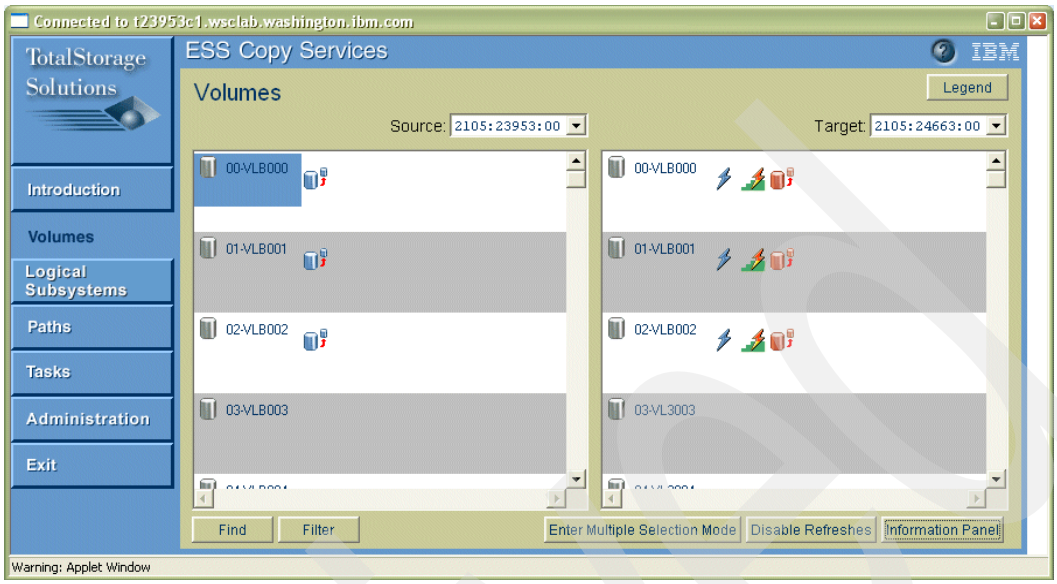


Figure 12-6 Establish Inband FlashCopy - nocopy, changerecording, inhibit write to target

You can use the command shown in Example 12-4 to create a second FlashCopy target volume for the same source volume to be used for disaster recovery testing purposes, as described in 12.2.3, “Define testing environment requirements” on page 172.

Example 12-4 Establish FlashCopy Pair

```
FLASH   ESTABLISH                                  -
        DDNAME(VLB000)                             -
        SOURCEVOL(X'00',X'00',X'1070',24663)       -
        TARGETVOL(X'01',X'00',3100)                -
        INHIBITTARGETWRITES(YES)                   -
        CHANGERECORDING(YES)
```

The PPRC target volumes now have two FlashCopy targets. Only the first relationship belongs to Asynchronous PPRC. The second relationship is not persistent and cannot have change recording on. It can only be used for testing purposes or for making a tape copy of your recovery DASD environment.

12.3.5 Define the Asynchronous PPRC session

This has to be done for a group of PPRC primary logical subsystems that should be included in the session regardless of the location in the Master or Subordinate ESSs. A session number from 1 to 255 has to be assigned, and the included LSSs can be addressed by any one of its volumes. Example 12-5 defines a session with ID 50 for two LSSs.

Example 12-5 Defining and opening a session

```
PPRC    DEFINESESSION          -
        DDNAME(VLB000)         -
        SESSIONNO(50)          -
        OPEN
```

```
PPRC    DEFINESESSION              -
        DDNAME(VLB200)             -
        SESSIONNO(50)              -
        OPEN
```

The determination of the Master LSS is done by the STARTASYNCCOPY command, see.12.3.7, “Start the session” on page 181. Currently only one session per ESS, and therefore one Master per ESS, is allowed.

The status of the session can be queried by the QUERY ASYNCCOPY command, see 12.5.1, “Determine the status of Asynchronous PPRC” on page 189.

12.3.6 Add volumes to the session

All the primary volumes of the previously defined LSSs, that should be part of the Asynchronous PPRC session, have to be inserted into the session.

One or more ranges of CCA can be specified for each LSS that belongs to a session. Volumes can be joined or removed dynamically any time a session exists. Both individual volumes or volume ranges can be specified in the command.

Only PPRC primary volumes can join a session.

The volumes remain in *joined pending* state as long as they have not reached the first full copy status during the PPRC-XD initial copy. When all volumes of the session have reached this state, the Asynchronous PPRC process starts creating Consistency Groups.

If new volumes join the session after the STARTASYNCCOPY command, the creation of Consistency Groups continues and the new volumes join the session when they have completed the first pass.

Example 12-6 shows adding six primary volumes (3 volumes per LSS) to the session.

Example 12-6 Populating the Asynchronous PPRC session with volumes

```
PPRC    POPULATESESSION            -
        DDNAME(VLB000)             -
        SESSIONNO(50)              -
        VOLCOUNT(1)                -
        RANGE(YES)                 -
        RVOLLIST(X'00',X'02')      -
        JOIN

PPRC    POPULATESESSION            -
        DDNAME(VLB200)             -
        SESSIONNO(50)              -
        VOLCOUNT(1)                -
        RANGE(YES)                 -
        RVOLLIST(X'00',X'02')      -
        JOIN
```

The progress of volumes joining a session can be monitored by the QUERY SESSIONDEVICES command. See 12.5.2, “Query the status of the volumes in the session” on page 191.

12.3.7 Start the session

The command in Example 12-7 starts the Asynchronous PPRC session. The session number is 50, and you can see the three timers for Asynchronous PPRC processing:

- The Maximum Coordination Time, which is set to 50 ms
- The Maximum Drain Time, which is set to is 4 minutes
- The Consistency Group Interval Timer value, which is set to 10 seconds

These timers can be customized to match Asynchronous PPRC processing with the RPO required by your disaster recovery solution.

Example 12-7 STARTASYNCCOPY - command syntax

```
PPRCOPY STARTASYNCCOPY          -
        DDNAME(VLB000)          -
        SESSIONNO(50)           -
        MAXCOORDTIME(50)        -
        MAXDRAINTIME(240)       -
        CGINTERVALTIME(10)      -
        START
```

Figure 12-7 shows the new icons for the joining volume.

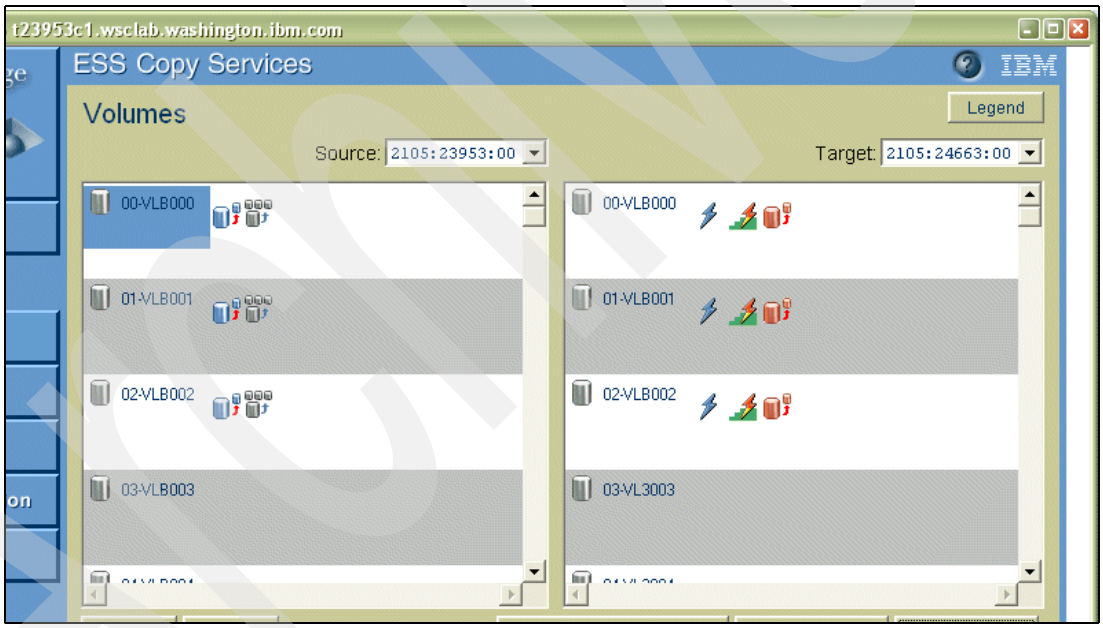


Figure 12-7 Asynchronous PPRC session started, primary volumes join pending

Figure 12-8 shows details of the primary volume 00.

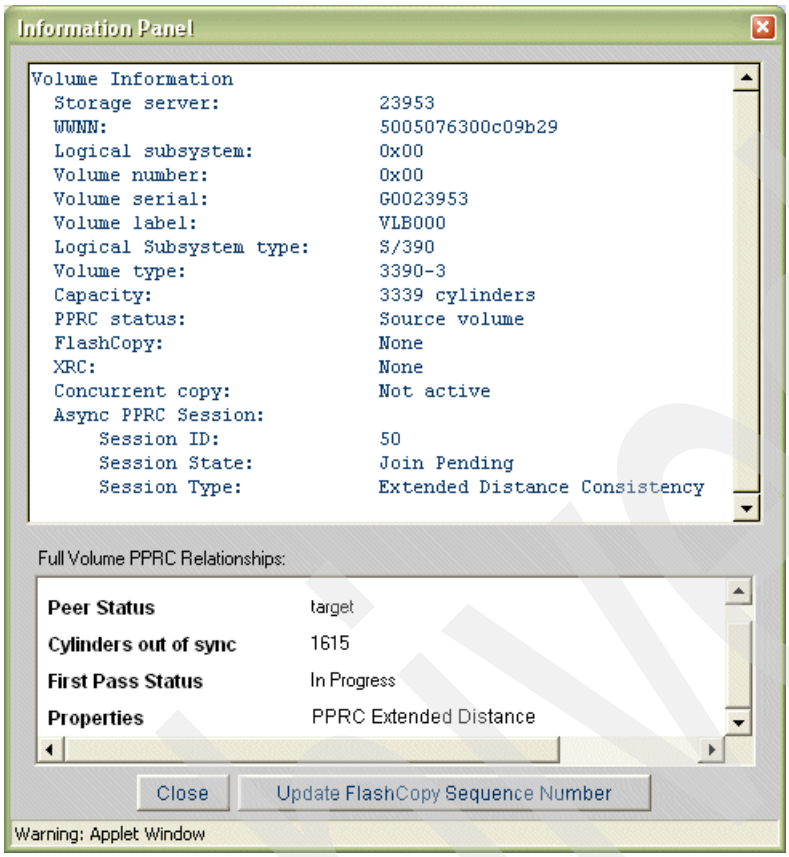


Figure 12-8 PPRC-XD primary volume first pass in progress, volume status: join pending

Figure 12-9 shows the detailed status of volume 00 in LSS 00 of the target ESS 24663. You can see details of the PPRC and FlashCopy relationships and that there is no Consistency Group number yet assigned.

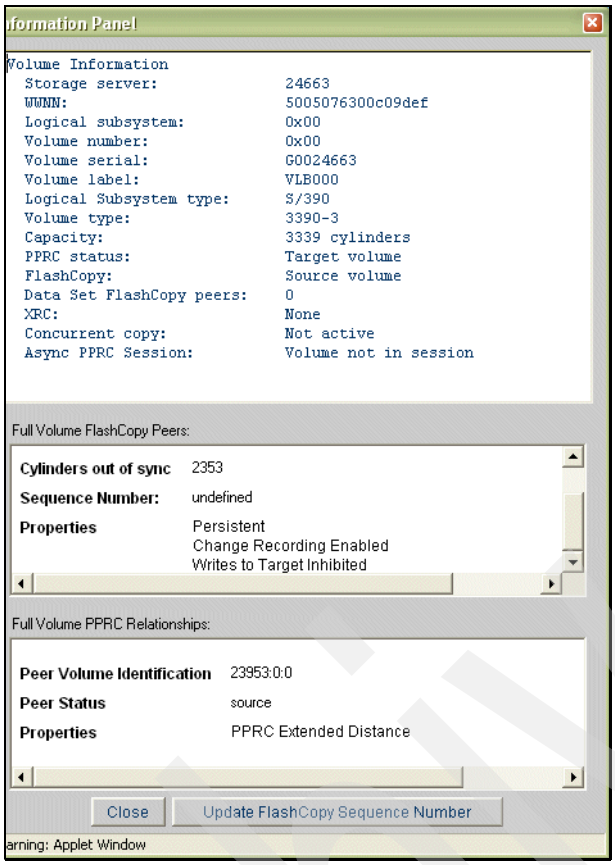


Figure 12-9 Secondary volume status

Figure 12-10 shows the LSS view with Asynchronous PPRC. LSS 00 is the Master LSS that was addressed to start the session and a special new icon is shown that was introduced to identify this LSS. All subsequent session commands should be communicated to this LSS.

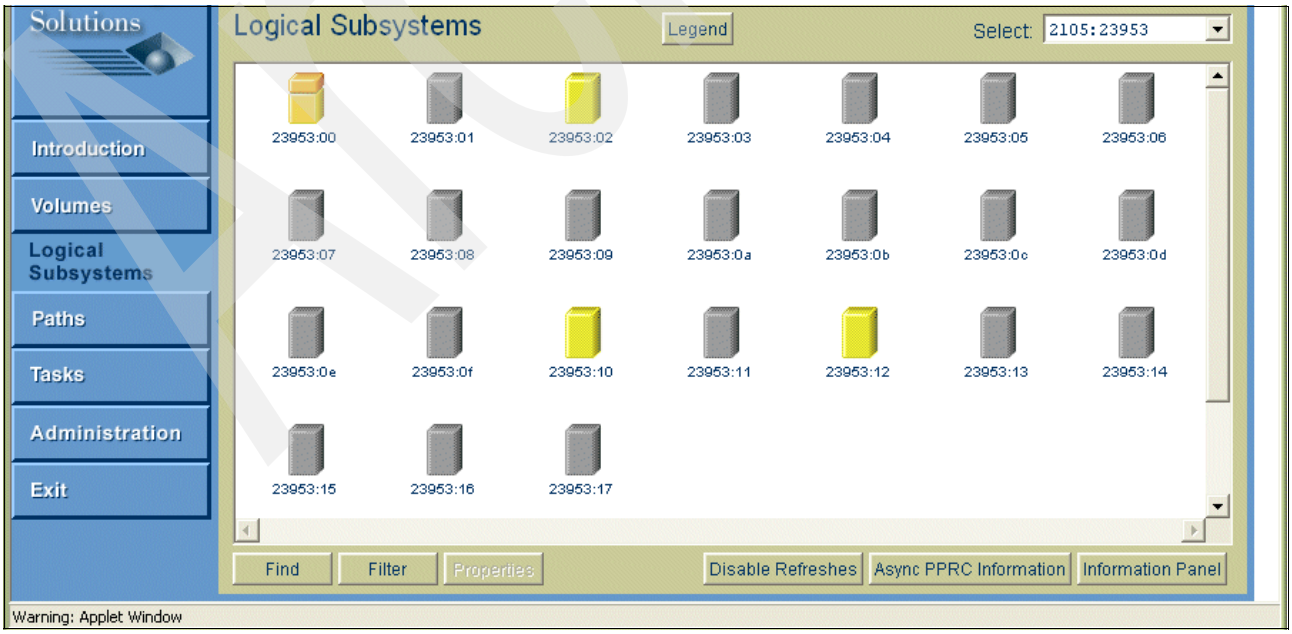


Figure 12-10 Logical Subsystem view of Asynchronous PPRC

If you click the button **Asynchronous PPRC Information**, you get the information shown in Figure 12-11. The rising value of *Successful Consistency Group Count* shows that the creation of Consistency Groups is running.

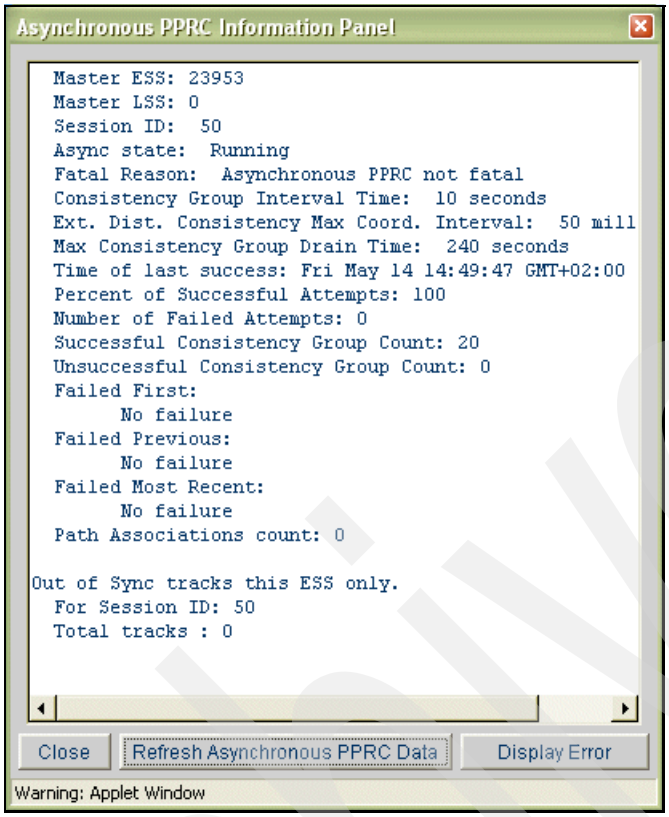


Figure 12-11 Status of the Asynchronous PPRC session

Figure 12-12 shows the progress of joining volumes. Volumes 00 and 01 have already joined the session, but volume 02 has not.

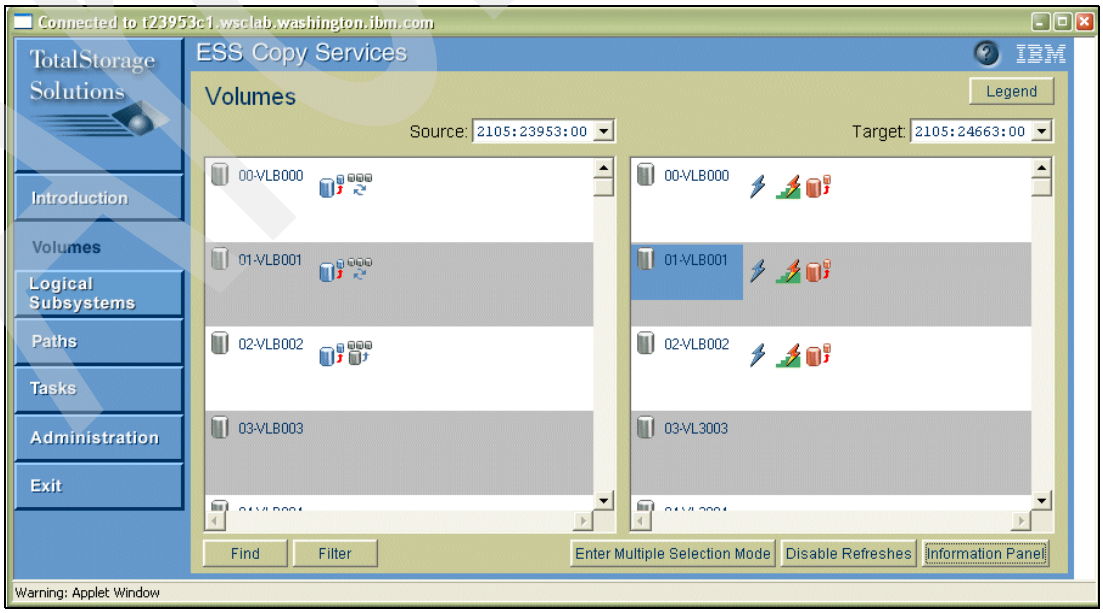


Figure 12-12 Primary volumes 00 and 01 joined the session, volume 02 is still join pending

Figure 12-13 shows a detailed volume query of an already joined volume 00.

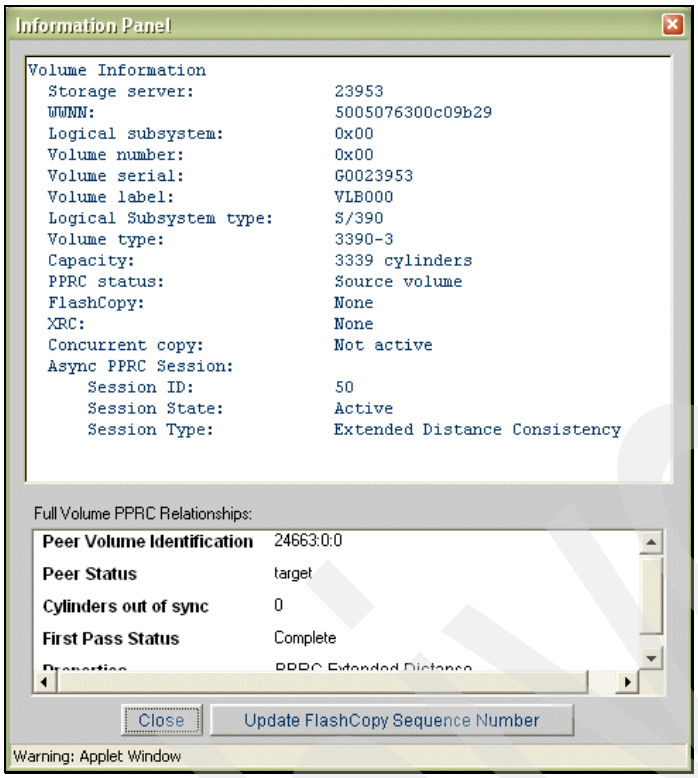


Figure 12-13 PPRC-XD primary volume with first pass complete and part of the session

Once all the volumes have joined the session, Asynchronous PPRC periodically creates Consistency Groups and increments the Consistency Group number. You can check this progress both at the PPRC secondary volumes, as well as the FlashCopy target volumes.

This is shown in Figure 12-14 for the secondary volumes.

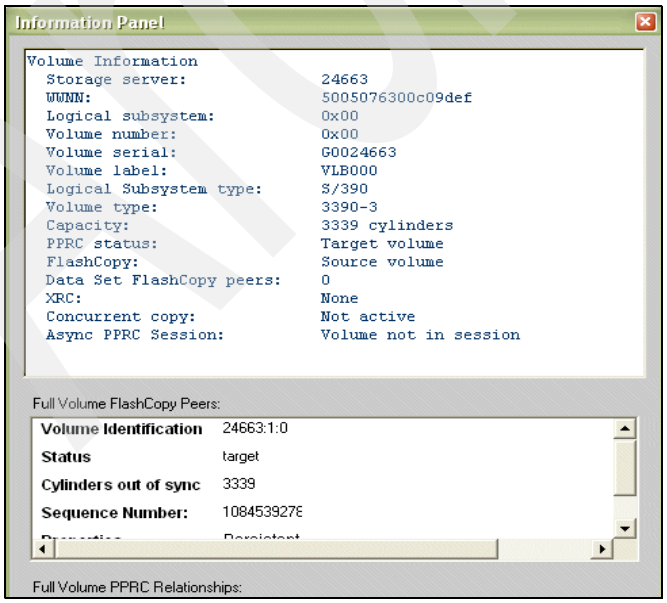


Figure 12-14 PPRC target volume shows the FlashCopy sequence number

And, it is shown in Figure 12-15 for the target volumes.

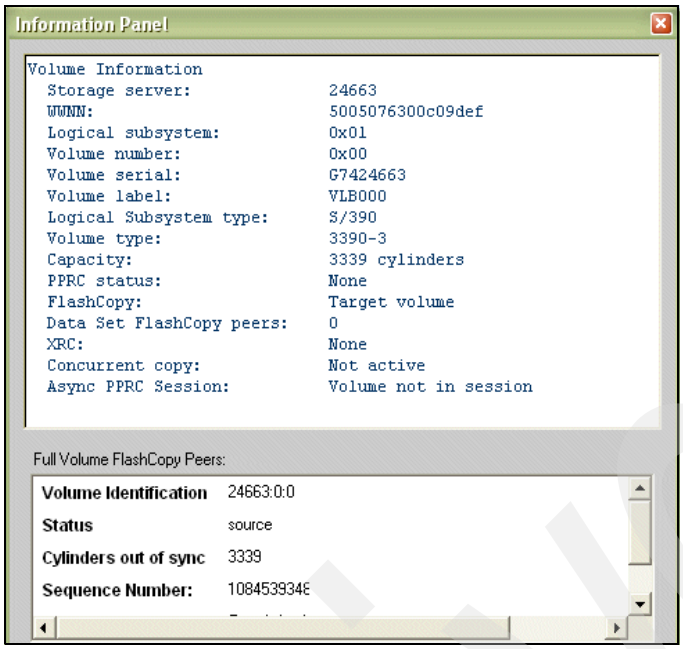


Figure 12-15 FlashCopy target volume with Consistency Group number

Finally, you now can watch the session on the Asynchronous PPRC information panel shown in Figure 12-16.

If there are failures, or if the Consistency Groups cannot be created anymore or creation is delayed, you can see the number of failed attempts in the information panel. The number of out-of-sync tracks for the selected ESS is displayed at the bottom, and you can check if this is periodically increasing or decreasing as Consistency Groups are created and drained to the remote site. To update the view, you have to click the **Refresh Asynchronous PPRC Data** button.

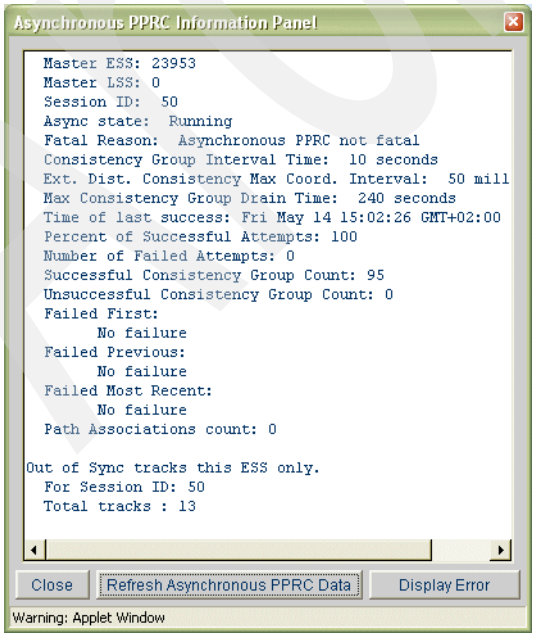


Figure 12-16 Asynchronous PPRC information panel, all volumes in session

12.4 Run the solution

In this section we describe how to manage and test the session.

12.4.1 Add/remove volumes to/from the session

Only PPRC primary volumes can join a session, thus to add a volume to a session, you must first define PPRC-XD pairs.

If you add a primary PPRC volume for which a PPRC-XD initial copy is still running, that volume remains in *joined pending* state as long as it has not reached the first full copy status. However, the creation of Consistency Groups continues for the other volumes and the new volume joins the session when its *First Pass Status* is “Complete”.

To add a volume in a session, you can use PPRC POPULATESESSION with the JOIN parameter. See Example 12-6 on page 180 for this command.

To remove a volume from a session, you can use PPRC POPULATESESSION with the REMOVE parameter. Example 12-8 shows removing two primary volumes from the session.

Example 12-8 Removing volumes from a session

```
PPRC    POPULATESESSION          -
        DDNAME(VLB000)           -
        SESSIONNO(50)            -
        VOLCOUNT(1)              -
        RANGE(NO)               -
        IVOLLIST(X'00')         -
        REMOVE

PPRC    POPULATESESSION          -
        DDNAME(VLB200)           -
        SESSIONNO(50)            -
        VOLCOUNT(1)              -
        RANGE(NO)                -
        IVOLLIST(X'00')          -
        REMOVE
```

12.4.2 Modify the session

The *Maximum Coordination Interval Timer*, *Maximum Consistency Group Drain Time* and *Consistency Group Interval Timer* values can be changed using the MODIFY parameter on the STARTASYNCCOPY command if the session is already started. Before changing the timers, the session has to be paused by the TERMASYNCCOPY PAUSE.

The session number and the ESS relationships in the session are unchangeable as long as the session is started.

You can dynamically add LSSs and volumes to the session that belongs to the ESSs which are already in the Asynchronous PPRC session. If you need to include a new ESS, you have to terminate the session first. See Example 12-9.

All the STARTASYNCCOPY commands to modify the session should be issued to the Master LSS (the LSS to which you send the first STARTASYNCCOPY command) as long as you did not terminate the session with the TERMASYNCOPY command.

Example 12-9 Modify a session

```
PPRCOPY STARTASYNCCOPY            -
        DDNAME(VLB000)            -
        SESSIONNO(50)             -
        MAXCOORDTIME(50)          -
        MAXDRAINTIME(30)         -
        CGINTERVALTIME(10)        -
        MODIFY
```

The status of the session can be monitored by the QUERY ASYNC command.

12.4.3 Pause/terminate the session

The session can be terminated or paused.

A paused session completes the last Consistency Group and keeps the configuration of the Master and Subordinate relations. Example 12-10 shows how to pause a session.

Example 12-10 Pause a session

```
PPRCOPY TERMASYNCCOPY             -
        DDNAME(VLB000)            -
        SESSIONNO(50)             -
        PAUSE
```

A paused session can be restarted with the STARTASYNCCOPY command.

If you terminate the session with TERMINATE option, the creation of Consistency Groups stops immediately. Example 12-11 shows how to terminate a session.

Example 12-11 terminate a session

```
PPRCOPY TERMASYNCCOPY             -
        DDNAME(VLB000)            -
        SESSIONNO(50)             -
        TERMINATE
```

12.4.4 Test the solution

For disaster recovery testing, application testing or application development, you need a consistent and accessible copy of your production data at the remote site.

The following steps describe a procedure to create this set of data with minimal impact at the primary site, based on what we described in 12.2.3, “Define testing environment requirements” on page 172 and the configuration depicted in Figure 12-3 on page 173:

1. Pause the Asynchronous PPRC session with the PPRCOPY TERMASYNCCOPY command.
2. Issue the FlashCopy inband command with mode COPY to the PPRC primary volumes by addressing a second set of FlashCopy target volumes, D, at the remote site. Refer to Figure 12-3 on page 173.
3. The remote D volumes can now be used at the remote site.

4. Resume the paused Asynchronous PPRC session using the PPRCOPY STARTASYCCOPY command.

> **Attention:** Referring to Figure 12-3 on page 173, if the FlashCopy to the D volumes is persistent, then before any recovery can be done from the C volumes in a disaster recovery situation, the D volume FlashCopies must be withdrawn.

To withdraw a persistent flashcopy relationship, you can use the command in Example 12-12.

Example 12-12 Withdraw a FlashCopy Pair

```
FLASH   WITHDRAW                                -
          DDNAME(VLB000)                        -
          SOURCEVOL(X'00',X'00',X'1070',24663) -
          TARGETVOL(X'01',X'00',3100)
```

12.5 Monitor Asynchronous PPRC

In this section we describe the ICKDSF Query commands. You can use automation procedures to check for special conditions and highlight important messages to the operator if failure conditions are detected.

12.5.1 Determine the status of Asynchronous PPRC

The QUERY ASYNCCOPY command provides the following important information on ESS box level about an Asynchronous PPRC session:

- Time of the last Consistency Group formed
- Session status
- Information about the three timers
- Master and Subordinate relations
- Error Conditions

If z/OS LSSs as well as Open Systems LSSs are members of the session you can see the session information that is valid for both of them in the report. See the syntax of the command in Example 12-13.

Example 12-13 Query the status of Asynchronous PPRC - command syntax

```
PPRC QUERY DDNAME(VLB000) ASYNCCOPY
```

Example 12-14 shows the messages resulting from a QUERY ASYNCCOPY command.

Example 12-14 Messages about the status of Asynchronous PPRC

```
ASYNCHRONOUS PPRC CONFIGURATION INFORMATION
MASTER CU SERIAL NUMBER/LSS                :  23953/00

NUMBER OF MASTERS IN THIS ESS BOX          :  1
NUMBER OF OTHER MASTERS TO THIS SUBORDINATE:  0
SESSION INFORMATION
CURRENT TIME = 2004.04.27.20:23:57
+-------------------------------------------------------------------------------------------------------------------+
!    !     !       !       !       !          ! TOTAL #  !      !           ! TOTAL #  !                            !
!    !     !       !       !       !          ! GOOD CGS !      ! # FAILED  ! UNSUCCFL !                            !
!    !     !       !  MAX  !  MAX  !          !  FORMED  !      ! TRIES TO  ! TRIES TO !   THIS MASTER TO OTHER    !
!    !     !  CG   ! COORD !  CG   !  TIME    !  SINCE   ! %AGE ! FORM CG   ! FORM CG  !    SUBORDINATES DATA      !
!    !     ! INTVL ! INTVL ! DRAIN !  LAST    ! LAST IML ! GOOD ! SINCE     !  SINCE   !---------------------------!
!SESS!     ! TIME  ! TIME  ! TIME  ! GOOD CG  !    OR    ! CG   ! LAST GOOD ! LAST IML !MSTR/LSS!SUB/LSS!  SUBORD  !
! NO !STATE! (SEC) ! (MSEC)! (SEC) ! FORMED   ! FAILOVER ! FRMD ! CG FORMED !OR FAILOVR!SSID    !SSID   !   CUSN   !
```

```
!----+-----+-------+-------+-------+----------+----------+------+-----------+----------+--------+-------+----------!
! 50 !RUN  !   10  !   50  !  240  ! 04/27/04 !    108   !  97  !    00     !     3    ! NO SUBORDINATES DEFINED   !
!    !     !       !       !       ! 20:20:02 !          !      !           !          !                           !
+-------------------------------------------------------------------------------------------------------------------+
LEGEND
  SESS NO
    UA   : UNABLE TO RETRIEVE INFORMATION, DATA STRUCTURES UNAVAILABLE

  STATE
ICKDSF - MVS/ESA    DEVICE SUPPORT FACILITIES 17.0                TIME: 16:24:59        04/27/04     PAGE   3
    RUN  : SESSION IS RUNNING
    PAUSE: SESSION IS PAUSED
    P PND: SESSION IS PAUSE PENDING
    FTL01: REVERT FLASHCOPY FAILED
    FTL02: ASYNC PPRC STRUCTURES INACCESSIBLE OR IN FAILED STATE
    FTL03: ASYNC PPRC CONSISTENCY CHECK OF SEQUENCE NUMBERS FAILED
    FTL04: ASYNC PPRC INVALID SESSION ID DETECTED
    FTL05: CONSISTENCY GROUP CORRUPTED
    FTL06: TIMEOUT DURING CONSISTENCY CHECK
    FTL07: COMMUNICATION FAILURE DURING CONSISTENCY CHECK
    FTL08: UNABLE TO COMPLETE CONSISTENCY CHECK
    FTL09: BUSY CONDITION PREVENTS CONSISTENCY CHECK
    FTL0A: REVERT FLASHCOPY FAILED DUE TO TIMEOUT

  TIME LAST GOOD CG FORMED, NUMBER FAILED TRIES TO FORM CG, %AGE TRIES FORMED GOOD CG
    HH:MM:SS: INDICATED TIME
    00      : CONSISTENCY GROUP NEVER FORMED (TIME LAST GOOD CG FORMED, ONLY)
    FL      : INFORMATION UNOBTAINABLE DUE TO FAILOVER OR IML OF ESS

CONSISTENCY GROUP FAILURES REPORT
+-------------------------------------------------------+
!      !         !            !     !        ! MASTER !
! SESS !  WHICH  !            !     ! ERROR  ! STATE  !
!  NO  ! FAILURE !   CUSN     ! LSS ! REASON ! CODE   !
!------+---------+------------+-----+--------+--------!
!  50  !  FIRST  !   23953    !  02 !    FC9 !    04  !
!      !---------+------------+-----+--------+--------!
!      !  PREV   !   23953    !FO/IM!     4  !    06  !
!      !---------+------------+-----+--------+--------!
!      !  M RCNT !   23953    !  10 !    FD1 !    04  !
+-------------------------------------------------------+
LEGEND
  SESS NO
    UA   : UNABLE TO RETRIEVE INFORMATION, DATA STRUCTURES UNAVAILABLE
  WHICH FAILURE
    FIRST : FIRST CONSISTENCY GROUP FAILURE
    PREV  : PREVIOUS CONSISTENCY GROUP FAILURE
    M RCNT: MOST RECENT CONSISTENCY GROUP FAILURE

  LSS, ERROR REASON, OR MASTER STATE CODE
    FO/IM: INFORMATION UNAVAILABLE DUE TO FAILOVER OR IML

  ERROR REASON
    01  : ASYNCHRONOUS PPRC STRUCTURES CANNNOT BE ACCESSED
    02  : ASYNCHRONOUS PPRC COMMUNICATION PATH FAILURE
    03  : EXTENDED DISTANCE CONSISTENCY SESSION MEMBERS NOT IN CORRECT STATE
    04  : MAXIMUM CONSISTENCY GROUP DRAIN TIME EXCEEDED
    400 : INVALID PARAMETER
    FC9 :SESSION ALREADY OPEN ON LSS OR MAXIMUM SESSIONS ALLOWED ALREADY METY
    FD1 : START REQUESTED AND ANOTHER MASTER IS ALREADY RUNNING OR MODIFY REQUESTED AND PARAMETERS DO NOT MATCH
EXISTING SESSION OR COMMAND ISSUED TO OTHER THAN ALLOWABLE LSS
    7ZZZ: MICROCODE LOGIC ERROR - ZZZ DESCRIBES ERROR
    F005: TEMPORARILY UNAVAILABLE
    F01D: LONG BUSY
 MASTER STATE
    01: PAUSE/TERMINATE ASYNCHRONOUS PPRC IN PROGRESS
    02: START/RESUME ASYNCHRONOUS PPRC IN PROGRESS
    03: ASYNCHRONOUS PPRC IS BETWEEN CONSISTENCY GROUP FORMATIONS
    04: XDC START INCREMENT IN PROGRESS
    05: XDC RUN IN PROGRESS
    06: DRAIN IN PROGRESS
    07: FLASHCOPY ESTABLISH WITH REVERTIBLE IN PROGRESS
    08: FLASHCOPY WITHDRAW WITH COMMIT IN PROGRESS
    09: XDC INCREMENT COMPLETE IN PROGRESS
    0A: FLASHCOPY WITHDRAW WITH REVERT IN PROGRESS
    0B: ASYNCHRONOUS PPRC IS PAUSED
    0C: ASYNCHRONOUS PPRC IS PERFORMING POST-CONSISTENCY GROUP TASKS
    0D: ASYNCHRONOUS PPRC IS FATAL
    0E: ASYNCHRONOUS PPRC IS COMPLETING ERROR RECOVERY

OTHER MASTERS TO THIS SUBORDINATE ASSOCIATIONS
 THERE ARE NO OTHER MASTERS TO THIS SUBORDINATE
```

12.5.2 Query the status of the volumes in the session

This query provides information on the queried LSS about the number of volumes that are part of a session or are still join pending. In the example below, the session 50 is in normal state and the first volume in LSS02 with CCA=00 is already a member of the session (IS), but the remaining volumes are still join pending (JP). See Example 12-15 for the syntax of the command.

Example 12-15 Query Session devices - command syntax

```
PPRCOPY QUERY              -
         DDNAME(VLB200)    -
         SESSDEV
```

Example 12-16 shows the output messages from QUERY SESSDEV command.

Example 12-16 Messages about Query Session devices command

```
ICK00700I DEVICE INFORMATION FOR B200 IS CURRENTLY AS FOLLOWS:
         PHYSICAL DEVICE = 3390
         STORAGE CONTROLLER = 2105
         STORAGE CONTROL DESCRIPTOR = E8
         DEVICE DESCRIPTOR = 0A
         ADDITIONAL DEVICE INFORMATION = 4A000035

ICK04030I DEVICE IS A PEER TO PEER REMOTE COPY VOLUME

EXTENDED DISTANCE CONSISTENCY SESSIONS AND DEVICES TABLE
+-------------------------------------------------------------------------------+
! SESS ! SESS !                                                                 !
!  NO  ! STAT ! VOLUMES IN SESSION                                              !
!------+------+-------------------------------------------------------------------!
!  50  ! N    !  02.00(IS,DP)   02.01(JP,DP)   02.02(JP,DP)                     !
+-------------------------------------------------------------------------------+
LEGEND
 SESSION STATUS
  CGIP = CONSISTENCY GROUP IN PROGRESS
  INAC = NO INFORMATION, DATA STRUCTURES INACCESSIBLE
  IP   = INCREMENT PENDING
  NAV  = NO ACTIVE VOLUMES IN SESSION
  N    = NORMAL
  NSES = NO SESSIONS DEFINED ON ESS

 VOLUME STATUS (1ST ENTRY IN PARENTHESES)
  IS   = IN EXTENDED DISTANCE CONSISTENCY GROUP SESSION
  JP   = EXTENDED DISTANCE CONSISTENCY VOLUME IS JOIN PENDING
  RP   = EXTENDED DISTANCE CONSISTENCY VOLUME IS REMOVE PENDING
  1P   = PPRC PAIR IS IN FIRST PASS OF INTIAL COPY

 VOLUME STATE (2ND ENTRY IN PARENTHESES)
  SX = PAIR IS SIMPLEX
  DP = PAIR IS DUPLEX PENDING
  FD = PAIR IS FULL DUPLEX
  SP = PAIR IS SUSPENDED
```

12.5.3 Query the number of volumes that are out-of-sync

The Query OUTOFSYNCSTATE returns information about the number of volumes in the specified session that are out-of-sync due to the initial PPRC-XD or cyclic synchronization process. The more volumes at the primary site that have a high workload, the more volumes will be out-of-sync. With quiesced workloads on all primary volumes, the number of out-of-sync volumes become zero in time, so you can get an indication of how the application workload is distributed across your primary volumes. Example 12-17 shows the syntax of the command.

Example 12-17 Query the number of out-of-sync volumes - command syntax

```
 PPRCOPY QUERY OOSSTATE            -
         DDNAME(VLB000)            -
         SESSIONNO(50)
```

Example 12-18 shows the output messages from the QUERY OOSSTATE command.

Example 12-18 Messages from Query the number of out-of-sync volumes

```
ICK00700I DEVICE INFORMATION FOR B000 IS CURRENTLY AS FOLLOWS:
         PHYSICAL DEVICE = 3390
         STORAGE CONTROLLER = 2105
         STORAGE CONTROL DESCRIPTOR = E8
         DEVICE DESCRIPTOR = 0A
         ADDITIONAL DEVICE INFORMATION = 4A000035
ICK04030I DEVICE IS A PEER TO PEER REMOTE COPY VOLUME

ASYNCHRONOUS PPRC OUT-OF-SYNC INFORMATION AT LSS LEVEL
 3    VOLUMES IN SESSION 50  ARE OUT OF SYNC
```

12.5.4 Query the status of the FlashCopy volumes after an outage

When Asynchronous PPRC fails in case of a disaster or a planned failover and the state of the FlashCopy volumes could be inconsistent. The query output provides an indication whether the FLASHCOPY WITHDRAW command to the FlashCopy primary volumes at the remote site has to be issued, and, if yes, which of the parameters REVERT or COMMIT has to be used with that command.

- *COMMIT* means that all FlashCopy volumes are set to the latest Consistency Group
- *REVERT* means that all FlashCopy volumes are reset to the previous Consistency Group.

The decision depends on the state of the FlashCopy relations of each volume pair and if the Consistency Group number is equal for each pair or not.

> **Note:** Usually none of these Withdraw commands have to be applied because the FlashCopy volumes are in a consistent state.

The query command to use depends whether the Master ESS and links are available. In case of a real disaster, the primary site usually is not available and you should assess the status of FlashCopies at the remote site using the QUERY RELATIONS command.

Query for the Increment Status if the Master ESS and links are available

This query can be only addressed to the Master or Subordinate ESSs and it should be used for debugging purposes only. Example 12-19 shows the command syntax.

Example 12-19 Query INCREMENTSTATUS - command syntax

```
FLASHCPY QUERY INCRSTAT          -
         DDNAME(VLB000)          -
         SESSIONNO(50)
```

Example 12-20 shows the output messages from the QUERY INCRSTAT command.

Example 12-20 Messages from Query INCREMENTSTATUS command

```
ICK00700I DEVICE INFORMATION FOR B000 IS CURRENTLY AS FOLLOWS:
         PHYSICAL DEVICE = 3390
STORAGE CONTROLLER = 2105
STORAGE CONTROL DESCRIPTOR = E8
DEVICE DESCRIPTOR = 0A
         ADDITIONAL DEVICE INFORMATION = 4A000035
ICK04029I DEVICE IS IN SUSPENDED PPRC STATE

FLASHCOPY ASYNCHRONOUS PPRC INCREMENT STATUS INFORMATION
 SESSION NUMBER: 50
 SESSION IS NON-REVERTIBLE AND EQUAL (NO WITHDRAW REQUIRED)
```

Use QUERY RELATIONS if Master ESS and links are not available

If the Master ESS is no longer available and/or the PPRC paths to the secondary volumes do not exist anymore, you can assess the status of FlashCopies using the **QUERY RELATIONS** command. This query has to be submitted to each PPRC secondary volume to obtain the FlashCopy Sequence Number and the other pertinent information, like if the volume is revertible or not.

You need to summarize all the volume information and set the appropriate action based on the corresponding indication as in Table 12-2. Due to the extensiveness of this process, we suggest that you use automation tools to collect all the information.

Table 12-2 Indication of revertible or non-revertible

Indication	Specification on FlashCopy Withdraw command
Revertible or non-revertible volumes and equal sequence numbers	Commit
Non-revertible volumes and equal sequent numbers	No withdrawal required
Revertible and non-revertible volumes	Revert
All volumes revertible	Revert
Consistency Group corrupted	N/A

Example 12-21 shows the command syntax.

Example 12-21 Query Relations - command syntax

```
FLASHCPY QUERY RELATIONS UNITADDRESS(3000)
```

Example 12-22 shows the output messages from the QUERY RELATIONS command.

Example 12-22 Messages from Query Relations command

```
ICK00700I DEVICE INFORMATION FOR 3000 IS CURRENTLY AS FOLLOWS:
         PHYSICAL DEVICE = 3390
         STORAGE CONTROLLER = 2105
         STORAGE CONTROL DESCRIPTOR = E8
         DEVICE DESCRIPTOR = 0A
         ADDITIONAL DEVICE INFORMATION = 4A000035
ICK04032I DEVICE SPECIFIED IS A PPRC SECONDARY
FLASHCOPY RELATIONS INFORMATION TABLE
SOURCE CU SERIAL NUMBER: 0000024663
+-----------------------------------------------------------------------+
! FLASHCOPY  !     !   !    !     !     !     !     SOURCE/TARGET     !
! SEQUENCE   !     !   !    !     !     !     !-----------------------!
! NUMBER     ! R/T ! R ! FV ! BCE ! BCP ! CRA ! VR ! SSID ! LSS ! CCA !
!-----------------------------------------------------------------------!
! 1083097742 ! SRC ! F !  T !  F  !  F  !  T  ! F  ! 1071 ! 01  ! 00  !
+-----------------------------------------------------------------------+
LEGEND
 R/T = FLASHCOPY RELATIONSHIP TYPE (SRC = SOURCE, TGT = TARGET)
 R   = REVERTIBLE (TRUE ! FALSE)
 FV  = FULL VOLUME RELATION (TRUE ! FALSE)
 BCE = BACKGROUND COPY ENABLED (TRUE ! FALSE)
 BCP = BACKGROUND COPY IN PROGRESS (TRUE ! FALSE)
 CRA = CHANGE RECORDING ACTIVE (TRUE ! FALSE)
 VR  = VALIDATION REQUIRED (T!F), REPORTED INFORMATION NOT CURRENT AND MAY NOT BE ACCURATE
```

12.5.5 Asynchronous PPRC Session Manager

Asynchronous PPRC Session Manager (APSM) is designed to provide commands for the single point setup and management of Asynchronous PPRC, where automation is needed for a large number of volumes. It is designed to complement existing interfaces, such as ICKDSF, CLI, Web User Interface, and the Multiple Device Manager Replication Manager. APSM simplifies the setup, management, and recovery of Asynchronous PPRC implementations. The commands provided can easily be scripted to integrate with a client's environment.

The APSM is distributed on a separate CD with ESSs at LIC 2.4.0 or higher. On that CD there is a README file for installation of the commands as well as a *User's Guide*. There are two components, the APSM server and the user client.

The APSM will run on a dedicated Windows® server which meets the following requirements:

- Intel® Pentium® 4 or compatible host server
- Microsoft® Windows 2000 Advanced Server with SP3
- Minimum memory of 1 GB
- Minimum disk space of 40 GB
- One of the following Web Server products:
  - WebSphere Application Server 5.1 or compatible Web Server or
  - Apache Tomcat 5.0 has been fully tested and is supported.

The APSM client workstation requirements are:

- Microsoft Windows 2000 or Windows XP
- Java™ 2 Runtime Environment, Standard Edition (build 1.3.1)

The APSM server receives requests from the client and communicates over TCP/IP to the ESS for the execution of the APSM commands. The initial set of commands includes:

- Create, modify, start, stop, and resume an Asynchronous PPRC session
- Manage Failover and Failback operations including managing consistency
- Perform planned outages

To monitor the ESS volume status and the Asynchronous PPRC session status you will use the appropriate ESS Web User Interface or the CLI commands.

The APSM commands rely on configuration files which are created by the user. Future versions of APSM will complete and enhance these functions. It exists as a standalone set of commands and in the future will also be integrated into the functionality of the Multiple Device Manager Replication Manager.

12.5.6 RMF enhancements: Performance statistics reports for ESS links

With ESS LIC version 2.3.x or higher and APAR OA04877 for RMF™, new performance statistics of the Enterprise Storage Server for Fiber Channel (FC) PPRC adapters or servers connected via FC adapters can be collected and reported. The reports may be used for the analysis of the external link usage and for performance planning of PPRC and XRC. Currently there is no support available for ESCON adapters.

These link statistics are saved in a new subtype 8 of SMF record 74. The RMF Postprocessor is enhanced by a new ESS Link Statistics report. Refer to the APAR for the detailed description of the field format of the new subtype 8 of SMF record 74.

RMF Monitor I options

In order to enable or disable the gathering of the RMF Monitor I you simply specify these parameters in the PARMLIB options member for RMF:

```
ESS / NOESS
```

NOESS is the default. ESS specifies that Enterprise Storage Server link statistics should be collected. As ESS data gathering involves cache activity measurement, it is recommended to specify ESS *and* CACHE options. If ESS and NOCACHE is specified, cache data gathering is performed implicitly without writing SMF 74 subtype 5 records. Options CACHE and ESS can be specified on any system sharing the selected devices in a Sysplex. Therefore, specify options ESS and CACHE together on one selected system to avoid duplicate data gathering.

This APAR updates PARMLIB members ERBRMF00 and ERBRMF02 for Monitor I gathering sessions. ERBRMF00 specifies the NOESS option while ERBRMF02 specifies ESS.

RMF Postprocessor Report for link statistics

The syntax of the reports and the values for the exception conditions are described in *z/OS V1R5.0 Resource Measurement Facility User's Guide*, SC33-7990.

To produce this report, specify the REPORTS(ESS) option in the Postprocessor job. This report contains statistics about the I/O operations for each adapter of an ESS.

In Example 12-23 you can see the throughput of two PPRC links and two adapters connected to server HBAs.

The following link types are currently reported in RMF:

- SCSI protocol read and write for Fibre Channel links
- PPRC send and receive activity for Fibre Channel

The report contains values for:

- Transfer rate in MB per second
- Number of MB/s per operation
- Operations per second
- Response Time per operation
- I/O intensity

Example 12-23 New RMF Link Performance Report

```
1                                          E S S  L I N K  S T A T I S T I C S

PAGE    1
            z/OS V1R4                SYSTEM ID GDP5             START 05/10/2004-16.35.00  INTERVAL 000.04.59
                                     RPT VERSION V1R2 RMF       END   05/10/2004-16.40.00  CYCLE 1.000 SECONDS
0SERIAL NUMBER 0000023953   TYPE-MODEL    2105-800       CDATE  05/10/2004     CTIME 16.35.00      CINT  00.04.59
0------ADAPTER------   --LINK TYPE--       BYTES      BYTES          OPERATIONS      RESP TIME         I/O
 SAID  TYPE                                /SEC       /OPERATION     /SEC            /OPERATION        INTENSITY
00008  FIBRE 2Gb       NO DATA TO REPORT OR ZERO
0000C  FIBRE 2Gb       NO DATA TO REPORT OR ZERO
00024  FIBRE 1Gb       SCSI READ            0.0        0.0            0.0             0.0               0.0
                       SCSI WRITE           8.1M     262.1K          30.9            18.7             579.9
                                                                                                        ------
                                                                                                         579.9
00028  FIBRE 2Gb       NO DATA TO REPORT OR ZERO
0002C  FIBRE 1Gb       PPRC SEND            5.0M      27.0K         183.5            29.2            5356.5
                       PPRC RECEIVE         0.0        0.0            0.0             0.0               0.0
                                                                                                        ------
                                                                                                       5356.5
00084  FIBRE 1Gb       SCSI READ            0.0        0.0            0.0             0.0               0.0
                       SCSI WRITE           8.7M     262.3K          33.0            19.0             626.8
                                                                                                        ------
                                                                                                         626.8
00088  FIBRE 2Gb       NO DATA TO REPORT OR ZERO
0008C  FIBRE 1Gb       PPRC SEND            4.9M      26.8K         184.5            29.1            5362.3
                       PPRC RECEIVE         0.0        0.0            0.0             0.0               0.0
                                                                                                        ------
                                                                                                       5362.3
000A8  FIBRE 2Gb       NO DATA TO REPORT OR ZERO
000AC  FIBRE 2Gb       NO DATA TO REPORT OR ZERO
```

This ESS Link Statistics report provides information for all FC adapters that have activities in the report interval. The report shows two outgoing PPRC links with 5 MB/s data rate during the time interval of 5 minutes and an average I/O rate of 183 I/Os and 262 KB I/O block size.

The *Response Time per Operation* is the entire time from sending out a data block until the return receipt of the receiver arrives. On an FC, multiple data blocks can be sent in sequence regardless of the return receipts of a previous blocks. Therefore the *Number of Operations per Second* times the *Response Time per Operation* can be more than 1000 ms per second. This value is called the *IO Intensity* of the adapter and is displayed in the right column.

12.6 Recovery procedures

In many ways, Asynchronous Peer-to-Peer Remote Copy recovery from an unplanned outage is very similar to the two-site synchronous Peer-to-Peer Remote Copy Failover/Failback procedure.

12.6.1 Unplanned outage and switch to the remote site

Asynchronous PPRC is designed to provide application consistent data at a remote site at a global distance even in cases of a rolling disaster at the local site.

If you had a disaster at the primary site and therefore have to start the production at the remote site, you will need to ensure that you are going to use the data from the most valid Consistency Group at the secondary site.

If you have a running system at the remote site that is independent of the local site, you can use the ICKDSF commands to perform the failover, and you should have already prepared and tested all jobs and/or automation procedures you might need. Otherwise, you have to use the Web User Interface with prepared tasks.

A basic failover procedure could be done as follows:

1. If the Master of the Asynchronous PPRC session is still active, issue the TERMINATE SESSION command to the Master, otherwise go to step two. This will stop the Master trying to create Consistency Groups if this process is still running. Address this command to the same LSS you used to start the session.
2. Failover to the PPRC secondary volumes using the ESTABLISH PAIR command with the FAILOVER option at the remote site to the PPRC secondary volumes. These volumes become *primary suspended*.
3. It is required to check the consistency status of the FlashCopy pairs even if it is unlikely to have inconsistent data at the FlashCopy target volumes. This can be done by issuing the FLASHCPY QUERY RELATIONS command to each of the FlashCopy primary volumes. In most of the cases the volumes are consistent and you can continue with step 6. See Table 12-2 on page 193 for more details on the consistency and the necessary actions. If the indication directs you to revert the FlashCopy pairs back to the previous FlashCopy relationship, submit the FLASHCPY WITHDRAW command with the REVERT option. If you are directed to commit all of the FlashCopy pairs to the new FlashCopy relationship, submit the FLASHCPY WITHDRAW command with the COMMIT option.

   Note: These commands do not withdraw the relationships.

4. Issue a FLASHCPY ESTABLISH PAIR command with FASTREVERSERESTORE and TGTOKASPPRCPRIM(YES) parameter to the PPRC secondary volumes. This will reverse the direction of the FlashCopy relationship and copy the consistent data from the FlashCopy targets to the suspended PPRC secondary volumes at the remote site.

   Note: The **FlashCopy** relationship will end after the background process has finished. The former FlashCopy target volumes, the **C** volumes (see Figure 12-3 on page 173) are no longer usable.

5. Before you continue, the background copy process should be finished. This should be very fast, as only small data differences have to be copied.
6. Establish a FlashCopy from the PPRC secondaries to the original FlashCopy target volumes. This provides you with a safe copy of your environment before allowing anyone to update it.
7. Set the former PPRC secondary volumes online to the remote systems, test the data and start the applications.

12.6.2 Planned outage

When you have implemented Asynchronous PPRC, you may need to plan outages to enable hardware maintenance at the local or the remote site or to generate another copy of the data at the remote site for disaster recovery tests or production test purposes. Usually the hardware maintenance of the Enterprise Storage Server is non-disruptive but if you run Asynchronous PPRC you have many components in the chain of data flow that have to be considered.

The following steps can be done by using the Web User Interface or ICKDSF for the z/OS volumes.

Planned outage at the remote site

This can be done by implementing the following steps:

1. Issue the PPRCOPY TERMASYNCCOPY command to pause the session. This will stop the Asynchronous PPRC session but let finish the creation of the latest Consistency Group at the FlashCopy target volumes at the secondary site. Address this command to the same LSS you used to start the session. You should check the status of the session using the PPRC QUERY ASYNCCOPY command before you continue.
2. Suspend the PPRC-XD relationships. Issue the suspend command to all of the primary volumes. The PPRC secondary volumes become suspended secondary volumes and contain fuzzy copies of the data.
3. Now you can perform the actions you have planned for this outage at the remote site. Usually this is for maintenance or reconfiguration.
4. If the secondary site is ready again, you can reestablish the PPRC-XD copy pairs with the RESYNC parameter, to update the remote site volumes with the data that has changed at the local site.
5. Restart the Asynchronous PPRC session with the same options as before and check with the PPRC QUERY ASYNCCOPY command if the formation of Consistency Groups continues.

Planned outage at the local site and switch to the remote site

If you have to shut down the primary site for any reason and run the application at the other site for some time, and you want to return production at the local site later again, you can use the following failover and failback procedure:

1. Shut down the applications at the local site.
2. Wait until all PPRC-XD pairs do not have any out-of-sync cylinders. They should take a short time to catch up and the primary and secondary volumes will have the same data. You can get this information easily with QUERY OOSYNCSTATE of ICKDSF or the Asynchronous PPRC Information panel of the Web User Interface for every primary ESS.
3. Issue the TERMASYNCCOPY command with the PAUSE option to pause the session. This will stop the Asynchronous PPRC session but finish the creation of the latest Consistency Group at the FlashCopy target volumes at the secondary site. Address this command to the same LSS you used to start the session. You should check the status of the session using the PPRC QUERY ASYNCCOPY command before you continue.

Tip: Issue the pause first and ensure success before terminating to ensure a current consistent copy has been taken.

4. Next, issue the TERMASYNCCOPY command with the TERMINATE option to terminate the session. You should again check the status of the session using the PPRC QUERY ASYNCCOPY command before you continue.
5. Suspend PPRC for all the volumes using the PPRCOPY SUSPEND command.
6. Failover to the PPRC secondary volumes with PPRCOPY ESTPAIR command at the remote site to the PPRC secondary volumes. These volumes become *primary suspended*. Now the maintenance at the primary site can be done.
7. You can start the applications at the remote site now. The FlashCopy target volumes still contain the data from the time of failover. Be aware, that you are not protected from disasters at the remote site until you switch back to the local site and re-establish Asynchronous PPRC.
8. If the local site is ready for production again, you should establish PPRC paths to the local site ESS to enable the reverse PPRC mirror to the local site. As Fibre Channel PPRC links are bidirectional links these paths can be established prior to the procedure.
9. Issue the PPRC ESTABLISH PAIR command with mode XD and FAILBACK option to the remote site PPRC volumes. Now the local site volumes change from *primary suspended* to *PPRC pending* copy state. The changed data from the remote site will be drained now to the local site.
10. Wait until the first pass of the PPRC-XD copy process of all volumes has finished. The PPRC QUERY command can be used to check the volumes for out-of-sync cylinders. Alternatively you can catch-up the volumes to the duplex state with the PPRC ESTABLISH PAIR command with OPTION(SYNC).
11. When the PPRC copy has completed the first pass, shut down the applications at the remote site.
12. Submit the PPRCOPY ESTPAIR command with the FAILOVER option to the local volumes that will become primary suspended.
13. Now you can start the applications at the local site.

Tip: Ensure that applications are stable before copying data to the remote site. This will protect the data at the remote site which could be returned to if applications at the primary site failed to start.

14. Reestablish the paths from the local to the recovery site again. Issue the PPRCOPY ESTPAIR command with FAILBACK and Mode XD to the local volumes.
15. Finally, Asynchronous PPRC has to be restarted again by submitting the PPRCOPY STARTASYNCCOPY command to the local volumes.

12.6.3 Considerations

In summary, here are some considerations to keep in mind:

- Advantages:
  - The Consistency Group is maintained across multiple ESSs.
  - The RPO can be <5 seconds if adequate bandwidth exists.
- Disadvantages:
  - You are limited to a total of eight ESSs.
  - Three copies of data are needed (four copies to test the solution).
- Issues:
  - The Web User Interface has a limit of eight ESSs: ICKDSF has a limit of sixteen ESSs.

13

XRC: Global Mirror for z/OS

We have introduced XRC in Chapter 8, “eXtended Remote Copy” on page 113. In this chapter we provide a justification for the new name for it: Global Mirror for z/OS, by showing details of a disaster recovery solution based on XRC.

When planning to implement and use XRC, the following publications should be referenced to complement the information presented in this chapter:

- *IBM TotalStorage Implementing ESS Copy Services with IBM eServer zSeries,* SG24-5680-04
- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547
- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *DFSMS Extended Remote Copy Installation Planning Guide*, GC35-0481
- *DFSMS Extended Remote Copy Reference Information for Advanced Users*, GC35-0482

This chapter covers the following aspects of an XRC solution:

- Solution overview
- Design the solution
- Define XRC management
- Define XRC automation
- Implement the solution
- Run the solution
- Database recovery with XRC
- Using XRC with FlashCopy

13.1 Solution overview

XRC is a copy function available for the z/OS and OS/390 operating systems. It involves a System Data Mover (SDM) that is found only in DFSMSdfp component of OS/390 and z/OS. XRC maintains a copy of the data asynchronously at a remote location, and can be implemented over unlimited distances. It is a combined hardware and software solution that offers data integrity and data availability that can be used as part of business continuance solutions, for workload movement, and for data migration. XRC requires an optional feature 2240-XRC on the ESS.

For the scope of this chapter, we intend to build a solution for a basic XRC configuration as depicted in Figure 13-1.

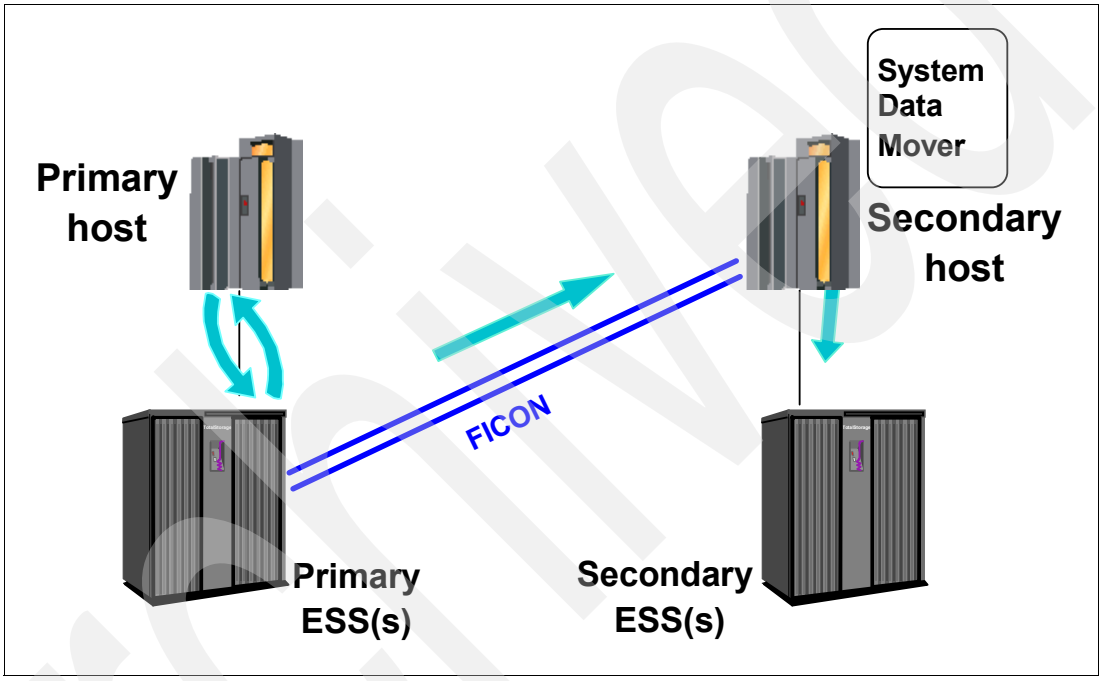


Figure 13-1 Our XRC environment

In this configuration the production environment runs on the primary host, while the System Data Mover runs on a host located at the recovery site. This second host will also be used to run the production environment in case the primary site is unavailable to provide services.

Possible XRC configurations

As the SDM is part of DFSMSdfp, it can run in any z/OS environment. Basically, there are three possible configurations:

- SDM at the primary site
- SDM at an intermediate site
- SDM at the secondary site

For disaster recovery, we suggest that you place SDM at recovery site, since this provides two benefits as compared with the other two options:

1. It requires less telecommunication bandwidth and thus reduces the total cost of ownership of the solution.
2. It keeps the data delay at minimum because the SDM buffers in the CPU are also safe; this allows SDM to keep on creating and applying consistency groups from its buffers.

For the rest of the chapter, we assume that the SDM is located at the secondary site. We also assume that primary and secondary disks in the XRC configuration are ESS model 800, and that the secondary disks subsystem are configured with FlashCopy capability as well as all the required physical space.

Remember that Global Mirror for z/OS is a z/OS solution only and it cannot handle data on other platforms, therefore only z/OS hardware can be used.

13.2 Design the solution

In these sections we provide the design concepts for a general disaster recovery solution using XRC.

13.2.1 Verify hardware requirements

To play the role of a Primary Subsystem, the ESS must have the proper XRC-capable Licensed Internal Code (LIC). This is an optional fee feature that must be ordered. Table 13-1 provides a list of optional Copy Services features needed to implement the solution.

Table 13-1 XRC and FlashCopy feature codes

Physical capacity tier	ESS Model 800 (2105-800) feature codes IBM 2240 ESS Function Authorization feature numbers		
	Extended Remote Copy 2240-XRC	FlashCopy 2240-FLC	
		Version 1	Version 2
Up to 1 TB	8100	8300	8600
Up to 2 TB	8101	8301	8601
Up to 3 TB	8102	8302	8602
Up to 4 TB	8103	8303	8603
Up to 5 TB	8104	8304	8604
Up to 6 TB	8105	8305	8605
Up to 8 TB	8106	8306	8606
Up to 10 TB	8107	8307	8607
Up to 12 TB	8108	8308	8608
Up to 16 TB	8109	8309	8609
Up to 20 TB	8110	8310	8610
Up to 25 TB	8111	8311	8611
Up to 30 TB	8112	8312	8612
Up to 40 TB	8113	8313	8613
Up to 50 TB	8114	8314	8614
Up to 60 TB	8115	8315	8615
Inactive 0TB	8199	8399	8699

Primary systems (CPUs) must have a common time reference. XRC uses timestamped record information to ensure that updates are not copied out of sequence at the secondary site:

- In environments where the primary systems are on different CECs, an IBM Sysplex Timer is required.
- When you have only one primary system, or the primary systems are running on different LPARs on the same CEC, the system time-of-day clock is used.

A compatible secondary volume must be available for each primary volume you want to copy. The secondary volume must have the identical track capacity and number of tracks per cylinder as the primary volume. The capacity of the secondary volume can be the same or larger than that of the primary volume.

The secondary disk subsystem is not required to have XRC-capable Licensed Internal Code (LIC). However, if you want to use the alternate site to manage planned unavailability of the production site, we suggest that the secondary disk subsystems to be at the same technology level ad the primary disk subsystems.

Lastly, XRC protects for physical event only; the use of FlashCopy and more secondary space provides with the ability to run backup from the FlashCopy environment and to have a safe second point of restart in case of any emergency.

13.2.2 Verify software requirements

XRC is a software solution and System Data Mover function requires current OS/390 or z/OS releases with the latest maintenance applied. For the latest maintenance information, refer to the Web site:

http://www.ibm.com/servers/storage/support/solutions/bc.html

VM considerations

When XRC manages volumes that are also attached to a VM system, you must define all XRC volumes to VM as unsupported disks. You *cannot* define them as minidisks.

Note that for a disk device that is defined as unsupported under VM, the SET PATH GROUP command is unable to reestablish the path group once a path has been removed.

13.2.3 Select volumes for remote copy

In this section, we discuss some considerations regarding which volumes to include in the remote copy solution.

Volume level copying

Remember that like any other disk mirroring solution, XRC copies the entire contents of a volume. As a result, all data sets on a primary volume are copied to the secondary volume. This support is for all data types, independent of the applications.

What data should be copied?

Before you start up your XRC environment, you have to identify the data you will want to mirror onto the secondary site.

You should identify the type of data required for a successful recovery. Many installations with an existing and tested traditional disaster recovery solution have completed this task already.

The SYSRES, master catalog, SPOOL volumes, as well as other data sets required to initialize the secondary system (including those used to start up JES, TSO, and RACF®) can be copied by XRC. The idea is that even if we do not need the data on some of these volumes, we do need either the data sets correctly allocated and the space formatted (pages, spool, SMF man data sets) available for the production jobs to use (temporary storage groups). Some of these volumes can be initially synchronized and then suspended, so XRC stops copying their updates.

However, the focus for remote copy is on your application data sets that are updated regularly, and are essential for your company. If a disaster occurs, critical application volumes must be made available to your primary application as quickly as possible, depending on the speed of recovery required. XRC offers a single command recovery strategy, which makes it the fastest method for recovering data at the secondary site. A single XRECOVER command recovers all the volumes in an XRC session at once, and all of the volumes are consistent to a single point-in-time.

Application volumes

Different applications at the primary site may have different priorities. Some of these applications may not warrant the investment required for remote copy. If only a subset of applications is to be copied, the volumes belonging to those applications must be identified and included in the XRC configuration.

Because applications deal with data sets and not volumes, multi-volume data sets require special attention. Multi-volume data set types include data sets that reside on multiple volumes, striped data sets, and VSAM spheres. If you want a copy of a multi-volumes data set at the secondary location, you therefore have to copy all volumes on which this data set resides.

System volumes

The following data sets should be given special consideration, as they might be eliminated from a particular installation's XRC configuration.

Page

Page data sets are of no use during recovery at the recovery site. They are owned by the host application site. However, if we do not have the page data sets, then we will not be able to IPL the production system at the recovery site until we create and catalog new ones. What can be done is to initially synchronize and then suspend the page data sets. You do not want to have the page data sets in duplex due to the type of I/O that they use.

Spool

The spool data sets should be copied to the recovery site if they are necessary for recovery. Copying the spool increases the amount of copy activity to the secondary. If the contents of the spool are easily re-created, consider excluding spool volumes from the XRC configuration. Alternatively, you can initially synchronize the spool volumes and then suspend them as with the page data sets.

Temporary data sets

Volumes containing temporary data sets (that is, those that exist for the duration of a job or step) can be re-created at the secondary by resubmitting jobs. These volumes should be excluded.

SYSRES volume

The change activity against data sets residing on the SYSRES often does not affect the applications. However, some installations share the SYSRES volume with active data, so the decision on whether to copy the SYSRES volume can depend on the data stored there. Because XRC copies data at the volume level, all data residing on that volume is copied.

If the SYSRES volume is not copied by XRC operations, another method must be used to ensure that a copy is available at the secondary site. Nevertheless, commonly the SYSRES as well as other system volumes will be copied.

Master catalog and user catalogs

Catalog changes are limited, but they are essential for recovery, and therefore they must be considered for XRC copying.

Program libraries

All program libraries that contain modules for XRC applications are essential for recovery and should be considered for XRC copying.

Control data sets

Control data sets, such as RACF, SMS, HSM, and RMM are essential for recovery and should be considered for XRC copying.

HSM migration level 1 volumes

An HSM migrated data is still a production data that hasn't been referenced for a time. Excluding HSM migrated level 1 volumes might generates errors days after the recovery, since when it is time to recall the data set, there might be no data set to recall at all.

> **Attention:** We suggest to treat HSM migration level 1 volumes the same as application volumes.

13.2.4 The ERRORLEVEL parameter

As part of the planning process, you should evaluate your data consistency requirements in the event of a failure. For a DB2 environment, we suggest you to use the ERRORLEVEL SESSION subparameter to suspend all volume pairs in the session.

If a volume with error level SESSION encounters an error, all volumes in the XRC session will be suspended at the same time. If a failure occurs on a single LSS, and that LSS attaches to any duplex volume pair with error level SESSION specified, then all volume pairs in the XRC session are put into the suspended state. XRC also suspends volume pairs that are in pending state, and the pairs remain where they are in the synchronization process.

You must correct the error and resynchronize all the suspended pairs. However, other options are available, and you can specify the default error level for the entire session with the ERRORLEVEL parameter on the XSTART command, but you can change the error level for specific volume pairs with the XADDPAIR ERRORLEVEL parameter.

The other options available are:

- The *VOLUME* option suspends an XRC volume pair that encounters a device error. XRC continues to process all other volumes in the session. If a volume that is suspended due to an error contains data that is dependent on data on other volumes, data consistency cannot be assured in the event of a disaster. In those cases you should use group_name or SESSION (see below) as the error level specification.

- The *group_name* option allows you to associate any valid TSO name with one or more volume pairs. If a volume with error level group_name encounters an error, XRC will suspend all volume pairs which have the same group_name specified. An error that occurs on a single LCU that attaches to any duplex volume pair that has group_name specified causes XRC to suspend all volume pairs with the same group name specified.

  Use group_name to associate volumes belonging to an application to ensure consistency in the application.

Table 13-2 summarizes what we have discussed so far on how ERRORLEVEL options determine volume pair suspension in the event of different types of failures.

Table 13-2 ERRORLEVEL and suspension of volume pairs

ERRORLEVEL options	Single volume failures	Primary LCU failures	SDM failures
VOLUME	One duplex volume pair is suspended	All active volumes on the LCU are suspended	All duplex volume pairs in the XRC session are suspended
SESSION	All duplex volume pairs in the session are suspended	All active volumes in the session are suspended	All duplex volume pairs in the XRC session are suspended
group-name	All duplex volume pairs in the group are suspended	All active volumes on the LCU are suspended, and all active volumes in this group on any LCU are suspended	All duplex volume pairs in the XRC session are suspended

13.2.5 Verify SDM requirements

The SDM manages the XRC activity for all volume pairs in an XRC session.

Number of XRC sessions

Up to five XRC sessions are allowed in each z/OS (or OS/390) system image. If you have more then 1500 to 1800 volumes in a single XRC session, you may need to split these into two or more XRC sessions. In addition to the workload balancing split, also note that unless you use CXRC there is no write sequence consistency between XRC sessions, so you must consider splitting the volumes between XRC sessions on an application basis also.

Virtual storage

XRC operates in up to six address spaces on a single LPAR. ANTAS000 is always active and processes XRC commands. Up to five XRC sessions may be started at a time. Each XRC session that is started on a system occupies its own address space, ANTAS001 through ANTAS005.

The ANTAS000 address space uses 65 MB of virtual storage, which includes a 20 MB trace data space. An ANTASnnn (started when the XSTART command is issued) has the following virtual storage requirements:

- Approximately 20 MB for base code.
- 30 MB for trace data spaces.
- Volume initialization requires 360 KB for each concurrent volume initialization task.

- About 35 MB for each active XRC storage control session, up to a maximum of 1.7 GB. SDM uses this space for buffers to hold data read from the primary LCU. These data buffers are required from the time when the data is first read from the ESS cache, through the Consistency Group formation process, the journaling process, and the writes to the cache and NVS of the secondary subsystem.

XRC needs a minimal amount of space (0.2 KB) in the extended common service area (ECSA). All other storage used is within the XRC address spaces.

Real storage

The main real storage requirements for SDM will be for storage buffers used during I/O operations. Those pages will be fixed for the duration of the input or output operation.

The actual real storage required at a given point depends on how much concurrent activity is occurring. You should provide enough real storage on the SDM system to support the projected peak I/O load.

For example, as SDM requires up to 35 MB data buffers from the time the data is read from the ESS cache until the data is written to the secondary volumes, you should plan for a real storage requirement equal to 35 MB times the number of XRC storage control sessions. Part or all of this can be permanently fixed real storage, using the PAGEFIX parameter of the XSET command. All storage beyond the PAGEFIX limit is fixed and freed as required. Specifying a high PAGEFIX value will reduce the MIPS required for fixing and freeing pages for those data buffers.

Tip: We highly recommend that all storage used by the SDM be pagefixed.

Paging space

The SDM requires paging space to support the address spaces, and you should plan for up to 2 GB for this space, depending on your virtual storage requirements (see “Virtual storage” on page 207).

Important: if you are using OS/390, we recommend that you use expanded storage for paging.

Processor (MIPS) requirements

This number varies mainly depending on the block size and the write I/O rate. With a larger block size, the SDM has to move more data, and consequently uses more resources. At a high I/O rate, SDM tends to be more efficient than at a low write rate.

The processor consumption is usually expressed as the number of MIPS required per 100 writes/sec to primary volumes from primary systems (XRC writes). There may be some increase (up to 25%) in the MIPS requirement when using channel extenders.

When the SDM is in its most efficient mode, it only requires 5 MIPS/100 write IOs/sec.

For more information on this subject, refer to:

http://www.ibm.com/servers/storage/support/solutions/bc.html

Then select *XRC*, then select *Technical notes*, and search for *MIPS*.

Tips:

- Use 5 MIPS/100 Write IOs to size SDM MIPS requirement (4 KB).
- Increase this estimate by 10% for higher (16 KB) average transfer size.
- Increase this estimate by 25% for channel extension to 2000 miles.
- Further increase this estimate if there are a significant sequential Write IOs.

SMS definitions

If the journal data sets are striped (which we recommend), they must be on SMS managed volumes. The state data set is a PDSE.

13.2.6 Verify the primary ESS subsystems

You have to assess the capacity of the primary ESS. In addition to reads and writes issued from the primary systems, the ESS must also be able to handle the reads associated with the SDM offloading process. The data written to primary volumes by primary systems will be grouped into record sets in cache, and those record sets will be offloaded by the SDM system with the read record set channel command.

The cache requirement should also be evaluated. There is no separate copy in the cache of the data to be offloaded, the ESS merely builds a logical queue using a directory structure.

When the SDM operation is balanced (meaning that the SDM does not get delayed when retrieving data from the primary ESSs and committing those updates to the journal data sets and secondary volumes) the primary ESS needs little more cache than what is normally required to satisfy primary application performance requirements.

As long as the SDM manages to retrieve the updates before the ESS caching algorithms would have removed them out of cache, minimal additional cache is required.

However, if the primary systems write rate is (temporarily) more than the SDM can absorb, or if there is a (temporary) disruption in the SDM data flow, the cache will be used as a buffer (see also 8.1.5, “Unplanned outage support for ESS” on page 118).

Our recommendation is that you evaluate the cache size required to satisfy your primary application needs, and then plan for XRC buffers. This might mean you have to install more cache (if possible), or increase your cache-to-backstore ratio by spreading the volumes across multiple ESSs. Your IBM Field Technical Sales Specialist can help you determine the cache needed for XRC by running the Disk Magic modeling tool.

13.2.7 Verify the secondary storage subsystem

The secondary storage subsystem can be any subsystem supported by the SDM system, as long as it provides volumes with the same geometry and the same capacity (or greater) as its corresponding primary volume.

Attention: If a secondary volume has more capacity than a primary volume, it has to be carefully considered if a copy-back procedure is implemented.

The secondary site needs the same number of XRC volumes as the primary site. Ideally, the number of LCUs at the secondary site should match the number of LCUs at the primary site, in order to provide configuration symmetry.

The secondary site storage controls must be configured to handle all of the primary volume writes plus the I/Os to the journal, control and state data sets as a minimum. In addition, they must also be capable of supporting the I/O activity related to the primary application's requirements in a disaster recovery situation.

When planning for the secondary storage subsystem, consider that you need FlashCopy capability at the secondary site, mainly for the following reasons:

- Like any other DASD mirroring solution, XRC doesn't protect from "logical" disasters — if something writes over your primary data, those writes will be mirrored on the secondary site. You need to have the capability to take some consistent restart-points during the day. XRC and FlashCopy allow this without stopping your primary applications.
- If a disaster unfortunately happened at your primary site, would you feel comfortable to restart from secondary disk, without having saved all that environment? Again, FlashCopy provides you this capability.
- Once in a disaster situation, supposing that your primary site will never been available again, the secondary site is now your only production site. You need to have the capability to execute at least a daily backup of your environment while you are rebuilding the XRC environment on another site.

13.2.8 Determine SDM-primary site bandwidth

One of the key elements in the sizing of an XRC configuration is to determine the required bandwidth between the SDM and the primary ESSs. You could use ESCON or FICON channels to connect the sites if the distance allows it, else you have to use channel extenders and telecommunication lines.

Regardless of connection technology used, you have to analyze the workload to determine the peak requirements. The peak period should cover a short duration, so that lower workload levels do not lower the average peak workload. Typically, you should analyze on-line requirements during the 5 minutes peak period, and batch requirements during the 15 minute peak. Taking this approach, you can still benefit from the dynamic workload balancing function in XRC, without really compromising application throughput or response time.

The challenge can be to find the peak intervals, because the period with the highest write activity is not necessarily the period with the highest bandwidth requirement. Typically, the write activity during daytime processing can be high, but this processing uses short write block sizes. The required bandwidth can therefore be higher during batch processing, even if the write rate is lower, because of the larger block sizes.

Tip: You could use report generator products to identify peaks of write I/Os and data transfer based on information in SMF record type 74. Subtype 1 (device activity) contains information about I/O rates and connect times, and subtype 5 (cache activity) contains information about the number of reads and number of writes per device. Make sure that the known peak days, like the end of month processing, are included in the analysis.

Important: Primary-secondary site connectivity is a key point in an XRC solution and must be carefully planned. IBM Global Services is available to assist you in this process.

In the following sections we give you a general view of all the options available.

ESCON channels

The ESCON architecture allows for a maximum transmission distance of a single link of 3 km with 50 micron multi mode (MM) fiber. With the ESCON extended distance feature (XDF) feature, which uses laser transmitters and single mode (SM) fiber, this maximum distance can be extended to 20 km.

ESCON directors are switches for ESCON channels, and can also be used as repeaters to extend the maximum distance. Up to two of them can be cascaded, resulting in a configuration with three connections: CEC to ESCD; ESCD to ESCD; and ESCD to ESS. Each of these can be 3 km apart, or 20 km if using XDF.

A local ESCON connection can have a throughput of up to 17 MB/s, but you must consider that the throughput is a function of distance, and drops with increased distances.

FICON channels

The Fiber Connection (FICON) is a high bandwidth channel type that offers more efficient and faster data transfer. Currently, installed single mode and multi mode fiber cables can still be used.

Without repeaters, link distances are up to 10 km for 9-micron single mode fiber, and up to 300 meters and 120 meters for 50 or 62.5 micron multi mode fiber respectively. Each FICON link is capable of supporting the same capacity as up to eight ESCON channels.

FICON channels have link rates of 100 MB/s (or 200MB/s with FICON Express), using full-duplex data transfer (it is possible to send data in both directions simultaneously; compared to ESCON, which is a half-duplex protocol). These channels can also implement multiple concurrent I/Os on a single channel.

Data rate drop does not occur with the FICON implementation at the shorter distances as with ESCON. Therefore, FICON is more appropriate for longer distances up to 100 km.

Using DWDMs

A DWDM is a high speed, high capacity, scalable fiber optic data transport system that uses Dense Wavelength Division Multiplexing (DWDM) technology.

DWDM is a technique to transmit several independent bit streams over a single fiber link. It is an approach to opening up the conventional optical fiber bandwidth by breaking it up into many channels, each at a different optical wavelength.

Each wavelength can carry a signal at any bit rate less than the upper limit defined by the electronics, typically up to 2,5 Gbps (Gigabits per second), although latest technologies can reach up to 10 Gbps.

DWDMs provide protocol independence, and a mixture of the following fiber optic data link types may be supported:

- ESCON (both multi mode and XDF single mode links)
- FICON (both FICON native (FC) and FICON bridge (FCV) single mode and multi mode links)
- Fibre Channel
- 9037 Sysplex Timer
- Parallel Sysplex Coupling Links
- Asynchronous Transfer Mode (ATM) (both 155 and 622 Mbits/sec)
- Fiber Distributed Data Interchange (FDDI)
- Fast Ethernet
- Gigabit Ethernet (GbE)

Using a DWDM in an ESS XRC configuration can serve two purposes: to extend the ESCON/FICON distance between primary ESS and the SDM at secondary site, and to reduce the number of fiber cables between the sites.

Using channel extenders

For XRC, the way to achieve connections beyond ESCON or FICON distances is by using channel extenders. IBM does not market this technology directly, but other suppliers are available. The exact configuration and limitations may vary among solutions provided by different suppliers. The channel extender vendor should be contacted for requirements and capabilities.

The distances that can be implemented with channel extenders are only limited by the telecommunication and the channel extender technology capabilities and limitations.

13.2.9 Select Utility device mode

ESS can use either fixed or floating utility devices, as described in 8.2.7, “Utility devices” on page 121. Our recommendation is to use fixed utility devices.

> **Tip:** with ESS you can define small custom volumes. You could define a 1 cylinder volume and add this volume to the SC session with XRCUTL parameter in the XADDPAIR command. Then you know that there will always be one device available to be used as a utility device with no activity and no reserves.

13.2.10 Understand XADDPAIR processing

Unless you specify the NOCOPY parameter, the XADDPAIR command will instruct SDM to make the initial copy of the primary volumes to the secondary volumes. XADDPAIR is also used to resynchronize suspended volumes.

XRC allows updates to the primary volumes during this synchronization or resynchronization process, and copies the updates to the secondary volumes.

> **Note:** You can use the QUICKCOPY option on the XADDPAIR command to copy only allocated tracks from the primary volume to the secondary. However, as a reserve is issued against the primary volume during initial processing, this option should not be used in a channel extender environment, as if the links fail the reserve may not be released, thus impacting the applications.

SDM does monitor the update and XADDPAIR activity, and attempts to throttle the XADDPAIR activity such that the update activity is minimally impacted. Even though this control is present, it is important to monitor the system activity during XADDPAIR times, and reducing the SCSYNCH and SYNCH parameters may be required. See 13.3.8, “XSET command” on page 219.

13.3 Define XRC management

XRC can be managed either through the set of z/OS (or OS/390) TSO commands, or the application programming interface (API), or through the Remote Copy Management Facility (RCMF) offering.

In this section we give you a brief description of the TSO commands used to start, control, and recover XRC sessions; explain what they are used for; and offer some recommendations regarding use of different parameters.

Remember that like most TSO commands, those commands are deferred requests. The initial message that you receive reports on the validity of the command syntax and that the system has accepted the command for processing. When you specify a userid (MSGROUTEID) in the command, only that ID receives the command processing messages. If this user ID is not logged on, the messages are saved in the TSO BRODCAST data set.

For a more detailed description of the XRC TSO commands, and the required and optional parameters, see *z/OS DFSMS Advanced Copy Services,* SC35-0428.

Including XRC TSO commands in procedures

You can include XRC TSO commands in procedures such as REXX execs. Within the procedure, use the MSGROUTID parameter to specify the TSO user ID to which you want XRC to send messages.

XRC routes operational messages to the user ID that issued the command. In order to have these messages appear in the syslog, ensure that the SYS1.PARMLIB CONSOLxx member *permits write to programmer* on the ROUTCODE parameter.

Remember that when you issue XEND, XDELPAIR, and XSUSPEND commands, XRC prompts you to confirm the requests. If those commands are invoked through an exec written in REXX, you should either specify the NOVERIFY option, or place your response to the prompt in the data stack prior to issuing the command. In a batch environment, you can stack the response.

13.3.1 Issuing XRC TSO commands

The status of the SDM XRC session determines which commands you can issue. Table 13-3 lists the XRC TSO commands and indicates for which sessions state you can issue them.

Table 13-3 XRC TSO commands

TSO command	This command can be issued when XRC session is						
	Active	Inactive	Suspended	Recovery	Coupled Active	Coupled Inactive	Master
XADDPAIR	Yes	No	No	No	Yes	No	No
XADVANCE	Yes	Yes	Yes	No	Yes	Yes	No
XCOUPLE ADD	Yes	No	No	No	No	No	No
XCOUPLE DELETE	No	No	No	No	Yes	No	No
XCOUPLE PURGE	No	No	No	No	No	Yes	No
XCOUPLE RELEASE	No	No	No	No	No	No	Yes
XDELPAIR	Yes	No	No	No	Yes	No	No
XEND	Yes	No	No	No	Yes (1)	No	Yes (2)
XQUERY	Yes	No	No	Yes	Yes	No	Yes

TSO command	This command can be issued when XRC session is						
	Active	Inactive	Suspended	Recovery	Coupled Active	Coupled Inactive	Master
XQUERY MASTER	No	No	No	No	No	No	Yes
XRECOVER	No	Yes	Yes	Yes	No	Yes	No
XSET	Yes	No	No	No	Yes	No	No
XSTART	No	Yes	Yes	No	No	Yes	No
XSUSPEND (Volume)	Yes	No	No	No	Yes	No	Yes (2)
XSUSPEND (Session)	Yes	No	No	No	Yes (1)	No	Yes (2)

Notes:
1. You may issue the XEND and the XSUSPEND commands to a coupled, active session after all volumes have been suspended (AVS) or there are no volumes in the session (NOV).
2. The XEND, XSUSPEND (VOLUME) and XSUSPEND (SESSION) commands issued to a master session will cause the specified command to be coordinated on the member coupled sessions.

13.3.2 XSTART command

The XSTART command validates and opens the state, control, and journal data sets with names that match the session ID, and starts an XRC session by creating the ANTASnnn address space.

The XSTART command is also used to restart a suspended session:

```
XSTART SBSESS SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

In this case, the XRC session is started with the name SBSESS.

The SESSIONTYPE parameter can either be XRC or MIGRATE. For the purpose of this book, we use XRC, which identifies to SDM that it operates in a disaster recovery solution and that it has to make use of the journal and control data sets to guarantee secondary volume consistency.

ERRORLEVEL is an important parameter because it specifies how the XRC session responds when an error occurs that causes the session to become unable to process a volume. This may impact the data consistency on secondary volumes. The ERRORLEVEL parameter specified with XSTART is the default for all volumes in the session, but can be overridden by specifications on the XADDPAIR command. For additional discussion of the ERRORLEVEL parameter, see 13.2.4, “The ERRORLEVEL parameter” on page 206.

13.3.3 XADDPAIR command

The XADDPAIR command can be used for:

- Defining new volume pairs to an XRC session. XRC makes an initial copy of the primary volume to the secondary volume unless you specify the *NOCOPY* option.
- Adding volume pairs that were previously suspended. XRC automatically resynchronizes suspended pairs and processes them ahead of new volume pairs. Instead of specifying the volume pairs by volser, the *SUSPENDED* option can be used to instruct XRC to add all suspended volumes into the session and schedule them for resynchronization.

- Adding a utility device to a storage control session. You specify a utility volume by using *XRCUTL* instead of a secondary volser. XRC does not copy data from a utility volume, and XRC ignores *NOCOPY* and `ERRORLEVEL` options.

A maximum of 50 pairs can be specified on a single XADDPAIR command. XRC allows the primary systems to update the primary volumes while the XADDPAIR copy operation is processing.

Example 13-1 shows various examples of XADDPAIR commands.

Example 13-1 XADDPAIR command syntax examples by option

```
1. XADDPAIR SBSESS VOLUME(PVOL01,SVOL01) DONOTBLOCK
2. XADDPAIR SBSESS VOLUME(PVOL03,SVOL03) ERRORLEVEL(VOLUME)
3. XADDPAIR SBSESS VOLUME(PMIN01,XRCUTL) SCSESSION(AA)
4. XADDPAIR SBSESS SUSPENDED
5. XADDPAIR SBSESS VOLUME(PVOL03,SVOL03) NOCOPY
6. XADDPAIR SBSESS VOLUME(PVOL04,SVOL04) QUICKCOPY
```

The options listed in Example 13-1 are explained below:

- **Example 1**: *DONOTBLOCK* option:

  If the SDM is unable to drain updates from the ESS cache rapidly enough, then the ESS microcode will by default slow down write activity from the primary systems to the primary volumes with high write I/O activity. The *DONOTBLOCK parameter* specifies that XRC should not block write I/Os issued by the primary systems to the PVOL01 primary volume.

- **Example 2**: *ERRORLEVEL* option:

  This parameter, specified in this case with the XADDPAIR command, overrides the default value specified in the XSTART command. See 13.2.4, “The ERRORLEVEL parameter” on page 206 for more information about the ERRORLEVEL option.

- **Example 3**: *XRCUTL* option, *SCSESSION* option:

  XRCUTIL specifies that PMIN01 is a utility device, and PMIN01 is not copied to a secondary volume.

  SCSESSION option instructs SDM to assign the volumes (in this case a utility volume) to a specific storage control session (AA). You can assign one or two alphabetic characters for the storage control session value, but it cannot be changed for suspended volumes.

- **Example 4**: *SUSPENDED* option:

  It specifies that SDM should add all suspended volumes back into the session (SBSESS) and schedule them for resynchronization. Suspended volumes keep the same characteristics that existed when the suspension occurred, and therefore you should not use this option to add volumes back into a session if you want to change any of the volume pair characteristics (such as error level, storage control session assignment, or device blocking).

  Note: The SUSPENDED option is mutually exclusive with VOLUME, DONOTBLOCK, NOCOPY, SCSESSION, FULLCOPY, QUICKCOPY, and ERRORLEVEL options.

- **Example 5**: *NOCOPY* option:

  The *NOCOPY* option bypasses the initial primary to secondary volume copy; SDM only starts mirroring updates to the primary volume after the volume is added.

Attention: Because of potential data integrity exposures, the NOCOPY option should be used carefully.

- **Example 6**: *QUICKCOPY option:*

  The *QUICKCOPY* option specifies that only *allocated* tracks will be copied onto the secondary volume when the pair is being established. XRC ignores QUICKCOPY when establishing utility volumes and when resynchronizing volumes.

  Note: To ensure data integrity, the initial processing for QUICKCOPY must issue a reserve and then a release for the primary volume during the initial phase of the synchronization process. This might impact your production workload.

13.3.4 XQUERY command

The XQUERY command is the primary tool to request information about the XRC environment. It provides status at the session and volume level for all primary and secondary volume pairs.

Parameters are available to customize the reports, both regarding the type of information and the level of detail. Here are some examples of reports available by selecting XQUERY command parameter options:

- XRC session statistics report
- XRC volume pair reporting
- XRC session default setting report
- XRC related storage control statistics report
- XRC related configuration report

The data consistent time in the XQUERY report is the consistency time for volumes in the session at the time the report is given. The timestamp variable specifies the UTC time from the application system. XRC generates this time from the updates received from the application system. Data on the secondary volumes was last consistent up to this timestamp.

XQUERY also reports a delay time. This is the current delay between the last application I/O to a primary LCU, and the update made by the data mover to a secondary volume. This field provides an approximation of the time delay between the primary and secondary volumes.

Output from XQUERY directed to the system log has `ANTQxxxx` message identifiers, and *z/OS MVS System Messages, Volume 1 (ABA-AOM),* SA22-7631 contains more explanations about XQUERY output.

XQUERY output can be directed to a sequential data set, and our recommendation is that you have the command automated, and that you direct the output to a data set for further processing.

The sample command in Example 13-2 will direct the XQUERY output to data set XQUERY.OUT, and this output will be appended to this existing data set. Only the volumes from session SBSESS, which match the serial number prefix PVOL will be reported.

Example 13-2 XQUERY command syntax example

```
XQUERY SBSESS VOLUME(PVOL*) dataset(XQUERY.OUT) DISP(MOD)
```

XQUERY MASTER in a coupled environment

Issue the XQUERY MASTER command to a master session to display information about coupled XRC sessions associated with the master session, including the XRC session name, session status, command pending status, journal delta time, RCV/ADV delta time, master session status, master session recoverable time, and master pending command.

Example 13-3 shows an example of a command where the master session name is *HOSPITAL*, and the high level qualifier of the master session control data set is *SYS2*.

Example 13-3 XQUERY MASTER example

```
XQUERY HOSPITAL MASTER MHLQ(SYS2) dataset(XQUERY.OUTPUT2) DISP(OLD)
```

13.3.5 XDELPAIR command

The XDELPAIR command directs the XRC session to stop processing a primary volume and its secondary volume. For each volume, the XDELPAIR command returns a timestamp indicating that the data on the secondary volume is consistent with the data that was on the primary at that time.

You can delete any XRC volume pair with the XDELPAIR command, regardless of the status of the pair. Volume pairs in *pending*, *duplex*, *copy, seqcheck*, and *suspend* status all become inactive when deleted, and are no longer eligible for recovery.

You may delete up to 100 volume pairs in a single XDELPAIR command. Note that XRC does not process any volumes if there is a command syntax error, or if any of the volumes in the list are not part of the XRC session.

Example 13-4 shows various examples of XDELPAIR commands.

Example 13-4 XDELPAIR command syntax examples by option

```
1. XDELPAIR SBSESS VOLUME(PVOL01 PVOL02 PVOL03) DRAIN
2. XDELPAIR SBSESS VOLUME(PVOL02) CANCEL
3. XDELPAIR SBSESS VOLUME(ALL)
```

The options listed in Example 13-4 are explained below:

- **Example 1**: As an optional parameter you can specify when the pairs are going to be deleted by specifying *IMMEDIATE*, *DRAIN*, or *ATTIME*:

  IMMEDIATE: This is the default. It specifies that SDM immediately deletes the volume pair or pairs when the current Consistency Group has been applied. The secondary volumes are consistent up to the UTC timestamp that is reported on the successful completion of the command. This is the only option that can be used for suspended volumes.

  DRAIN ensures that all timestamped record updates that the SDM has not yet written to the secondary storage control are drained from the primary storage control, and applied to the secondary volume or volumes. All secondary volumes are consistent up to the UTC timestamp that is reported by successful completion of the command.

  ATTIME specifies to SDM to delete the specified volume pairs after applying all updates to the secondary volumes up to the specified time. XRC will not apply any updates beyond the specified UTC timestamp to the secondary volumes. All affected secondary volumes are consistent up to the UTC timestamp that is reported on successful completion of the command.

- **Example 2**: *CANCEL:*

  CANCEL specifies to SDM to cancel the previous XDELPAIR command with an ATTIME or DRAIN request. The cancel option is only valid when there is a pending XDELPAIR command.

- **Example 3**: *VOLUME(ALL):*

  VOLUME(ALL) specifies to SDM to delete all eligible volumes in the session, including utility volumes. The XRC session will still be active.

13.3.6 XEND command

Use the *XEND* command to stop all XRC activity to active volumes and to end the XRC session. The *XEND* command returns a timestamp indicating that the data on the secondary volumes of all active pairs is consistent with the corresponding data that was on the primary volumes at the specified time.

Note: The *XEND* command has the same optional parameters as *XDELPAIR*. See 13.3.5, “XDELPAIR command” on page 217 for further description of those parameters.

In a coupled environment, you can issue the *XEND command* to a master session to have all sessions end with their volumes consistent to the same time.

The *XEND* command in Example 13-5 ends session SBSESS and all its associated volumes.

Example 13-5 XEND command syntax example

```
XEND SBSESS
```

13.3.7 XSUSPEND command

Use the *XSUSPEND* command either to suspend an XRC session, or to suspend one or more volume pairs from the XRC session. In a coupled environment, you may either suspend all XRC sessions coupled to the master session, or suspend all volume pairs from sessions that are coupled to the master session.

When an XRC session is suspended, the ANTASnnn address space is ended. The system data mover is no longer active and the volume pairs are in suspended state.

When all volume pairs in an XRC session are suspended, the SDM remains active. As ESS supports hardware bit maps, the ESS will continue to record changed tracks, and there will be no impact to the primary systems. XRC will use this bit map to resynchronize the volumes when the session is restarted, and volumes added back to the session with XADDPAIR command.

Example 13-6 shows various examples of *XSUSPEND* commands.

Example 13-6 XSUSPEND command syntax examples

```
1. XSUSPEND SBSESS VOLUME(PVOL03) DRAIN
2. XSUSPEND SBSESS VOLUME(ALL)
3. XSUSPEND SBSESS TIMEOUT(01:15:00)
```

The options listed in Example 13-6 are explained below:

- **Example 1**: The *VOLUME* parameter indicates that this is a volume suspension. The *VOLUME* parameter is mutually exclusive with the *TIMEOUT* parameter. The *IMMEDIATE*, *ATTIME*, and *DRAIN* keywords allow you to control when the suspension occurs. For more information about those parameters; see 13.3.5, “XDELPAIR command” on page 217.
- **Example 2**: When *VOLUME(ALL)* is specified, XRC suspends all volumes in the session, but does not suspend utility devices.
- **Example 3**: *TIMEOUT* indicates that the entire XRC session should be suspended, and the ANTASnnn address space ended. The value specified in the *TIMEOUT* parameter specifies the maximum time that the storage control can wait for the XRC session to restart. The *TIMEOUT* parameter is mutually exclusive with the *VOLUME* parameter.

13.3.8 XSET command

XSET command is used by the storage administrator to control and dynamically change some SDM tuning values.

For a more detailed description of the XSET command, and its required and optional parameters, see *z/OS DFSMS Advanced Copy Services,* SC35-0428.

13.3.9 XRECOVER command

The *XRECOVER* command will bring data on the secondary volumes to a consistent, recoverable state where they are equivalent to the primary volumes, as of the given timestamp.

If the XRC session has either been suspended or ended normally by a command, the *XRECOVER* command reports the timestamp up to which all data is consistent, and it will also automatically generate an *XQUERY storage control* report. The report includes the last applied timestamp that defines the recovered, consistent state for all volumes.

If a session is active when you issue the *XRECOVER* command, SDM rejects the command. You must first issue an *XSUSPEND* or *XEND command* for the session, and then make all primary volumes off-line to the SDM system.

You can also issue the *XRECOVER* command to recover a group of interlocked coupled sessions to the same consistency time. In order to provide master session consistency, CXRC applies updates to volumes on the sessions that are behind to allow them to advance to the forward session. CXRC will forward-recover volumes that are suspended because of an error, and that have the necessary journal data available. It will not forward-recover volumes that are suspended because of an *XSUSPEND command.*

An example of a *XRECOVER command* for bringing the data of session SBSESS to a consistent and recoverable state is:

```
XRECOVER SBSESS
```

You have to specify the *session_name* that was previously specified on the *XSTART command.*

13.3.10 XADVANCE command

The *XADVANCE* command differs from the *XRECOVER command* in that it obtains a consistent set of secondary volumes without updating the secondary volume VOLSERs.

The *XADVANCE* command requires all volumes in the session to be suspended. It may be used with an active or an inactive session. You cannot use this command when the session is active on another system, or if the session has any non-suspended volumes.

Use the *XADVANCE* command to create a time-consistent backup of the secondary volumes to a time that is consistent with the volumes of other sessions that are coupled to the master session. This is the same time to which an XRECOVER command would recover the session.

The *XADVANCE* command produces an XQUERY report that displays YES in the ADV column to indicate if each volume in the session was eligible to have updates applied. A NO in the column indicates that the volume was not eligible to have updates applied. The report also displays NO for volumes that have not reached duplex, that were suspended through a command, or that were suspended at a time before the consistency time that exists as of the beginning of the *XADVANCE* operation.

Issuing an *XADVANCE* command to a master session automatically generates an XQUERY MASTER report for the coupled sessions.

An example of an *XADVANCE* command for session UMC is:

```
XADVANCE UMC HLQ(SYS1) MSGROUTEID(OPER3)
```

SYS1 is the high level qualifier of the journal, state, and control data sets that were used with this session.

13.3.11 XCOUPLE command

Use the *XCOUPLE command* to add XRC sessions to, or delete XRC sessions from, a master session.

Your storage administrator can also use this command to manage the status of the master session.

For a more detailed description of the XCOUPLE command, and its required and optional parameters, see *z/OS DFSMS Advanced Copy Services,* SC35-0428.

13.3.12 ANTRQST application programming interface

The ANTRQST macro provides an application program call to the z/OS SDM's application programming interface (API). This macro allows you to call XRC, PPRC, FlashCopy, and SnapShot copy functions. You may need to ensure that necessary address spaces such as ANTAS000, ANTASnnn, or ANTMAIN are active when you execute the macro.

The ANTRQST macro is described in detail in *z/OS DFSMSdfp Advanced Services*, SC26-7400.

13.4 Define XRC automation

An XRC environment may involve hundreds if not thousands of volumes. We recommend that you keep the configuration as simple and symmetric as possible, for example, a one-to-one correspondence between primary and secondary LCUs.

Nevertheless, you should implement some form of automation. Automation for XRC may range from simple JCL-/REXX generation code, to fully automated monitoring and recovery actions.

13.4.1 JCL-/REXX generation

JCL-/REXX generation code is recommended to minimize errors when performing XRC operations. Code can be automatically generated to XADDPAIRs, to XSUSPEND, and so on. This will ensure that all eligible volumes are included in all lists. Even if major operations such as migration and recovery are going to be under manual control, errors can be minimized by automatic generation of jobs to be executed.

13.4.2 Error recovery

Many systems management software packages allows error messages to be intercepted, filtered and directed to an automation package. For example, if a volume drops out of duplex status, an operator could be notified, or an automatic XADDPAIR may be initiated.

Automation code may also be built to automate the full XRC recovery process in the event of a disaster. For Disaster Recovery it is important not to rely on personnel who may not be available to perform the recovery actions.

13.4.3 GDPS/XRC

GDPS/XRC, an industry leading e-business availability solution available through IBM Global Services, is a multi-site solution that is designed to provide the capability to manage the remote copy configuration and storage subsystems, automate Parallel Sysplex operational tasks, and perform failure recovery from a single point of control, thereby helping to improve application availability.

Geographically Dispersed Parallel Sysplex (GDPS) is an integrated solution offering that manages all aspects of switching computer operation from one site to another, planned or unplanned. In a GDPS/XRC configuration, the SDM is placed outside the production sysplex, and normally it is located at the recovery site.

One subset of the GDPS solution is the Remote Copy Management Facility (RCMF) offering. RCMF is an automated disk subsystem and remote copy management facility, with a high level user interface. This interface is implemented in the form of ISPF like displays, and virtually eliminates the tedious and time consuming work with TSO commands. Managing XRC with RCMF could be the first step of a full GDPS implementation.

For more information on GDPS, refer to these sites:

http://www.ibm.com/servers/eserver/zseries/library/whitepapers/gf225114.html
http://www.ibm.com/services/us/index.wss/so/its/a1000189

13.5 Implement the solution

The following chapter provide rough guidelines for the solution implementation.

13.5.1 Allocate the Journal data set

The SDM uses the journal data set(s) as temporary storage, to *harden* a Consistency Group until all of its data is committed to the secondary devices.

Journal data sets are allocated in pairs (odd/even). The even/odd pair can be allocated on the same volume because they will never have to compete with each other. SDM assigns a separate internal subtask for each pair of allocated journal data sets, and thereby allows for parallel transfer of CGs.

You must allocate journals data set using the following data set name:

`SYS1.XCOPY.session_id.JRNL`*nn* or `hlq.XCOPY.session_id.JRNL`*nn*

In this name, **nn** has to be **01**,**02**.... up to **16**. Numbers must be in sequence: if you define journal data sets **01**,**02**,and **04**, for example, XRC ignores **04** because you did not define data set **03**.

A maximum of 16 journal data sets can be specified for an XRC session, but at least 2 must be specified. Define an even number of journal data sets, preferably four to eight for smaller XRC configurations (for up to about 750 volumes), and define from eight to 16 journal data sets for large XRC configurations.

Each journal data set must be a fixed blocked sequential data set with the following attributes: DCB=(RECFM=FB,LRECL=7680,BLKSIZE=7680,DSORG=PS)

Here are some other recommended specifications:

- Allocate journal data sets as single extent data sets.
- Minimize non-XRC activity to journal data set volumes.
- Do not put journal data sets on the same volumes as the control and state data set.
- Allocate the journal data sets on high performance volumes.
- Do not put journal data sets behind storage controllers that also process secondary volumes.
- Define the journal data sets as striped data sets, for maximum performances.
- Spread the striped journal data sets across many disk volumes, possible in multiple ESSs/LSSs.

The allocations shown in Figure 13-2 will optimize journal performance for 2, 4, and 8 journal data sets.

<table>
<tr><th># of journal datasets</th><th colspan="4">Striped dataset</th><th colspan="2">Non striped dataset</th></tr>
<tr><td>2</td><td colspan="2">Vol1
JRNL01
JRNL02</td><td colspan="2">Vol2
JRNL01
JRNL02</td><td colspan="2">Vol1
JRNL01
JRNL02</td></tr>
<tr><td>4</td><td>Vol1
JRNL01
JRNL02</td><td>Vol2
JRNL01
JRNL02</td><td>Vol3
JRNL03
JRNL04</td><td>Vol4
JRNL03
JRNL04</td><td>Vol1
JRNL01
JRNL02</td><td>Vol2
JRNL03
JRNL04</td></tr>
<tr><td rowspan="2">8</td><td>Vol1
JRNL01
JRNL02</td><td>Vol2
JRNL01
JRNL02</td><td>Vol3
JRNL03
JRNL04</td><td>Vol4
JRNL03
JRNL04</td><td>Vol1
JRNL01
JRNL02</td><td>Vol2
JRNL03
JRNL04</td></tr>
<tr><td>Vol5
JRNL05
JRNL06</td><td>Vol6
JRNL05
JRNL06</td><td>Vol7
JRNL07
JRNL08</td><td>Vol8
JRNL07
JRNL08</td><td>Vol3
JRNL05
JRNL06</td><td>Vol4
JRNL07
JRNL08</td></tr>
</table>

Figure 13-2 Allocation of journal data sets

Size of journal data sets

The aggregate, minimum size for the allocated journal data sets should be at least twice the size of the total SDM buffer.

SDM, by default, allocates 576 buffers per each storage control session, so if you have 20 XRC storage control sessions running in an SDM, you would have 11520 full track buffers available.

This represents the equivalent of 768 cylinders, thus 1536 cylinders should be allocated.

If eight journal data sets are allocated, each with two stripes, then allocate a minimum of 192 cylinders per data set (96 cylinders per stripe).

Tip: The minimum recommended allocation is 100 cylinders for each journal data set, and remember that all journals data sets should be defined as being of equal size.

13.5.2 Allocate the control data set

The control data set contains control records indicating:

- The last set of data that was written to the secondary volumes, that is, the last Consistency Group.

- The amount of unwritten data that exists in the journal (this means the amount of data that has been written to the journal as CGs, but has not yet been completely written to the secondary volumes).
- The location of the unwritten data in the journal data sets.

You must allocate control data set using the following data set name:

```
SYS1.XCOPY.session_id.CONTROL or hlq.XCOPY.session_id.CONTROL
```

We recommend sequential allocation, since this provides much higher performance than the PDSE allocation. Use the following attributes:

```
DCB=(RECFM=FB,LRECL=15360,BLKSIZE=15360,DSORG=PS)
```

The data set must be allocated without defining secondary extents, and you should place the data set on a different volume from the journal data sets.

You should allocate 2 tracks for the sequential control data set.

13.5.3 Allocate the state data set

The state data set contains the status of the XRC session and the status of the volumes in the session. This data set also contains bit maps of changed tracks for primary volumes. The state data set is updated whenever an XADDPAIR, XDELPAIR, XSUSPEND, XRECOVER or XEND command is issued, or whenever a volume state changes.

You should use the following name for the state data set:

```
SYS1.XCOPY.session_id.STATE or hlq.XCOPY.session_id.STATE
```

The state data set is a PDSE, and you should use the following attributes:

```
DCB=(RECFM=FB,LRECL=4096,BLKSIZE=4096,DSORG=PO),DSNTYPE=LIBRARY
```

Here is the recommended minimum allocation:

- Allocate ten tracks for each XRC storage control session.
- Allocate one additional track for each 3390-3 volume pair in the session.
- Allocate three additional tracks for each 3390-9 volume pair in the session.

13.5.4 Allocate the master data set

> **Attention:** The master data set is required in a Coupled eXtented Remote Copy (CXRC) environment only. You do not need to allocate this data set for a non-coupled XRC implementation.

All of the XRC sessions coupled in a master session continuously write to and read from the master data set. This access allows communication between hosts as data is copied from the primary site to the recovery site. These are the rules that must be considered:

- Catalog the master data set and make it accessible to each host system that processes a coupled session, as well as to the system that processes the XRECOVER command. Place the master data set in a user catalog that contains only entries for the master data set.
- The master data set can reside on the same volume as the state and control data sets, but not on the same volume that contains the journal data sets.

- The master data set name can be in one of the following forms:

  `SYS1.XCOPY.msession_id.MASTER or mhlq.XCOPY.msession_id.MASTER`

  Where:

  msession_id Specifies the same name that you issued with the MSESSION parameter of the XCOUPLE ADD command. The name can be up to eight characters long. The name you specify for the msession_id cannot be the same as any session_id.

  mhlq Specifies the name of the master data set high-level qualifier that is specified on the MHLQ parameter of the XCOUPLE ADD command. The MHLQ must be one to eight characters, and can be any name acceptable to TSO. SYS1 is the default value.

- Allocate the master data set as physical sequential and not striped, with one cylinder primary space and zero cylinders secondary space as follows:

  `DCB=(RECFM=FB,LRECL=15360,BLKSIZE=15360,DSORG=PS),SPACE=(CYLS,(1,0))`

- Allocate the master data set on a single disk device.
- You must allocate the master data set with contiguous tracks, as the I/O is assumed not to span multiple extents.
- Access to the master data set is controlled using GRS enqueues, and therefore, all systems with coupled SDMs must be in the same GRS configuration.
- The required size of the master data set is one cylinder. The required size of the master data set is fixed, which allows maximum of 14 XRC sessions to be coupled into one master session.
- The master data set must be pre-allocated before use any XCOUPLE command.

13.5.5 Implement XRC commands and procedures

Once you defined the XRC environment and all the required resource are available, you need to build all the commands and procedure required to run the XRC solution.

If you choose not to adopt any automation tool, you can define all the XRC TSO command in a PDS data sets. Table 13-4 contains an example of TSO commands that are required.

Table 13-4 Example of TSO command required to run an XRC solution

Member	Contents
XSTART	Describe the XRC session and starts it. `XSTART sess ERRORLEVEL(SESSION) SESSIONTYPE(XRC)`
XADDPAIR	Add all the required primary-secondary pairs to the session. One XADDPAIR command for each pair. `XADDPAIR sess VOL(primary,secondry) DONOTBLOCK`
XDELPAIR	Remove all the primary-secondary pairs from the session. `XDELPAIR sess VOLUME(primary/ALL) DRAIN`
XSUSPSES	Suspend the session and close the SDM `XSUSPEND sess ATTIME(xxxxxxxx)`
XSUSPVOL	Suspend all volume in the session, but SDM remains active `XSUSPEND sess VOLUME(ALL) ATTIME(xxxxxxxxx)`
XRESYNCH	Add all volumes to the session and resync secondaries with primaries `XADDPAIR sess SUSPENDED`

Member	Contents
XEND	Terminate XRC session and close SDM. `XEND sess`
XADVANCE	Bring all secondary volumes at the latest consistency point. `XADVANCE sess`
XRECOVER	Recover all secondary volumes to bring their corresponding primary volumes label. `XRECOVER sess`

Moreover, you have to define the JCL and procedures required to create a disaster recovery environment using FlashCopy functions. For this, refer to 13.8, “Using XRC with FlashCopy” on page 232.

13.5.6 Implement XRC security

There are two ways to protect access to XRC commands: You can put the commands in a RACF-protected library, or define resource profiles in the RACF FACILITY class and restrict access to those profiles.

Placing TSO commands in a RACF protected library

Place the TSO commands in a RACF-protected library to restrict XRC TSO commands to authorized storage administrators.

To RACF-protect XRC commands, perform the following steps:

1. Issue the following RDEFINE command for each XRC command, and for each command abbreviation that you want defined to RACF:

   ```
   RDEFINE PROGRAM cmd ADDMEM('SYS1.CMDLIB'/vol/NOPADCHK) -
   UACC(NONE)
   ```

   The following terms apply to the above example:

   cmd: Defines the XRC TSO command name or an abbreviation of a command. Issue a separate RDEFINE command for each of the XRC commands, and any command abbreviations you plan to use. Examples of XRC command abbreviations are XADD, XDEL, and XRCV. RACF can only perform checking on commands and abbreviations that are defined to it.

   vol: Defines the name of the volume that contains the SYS1.CMDLIB data set.

2. Issue the PERMIT command for all commands and authorized XRC TSO command users as follows:

   ```
   PERMIT cmd CLASS(PROGRAM) ID(name) ACCESS(READ)
   ```

   The following terms apply to the above example:

   cmd: Defines the XRC TSO command name, or an abbreviation of a command.

   name: Defines the user ID receiving RACF access authority for that command name.

3. Issue the SETROPTS command from a user ID that has the appropriate authority:

   ```
   SETROPTS CLASSACT(PROGRAM) WHEN(PROGRAM) REFRESH
   ```

Defining resource profiles in the RACF FACILITY class

You can limit the use of XRC commands by defining resource profiles in the RACF FACILITY class and restricting access to those profiles. To use a protected command, you need read access authority to the applicable profile.

All XRC commands can be restricted by using the FACILITY class profile **STGADMIN.ANT.XRC.COMMANDS**.

Use of the XQUERY command can also be authorized in the FACILITY class profile **STGADMIN.ANT.XRC.XQUERY**. XRC first checks **STGADMIN.ANT.XRC.COMMANDS** for authorization, and if authorization is not permitted with this profile, XRC will check **STGADMIN.ANT.XRC.XQUERY**.

The following examples activate the RACF FACILITY class, define the profile for the XRC commands, and give user STGADMIN authority to use this profile:

1. Activate the RACF FACILITY class:

```
SETROPTS CLASSACT(FACILITY)
```

2. Define the profile for XRC commands, and authorize user STGADMIN to use this profile:

```
RDEFINE FACILITY STGADMIN.ANT.XRC.COMMANDS UACC(NONE)
PERMIT STGADMIN.ANT.XRC.COMMANDS CLASS(FACILITY) -
ID(STGADMIN) ACCESS(READ)
```

Restricting access to secondary volumes

XRC requires that the primary and secondary volumes have different volume serial numbers because they both must be on-line to the SDM system.

Non-SDM applications can access secondary volumes by using a different catalog or by explicit unit and volser reference.

Important: We absolutely recommend that you prevent access to secondary volumes from all non-SDM applications because:

- Write operations to the secondary other than by the SDM produce data integrity problems.
- Read activity to secondary volumes may update the last reference date for the data, and may experience various error conditions due to the way XRC handles secondary I/O functions.

Tip: One way to prevent access to the secondary volumes is to define the volumes to a storage group with the disable state (DISALL). SDM will not be affected by this status.

13.5.7 XRC testing

In this section we provide a summary of the recommended testing to go through before XRC is put into production.

Functional testing

This is to test that all XRC commands execute as expected and with correct return codes.

When performing these tests, you should use XQUERY to produce various reports, both to get familiar with the content, but also to verify that the status of volumes, volume pairs, and session is as expected.

Operational testing

This is to ensure that all required procedures are in place for operating the XRC environment. You should test scenarios that may be used in a production environment, such as suspending sessions, adding utility devices, changing error recovery levels, suspending and resynchronizing volume pairs, and so on.

Some suggested operational scenarios are documented in the publication *z/OS DFSMS Advanced Copy Services,* SC35-0428.

Error injection testing

You may want to simulate various failure situations to verify that your operating procedures work. When simulating a disaster, the most likely scenario is that one or more ESSs, or the links to these, are lost. These potential failure situations are best simulated through, for example, the ESCON director. Individual links may be removed, breaking the connection between the SDM and the primary ESSs. The recovery scenario can then be tested on the secondary site. A *rolling disaster* may be simulated by breaking the connections between the SDM and the primary ESSs at different times.

Automation testing

If you follow our automation recommendations (13.4, “Define XRC automation” on page 220), test the automation code to ensure that it operates correctly, generate complete and useful reports, intercepts error messages, and initiates the required actions.

Walk-through

If you are planning to use XRC for Disaster Recovery or workload migration, you may not have the facilities to test in a *production equivalent* environment. In these cases it can be a useful option to do a walk-through the critical procedures. This can identify problems such as ownership of responsibilities, and access to critical data. If you rely on manual intervention for “go” or “no-go” on Disaster Recovery, then the process must be clearly documented at both primary and recovery sites.

Dress rehearsal

The proof of a disaster recovery test is a full-scale dress rehearsal, preferably one that can be initiated without warning. Because XRC supports SUSPEND/RESUME processing, you can test recovery at a secondary site without disruption to your production environment. Once the test is complete, changes that have been accumulated at the primary site can be propagated to the secondary site, restoring disaster protection.

If it is essential to maintain disaster protection during a disaster recovery test, FlashCopy provides an ideal tool to take a quick backup of data at the secondary site. The disaster test can proceed against the FlashCopy of the secondary devices, while XRC between the primary and secondary devices is resumed.

Once a disaster recovery plan has been successfully validated by a full-scale test, it is important to make sure that documentation is maintained, and that the test is repeated at regular intervals.

13.6 Run the solution

Once the solution is built and tested, it has to be set in production and maintained. In the following paragraphs you can find information which you must be aware of when you run an XRC disaster recovery solution.

13.6.1 HCD reconfiguration restriction

Do not reconfigure a device through the Hardware Configuration Definition (HCD) while that device is a member (primary or secondary) of an XRC pair.

13.6.2 XRC volume format, track, and access mode restrictions

To ensure data integrity, all volumes that are part of XRC copy operations must conform to the following conditions:

- Volumes must have a standard format for record zero (R0). Volumes with R0 data lengths longer than 8 bytes can cause a track format error to remain undetected when the track is formatted in cache.
- Alternate tracks cannot be assigned in the user area. If the secondary address has a track assigned as an alternate track and the track is a user track on the primary address, user data may be overlaid.
- All storage control Define Extent commands must specify normal access authorization mode. Data written to a remote copy primary device while the storage control is in *diagnostic* or *device support* mode is not copied to the XRC secondary volume. It is therefore important to remove volumes from the session before running a utility program like ICKDSF.

13.6.3 ICKDSF and XRC volumes

Recent developments in ESS, XRC, and ICKDSF have eliminated the need to XDELPAIR XRC volumes prior to relabeling DASD volumes. These developments allow for concurrent management and operations, as well as increasing XRC's readiness as a Disaster Recovery solution.

Attention: You must be running on recent ICKDSF and XRC maintenance.

When ICKDSF is used to initialize a volume, the updates will be captured by XRC and not written to the secondary DASD. XRC will update its control blocks so that a subsequent recovery will reflect the new volser of the primary volume.

The V<volsername> member of the State Dataset is not updated until the next time the volume is suspended and XADDPAIR SUSPENDED. The volser in the MVS UCB control block is not updated until the next VARY OFFLINE/ONLINE is performed.

We no longer have to do a XDELPAIR/XADDPAIR full volume copy, but we do still have some operations to *sync up* XRC, MVS, and GDPS.

Here is a procedure to update the VOLSER while XRC is running:

- Vary the primary volume offline during ICKDSF operation.
- Use ICKDSF to relabel the VOLSER (XRC control blocks are now correct).
- Vary the primary volume online after ICKDSF operation (UCB is now correct).
- Use XRC SUSPEND VOLUME followed by an immediate XRC XADDPAIR SUSPENDED to update the state data set's member name (this can be done any time).
- *GDPS only*: Use GDPS CONFIG to update GDPS's policy (again, this can be done at any time).

13.6.4 XRC Performance monitor

The XRC Performance Monitor extends the z/OS commands of the XRC facility. It provides additional information to better manage an XRC environment and help XRC achieve the desired performance (typically measured as the latency in updating the mirrored system) at a low cost. The monitor provides information in real time and also maintains historic records. From a very high level or from a detailed perspective, administrators can view the parameters that contribute to system performance.

Its functionality not only helps determine the optimum XRC configuration, but also helps quickly identify performance bottlenecks. The XRC Performance Monitor is also useful in spotting trends that may lead to future performance problems; thus it can also assist with capacity planning.

Components of the XRC Performance Monitor

The XRC Performance Monitor comprises three separate modules, which are integrated under one user interface:

- The *History Monitor* summarizes how the XRC system was running at a prior point-in-time. At approximately 15-minute intervals, this report displays key performance information in an easy-to-read sorted table. Detailed information can be obtained about each reader and specific volumes. This information, displayed graphically, can help identify volume contention that may have created bottlenecks. This report includes details about the telecommunications link, such as the amount of data transferred, as well as measures of the mirrored site, such as peak delay times and average delay times.
- The *Realtime Monitor* provides a summary of how single or coupled system data movers are running (system data movers update the secondary storage devices).This report queries the Interactive System Productivity Facility (ISPF) to obtain the results. By displaying real time information, the XRC Performance Monitor enables administrators to detect and resolve problems quickly.
- The *Batch Exception Monitor* checks the monitor information at user defined intervals for predefined thresholds. This report generates console messages whenever thresholds are exceeded so administrators can take action, therefore eliminating the need for someone to constantly check the monitor.

For more information, refer to:

http://www-1.ibm.com/servers/storage/support/solutions/bc.html

13.6.5 Reporting on the XRC environment

For an active XRC session, SDM writes an SMF record type 42 subtype 11 whenever the SMF time interval expires. This record contains data about the LCUs participating in the XRC session and XRC statistics that can be used for further reporting to monitor XRC activity. For a complete layout of SMF type 42 records, see the manual, *z/OS V1R3.0 MVS System Management Facilities,* SA22-7630.

13.6.6 XRC diagnostic aids

In this section we discuss the diagnostic tools that can be used in XRC problem situations.

Restarting XRC address space

The XRC control address space, `ANTAS000`, is automatically started during system IPL. `ANTAS000` handles the TSO commands that control XRC.

There may be conditions that prevent the system from starting this address space at IPL time, or to restart it automatically if it is cancelled by the operator. For such cases, you can restart the ANTAS000 address space by submitting the job shown in Example 13-7.

Example 13-7 Restart ANTAS000 address space

```
//STARTANT JOB MSGLEVEL=(1,1),REGION=4096K
//STEP1 EXEC PGM=ANTSTRT
```

Diagnosing XRC with MVS MODIFY command

There are options to the MVS *MODIFY* command that can be used to diagnose, and in some cases resolve XRC problems. The format of the MODIFY command is illustrated in Example 13-8.

Example 13-8 MODIFY command

```
F ANTAS00x,operation optional_operands
```

If **x=1**, the command must be issued from the SDM address space. If **x=0**, the command can be issued from any system attached to the primary ESS that supports XRC (the ANTAS000 address space must be active).

Next we show some of the available operations. All operations mentioned here can be issued from any system (with ANTAS000 address space active) attached to the primary LCU. You have to specify a device address in the primary LCU that you want to address with your command.

- ***LISTSESS***: This operation returns the following information from the LCU:
  - Type of SC-sessions in the LCU
  - Status of the SC-session
  - SC-session number
  - SC-session owner
- ***TERMSESS***: This operation is very powerful and can be used to end active, quiesced, suspended or timed-out XRC or CC sessions for an LCU. TERMSESS provides four kinds of support:
  - To end a single SC session (CC or XRC) by specifying SC session number.
  - To end all active or time-out XRC SC sessions that were last active on the system from which this TERMSESS operation is issued.
  - To end all quiesced or suspended XRC SC sessions, regardless of which system the XRC operation was last active.
  - To end all XRC or CC SC sessions that are in time-out status.
- ***LISTDVCS***: This operation lists all devices that are part of a specific SC session. You have to specify the SC session number, which can be obtained by the LISTSESS operation.

For further description and additional options provided with the MODIFY command to diagnose and repair XRC, see the IBM publication *z/OS DFSMS Advanced Copy Services,* SC35-0428.

Abnormal XRC coupled session status

The IBM publication *z/OS DFSMS Advanced Copy Services,* SC35-0428 contains scenarios for managing the status of a CXRC environment in problem situations. Refer to this manual for a more detailed discussion.

13.7 Database recovery with XRC

Your normal operational requirements will already include procedures for restarting on-line applications in case of a short term outage (a power failure or CEC failure, for instance).

For XRC Disaster Recovery, the same procedure should be used at the secondary site. The same checks for guaranteeing data consistency and restarting your applications must be done at your secondary site, just as they are done today at your application site.

XRC provides data currency across all volumes in the XRC session. In the event of a disaster at the primary site, updates in flight are lost. Nevertheless, recovery can be achieved to an identified consistent point-in-time. The recovery technique does not require database restore followed by forward recovery. Recovery can follow the procedure used at the primary site for a system outage, consistent to the point-in-time given. Any transaction that have been completed after this point-in-time might require investigation and be recreated against the secondary disks after recovery (using XRECOVER) or accounted for in your business process.

13.8 Using XRC with FlashCopy

Using ESSs at the secondary site provides a unique ability to combine FlashCopy functions with XRC functions. For instance, you can use this combination to:

- Create data for data mining applications
- Create data for disaster recovery testing
- Create consistent data for point-in-time backup purposes

The following procedure describes a method to create data for disaster recovery testing using FlashCopy and XRC commands:

1. Issue the *XSUSPEND VOLUME(ALL)* command.
2. Issue *XADVANCE SESSION* command to have all secondary volumes at the same consistency point.
3. Submit DFSMSdss jobs to invoke FlashCopy to copy secondary volumes. The FlashCopy target volumes will be placed offline since they will result in having the same label as the secondary volumes.
4. As soon as the FlashCopy relationship has been established, you can restart the XRC session issuing *XADDPAIR SBSESS SUSPENDED* command, which adds all the suspended volumes back into the session and starts volume resynchronization.
5. As soon as all DFSMSdss jobs from step 3 are complete, use ICKDSF to clip all the FlashCopy target volsers to corresponding XRC primary volsers. Those volumes will continue to be offline to the SDM system.
6. Now you can activate a new LPAR which can restart a copy of your production system from the FlashCopy target. This will be your DR testing system.

14

Geographically Dispersed Parallel Sysplex

This chapter presents the Geographically Dispersed Parallel Sysplex (GDPS) offering. GDPS provides an automated disaster recovery solution for total business continuity in the z/OS and OS/390 environments.

For information on GDPS, refer to the following Web URL:

http://www.ibm.com/servers/eserver/zseries/gdps.html

For additional information on GDPS, contact your IBM representative or e-mail:

gdps@us.ibm.com

White papers on GDPS can be found at:

http://www.ibm.com/servers/storage/support

- *Geographically Dispersed Parallel Sysplex: The Ultimate e-business Availability Solution*, GF22-5114.

We describe GDPS in general in the first part of this chapter and then describe the effects on DB2 beginning with 14.2, “GDPS-Metro Solution” on page 240.

Another solution that requires GDPS, called “three site solution with no data loss” is described in 14.3, “Metro/Global Mirror solution” on page 245.

This chapter covers the following topics:

- GDPS overview
- GDPS-Metro Solution
- Metro/Global Mirror solution

14.1 GDPS overview

GDPS is a multi-site application availability solution that provides the capability to manage the remote copy configuration, automates Parallel Sysplex operational tasks, and performs failure recovery from a single point of control, thereby drastically improving application availability.

GDPS supports all transaction managers (for example, CICS, IMS, Web Sphere) and database managers (for example, DB2, IMS, and VSAM), and is enabled by means of key IBM technologies and architectures:

- Parallel Sysplex
- Tivoli Netview for z/OS and OS/390
- System Automation for OS/390
- IBM TotalStorage Enterprise Storage Server (ESS)
- IBM TotalStorage Peer-to-Peer Virtual Tape Server (PtP VTS)
- Optical Dense or Coarse Wavelength Division Multiplexor (DWDM or CWDM)
- Peer-to-Peer Remote Copy (PPRC) architecture
- Extended Remote Copy (XRC) architecture
- Peer-to-Peer Virtual Tape Server (PtP VTS) architecture

A GDPS configuration consists of production systems, controlling systems, and automation.

Production systems

The production systems execute the mission critical workload. There must be sufficient processing resource capacity, such as processor capacity, main storage, and channel paths available, which can quickly be brought on-line to restart a system's or site's critical workload. This is typically done by terminating one or more systems executing expendable (non-critical) work and acquiring its processing resource.

The *Capacity Backup (CBU)* feature, available on the IBM 9672, and in the IBM zSeries, is designed to provide additional processing power, which can help you to achieve significant cost savings. The CBU feature has the ability to increment capacity temporarily, when capacity is lost elsewhere in the enterprise. CBU adds central processors (CPs) or internal coupling facilities (ICFs) to the available pool of processors and is activated only in an emergency.

GDPS-CBU management automates the process of dynamically adding reserved central processors, thereby minimizing manual customer intervention and the potential for errors. The outage time for critical workloads can be reduced from hours to minutes.

Controlling system

The controlling system coordinates GDPS processing. All GDPS functions are initiated and coordinated by the controlling system.

Automation

All GDPS systems are running GDPS automation based upon Tivoli NetView® for OS/390 and System Automation for OS/390. Each system can monitor the Sysplex cluster, Coupling Facilities, and storage subsystems, and maintain GDPS status. GDPS automation can coexist with other automation products already implemented in the enterprise.

14.1.1 GDPS/PPRC

GDPS/PPRC is designed to manage and protect IT services by handling planned and unplanned exception conditions, and maintain full data integrity across multiple volumes and storage subsystems.

By managing both planned and unplanned exception conditions, GDPS/PPRC can help maximizing application availability and provide business continuity. GDPS is capable of providing the following functions:

- Continuous Availability solution
- Near transparent disaster recovery solution
- Recovery Time Objective (RTO) less than one hour for GDPS/PPRC
- Recovery Time Objective (RTO) less than two hours for GDPS/XRC
- Recovery Point Objective (RPO) of zero (optional)
- Protects against metropolitan area disasters

Figure 14-1 shows a simplified illustration of the physical topology of a GDPS/PPRC implementation, which consists of a Parallel Sysplex spread across two sites (site 1 and site 2 in Figure 14-1) separated by up to 100 kilometers of fiber with one or more z/OS systems at each site. The multi-site Parallel Sysplex must be configured with redundant hardware, for example, a Coupling Facility, and Sysplex Timer in each site, as well as alternate cross-site connections.

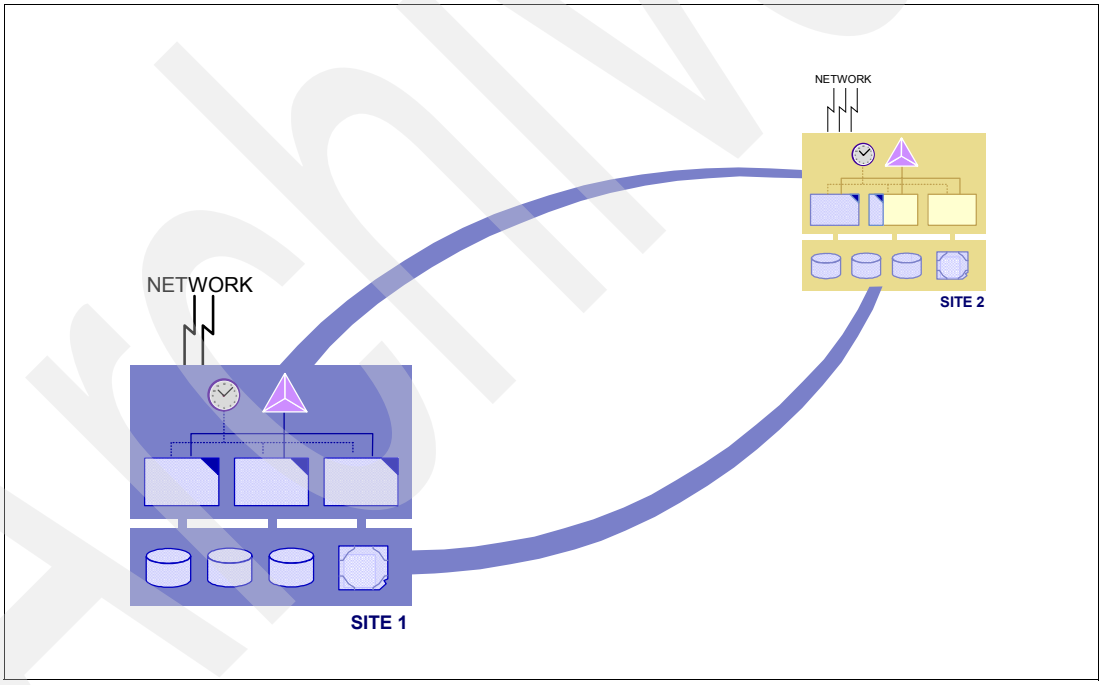


Figure 14-1 GDPS/PPRC topology

All critical data residing on disk storage subsystems in site 1 (the primary copy of data) is mirrored to the disk storage subsystem in site 2 (the secondary copy of data) through PPRC synchronous remote copy.

GDPS/PPRC provides the Parallel Sysplex cluster continuous availability benefits, and it significantly enhances the capability of an enterprise to recover from disasters and other failures, as well as managing planned actions.

Planned re-configuration support

GDPS/PPRC planned re-configuration support automates procedures performed by an operation center. These includes standard actions which can be initiated against a single system or a group of systems. For example, you can:

- Quiesce a system's workload and remove the system from the Parallel Sysplex cluster.
- IPL a system.
- Quiesce a system's workload, remove the system from Parallel Sysplex cluster, and re-IPL the system.

Additionally, GDPS/PPRC provides customized scripting capability for user defined actions; for example:

- Planned disk maintenance or planned site switch in which the workload is switched from processors in site-1 to processors in site-2.

Unplanned re-configuration support

GDPS/PPRC unplanned re-configuration support not only can automate procedures to handle site failures, but is designed to also minimize the impact and potentially mask a z/OS system, processor, Coupling Facility, disk subsystem or tape failure, depending on the GDPS/PPRC policies installed. Some examples of unplanned failures that can be managed by GDPS/PPRC are as follows:

- In the event of a z/OS or OS/390 system failure, the failed system and workload can be automatically restarted.
- In the event of processor failure, the failed system(s) and the workload can be restarted on other processors.
- In the event of primary disk storage subsystem failure, a disk re-configuration will allow access to the secondary PPRC volumes, which will contain mirrored data consistent with the primary data.

GDPS/PPRC HyperSwap

The GDPS/PPRC HyperSwap function provides the ability to transparently switch the applications I/O activity to the secondary PPRC volumes for both planned and unplanned reconfiguration. Large configurations can be supported, as HyperSwap has been designed to provide the swap of large number of volumes very quickly. The important ability to re-synchronize incremental disk data changes in both directions between primary and secondary PPRC disks is provided as part of this function.

The GDPS/PPRC HyperSwap function is designed to broaden the continuous availability attributes of GDPS/PPRC. This function can help significantly increase the speed of switching sites and switching disks between sites. The HyperSwap function is designed to be controlled by complete automation, allowing all aspects of the site switch to be controlled through GDPS.

The design objective of the GDPS/PPRC HiperSwap line-item is to improve Parallel Sysplex Continuous Availability and Disaster Recovery attributes by masking some disk subsystem outages for PPRC Level 3 capable disk subsystems, as well as masking planned disk subsystem maintenance activities. Figure 14-2 shows how GPDS/PPRC HyperSwap works.

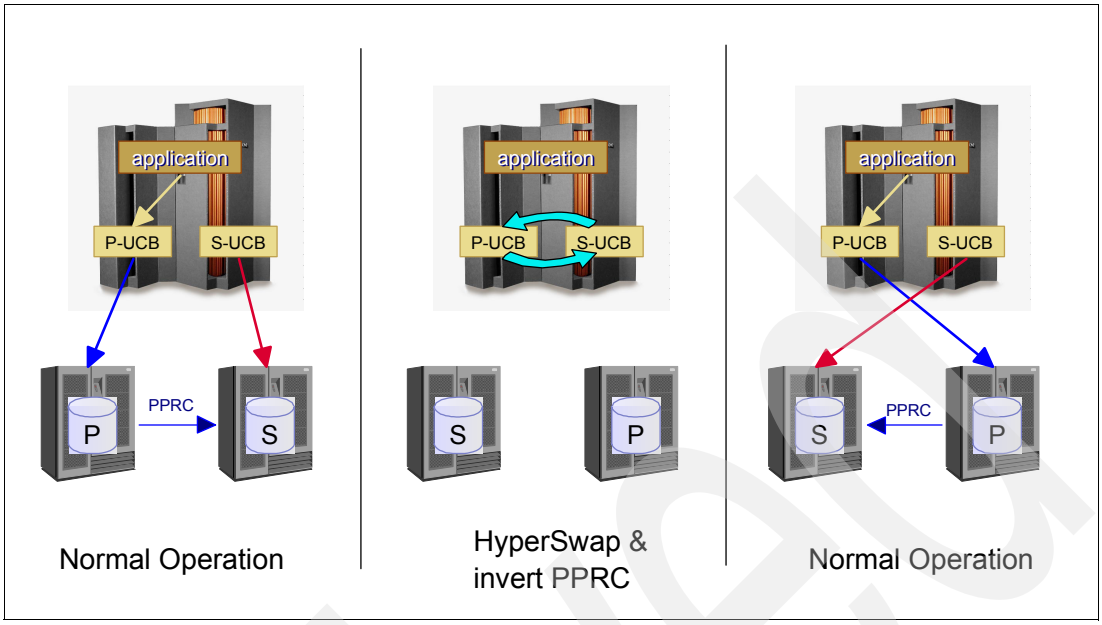


Figure 14-2 GDPS/PPRC HyperSwap

If a primary PPRC disk subsystem experienced a problem. GDPS/PPRC will automatically switch to use the secondary PPRC disk subsystem. If the same enterprise wanted to perform a maintenance action against a primary PPRC disk subsystem, GDPS/PPRC will transparently switch to use the secondary PPRC disk subsystem.

Additionally, the existing freeze processing for the *FREEZE&GO*, *FREEZE&STOP*, and *FREEZE&STOP CONDITIONAL* freeze options are extended to protect against primary PPRC disk subsystem failures. Specifically, HyperSwap adds the IEA497I message as a freeze trigger to prevent any logical contamination from being propagated to the secondary PPRC volumes. If a primary PPRC disk subsystem experiences a problem, GDPS/PPRC will perform the freeze function, and the enterprise will execute a TAKEOVER script to perform DASD switch.

The performance objective is for GDPS/PPRC to perform the HyperSwap for one SSID pair (up to 256 PPRC volume pairs) within 5 seconds and all the defined SSID pairs (10K PPRC spread over 40 SSID pairs) within 15 seconds for any disk subsystem(s) outage. Some enterprises cannot tolerate a longer outage. Considering that today enterprises experience a multi-hour outage when there is a disk subsystem failure or tolerate 85 second spin loops (the default spin loop factor under z/OS images under LPAR), up to 1 minute is significantly better.

GDPS/PPRC supports the HyperSwap function for planned re-configuration since version 2.8; in addition, version 3.1 supports HyperSwap for unplanned re-configuration.

There is no equivalent GDPS/XRC support since XRC is asynchronous remote copy.

14.1.2 GDPS/XRC

GDPS/XRC provides an automated solution to protect from regional disaster. Due to the distance, it exploit XRC as disk mirroring techniques to avoid performance impacts on production systems.

In addition to providing the automation for unplanned re-configuration support (the primary attribute of most disaster recovery solutions) GDPS/XRC is designed with the following characteristics:

- Disaster recovery solution
- RTO between an hour to two hours
- RPO less than two minutes
- Protects against metropolitan as well as regional disasters (unlimited distance)
- Minimal remote copy performance impact

GDPS/XRC consists of one or more production systems in site 1, where the production system can be a single system, multiple systems sharing disk subsystems, or a base or Parallel Sysplex. Figure 14-3 shows a simplified illustration of the physical topology of a GDPS/XRC implementation.

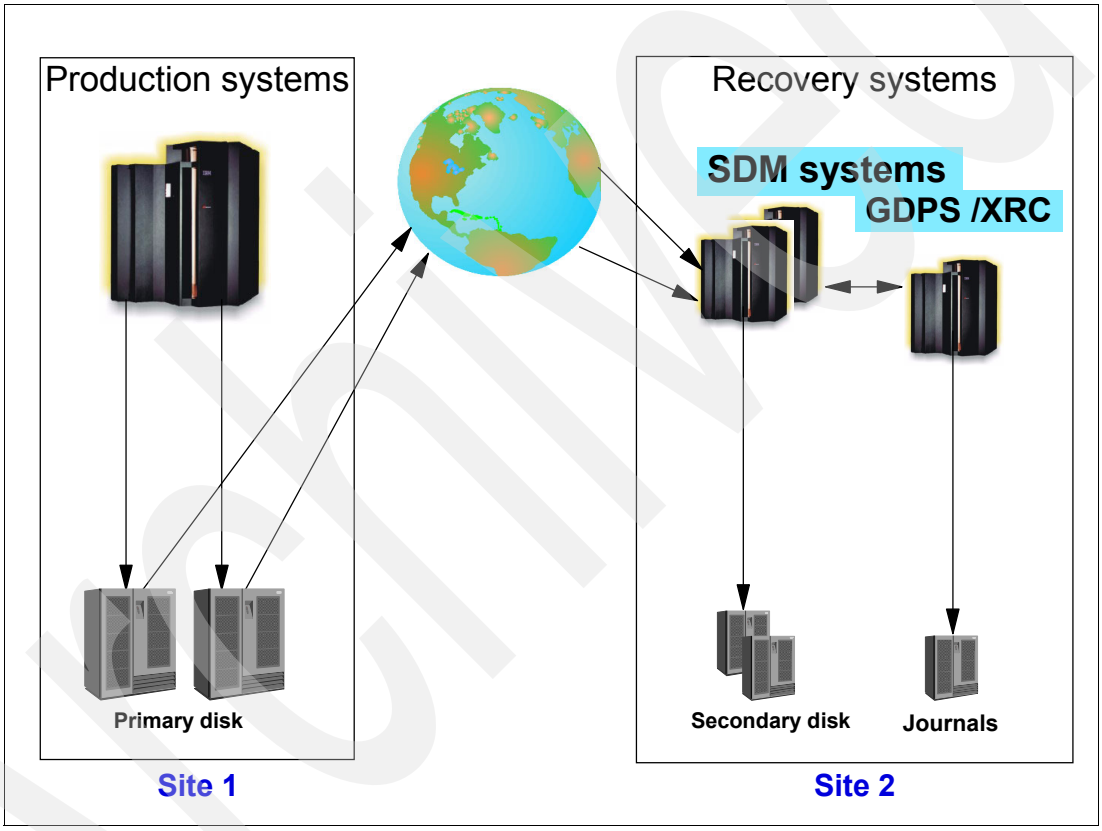


Figure 14-3 GDPS/XRC topology

Note that the Parallel Sysplex does not span sites 1 and 2. Site 2, (the recovery site) can be located at virtually any distance from site 1 (the production site). During normal operations, the XRC System Data Mover (one or more) is executed in site 2, and is in a Base Sysplex with the GDPS controlling system.

All critical data resides on storage subsystems in site 1 (the primary copy of data) and is mirrored to the storage subsystems in site 2 (the secondary copy of data) through XRC asynchronous remote copy.

Planned re-configuration support

All planned re-configuration actions (as supported by GDPS/PPRC) are provided by GDPS/XRC for the System Data Mover (SDM) Sysplex in site-2. For example, GDPS/XRC will manage the temporary relocation of the SDM, if it is needed. By managing the SDM Sysplex. GDPS/XRC can also manage the XRC remote copy configuration.

Unplanned re-configuration support

It is highly recommended that the disaster recovery solution be completely automated. GDPS/XRC automates the process of recovering the production environment with minimal manual intervention, thus providing significant value in minimizing the duration of the recovery window.

Coupled System Data Mover support

A single SDM can typically manage approximately 1,000 to 2,000 volume pairs depending on the write I/O rate. Coupled Extended Remote Copy (CXRC) expands the capability of XRC, by ensuring configurations with many thousands of primary volumes using more than one SDM, and enabling recovery of all the volumes in the configuration to a consistent point-in-time. CXRC provides the scalability that is required to support larger XRC configurations.

14.1.3 Functional highlights

Also, the following functions are supported by both GDPS/PPRC and GDPS/XRC:

- Peer to Peer Virtual Tape Server (PtP VTS):

  GDPS supports Peer to Peer Virtual Tape Server. By extending GDPS support to data resident on tape, the GDPS solution provides continuous availability and near transparent business continuity benefit for both disk subsystems and tape resident data.

  Enterprises are no longer forced to develop and use processes that create duplex tapes and maintain the tape copies in alternate sites. For example, previous techniques created two copies of each DBMS (Database Management System) image copy and archived log as part of the batch process and manual transportation of each set of tapes to different locations.

  Operational data, data that is used directly by applications supporting end users, is normally found on disk subsystems. However, there is another category of data that supports the operational data, which is typically found on tape subsystems. Support data typically covers migrated data, point-in-time backups, and archive data. For sustained operation in the failover site, the support data is required. Furthermore, several enterprises have mission critical data that only resides on tape.

  The PtP VTS provides a hardware based tape duplexing solution. In the event of a planned site switch or a site failure, GDPS will automatically manage the duplexed tapes. GDPS will FREEZE copy operations, so that tape data consistency can be maintained across GDPS managed sites during a switch between the primary and secondary VTSs.

- FlashCopy:

  FlashCopy, available on the IBM TotalStorage Enterprise Storage Server (ESS), provides a point-in-time copy of data in a minimum amount of time. FlashCopy enables you to copy or dump data while applications are updating the data.

  FlashCopy is automatically invoked before re-synchronization commences (based upon the policy activated), whenever a re-synchronization request is received. This function ensures that there is always a consistent image of data to fall back to, in the event that a disaster occurs, while re-synchronization is in progress.

- Management of zSeries Operating Systems:

  In addition to managing images within the base or Parallel Sysplex cluster, GDPS can also manage other production operating systems. These include z/OS, Linux for zSeries, z/VM®, and VSE/ESA™.

  The operating systems have to run on servers that are connected to the same Hardware Management Console (HMC) local area network (LAN) as the Parallel Sysplex cluster images. For example, if the volumes associated with the Linux images are mirrored using PPRC, GDPS can restart these images as part of a planned or unplanned site re-configuration. The Linux for zSeries images can either run as a logical partition (LPAR) or as a guest under z/VM.

14.1.4 IBM Global Services offerings

GDPS is available through IBM Global Services (IGS) only. The following GDPS services, for both PPRC and XRC, are offered:

- Technical Consulting Workshop (TCW)

  Typically, the TCW is a two day workshop that helps to determine whether GDPS is the right solution for your business. The objective is that IGS specialists work with customer representatives to understand the business objectives, service requirements, technological directions, business applications, recovery processes, cross-site and I/O requirements. High-level education on GDPS is typically provided, along with the service and implementation process. Various remote and local data protection options are evaluated.

  Usually, IGS specialists will present a number of planned and unplanned GDPS re-configuration scenarios, with recommendations on how GDPS can assist you in achieving your objectives.

- Remote Copy Management Facility (RCMF)

  With this service, the RCMF automation to manage the remote copy infrastructure is installed, the automation policy customized, and the automation verified along with providing operational education for the enterprise.

- Geographically Dispersed Parallel Sysplex (GDPS)

  Typically, this service will consist of the installation of the GDPS automation to manage the remote copy infrastructure, perform routine tasks, and recover from failures, with the automation policy customized, and the automation verified along with providing operational education for the enterprise.

14.2 GDPS-Metro Solution

We have already stated in Chapter 7, “Peer-to-Peer Remote Copy” on page 97 that PPRC Synchronous (Metro) is not sufficient tor a Disaster Recovery by itself, as it is susceptible to a rolling disaster. GDPS, a set of IBM services provides automation to overcome this deficiency. While GDPS supports a number of different options, we are conveying the concepts by describing a simple configuration and its solution in this chapter. The following sections describes a simple GDPS configuration

14.2.1 Initial setup

Figure 14-4 shows a parallel sysplex with two sites, each with a Coupling Facility (CF) and a Sysplex timer.

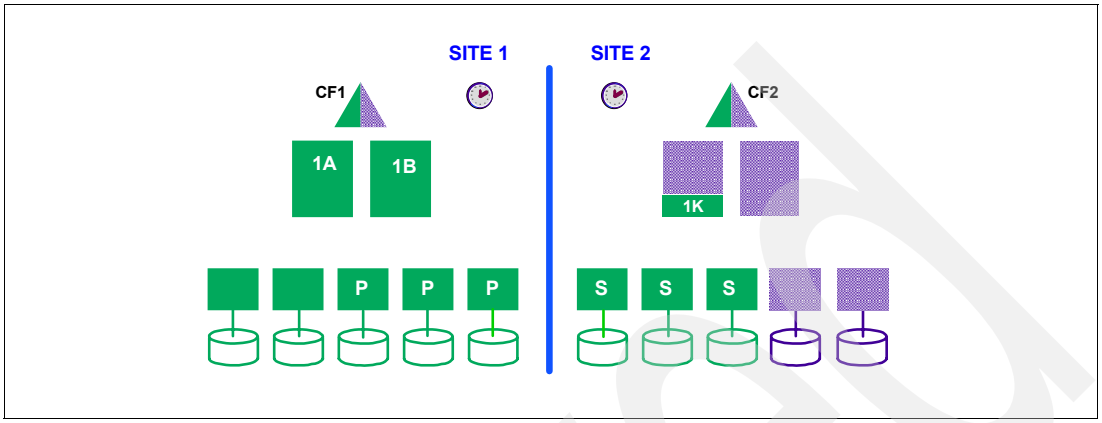


Figure 14-4 GDPS setup

Site 1 is the production site and holds all the production systems. Here are the site 1 components:

- All the primary DASD, labeled “P”. They are connected as PPRC primary DASD
- Two z/OS images, labeled “1A” and “1B”
- The shaded part of the CF illustrates spare capacity that is planned for use during failover if a loss of CF occurs at any time, not specifically for a site loss situation. CFs should always be configured such that if any one fails, the other can absorb its workload from both a storage and a link capacity standpoint.

Since all the workload is running at the primary site, there is more than one CF, which is not shown. In this figure, no workload from the primary site accesses a CF on the secondary site.

Site 2 is the recovery site. These are its components:

- All secondary DASD. It is connected as PPRC secondary DASD.
- A second CF configured similarly to the one at the primary site. It will be used by systems at the primary site only if there is a site failover. It is not intended to show that the DB2 Group Buffer Pools (GBP) are duplexed to the secondary site.
- A small z/OS image, labeled “1K”. This is the z/OS image that monitors activity at the primary site
- Other z/OS images, shown as shaded. They represent discretionary images and workload that will be displaced if a site failover occurs.

14.2.2 The Freeze Policy

As mentioned before, PPRC cannot guarantee time consistency between two different LSSs, so we need a function that can operate at the logical control unit level. In essence, at the first detection of a possible disaster condition, we need to *FREEZE* at the same time all the LSSs at the local site. FREEZE is a PPRC Level 2 capability available from ESS and from any DASD vendor that licenses it, and is eligible to support GDPS.

It is the FREEZE capability across all LSSs and the automation that allows GDPS to create a consistency group at the secondary site and overcomes the rolling disaster.

The Freeze Policy is defined by you to incorporate the business decisions by your enterprise as to what should happen should a disaster occur at the local site. You can choose among four different policies as shown in Figure 14-5.

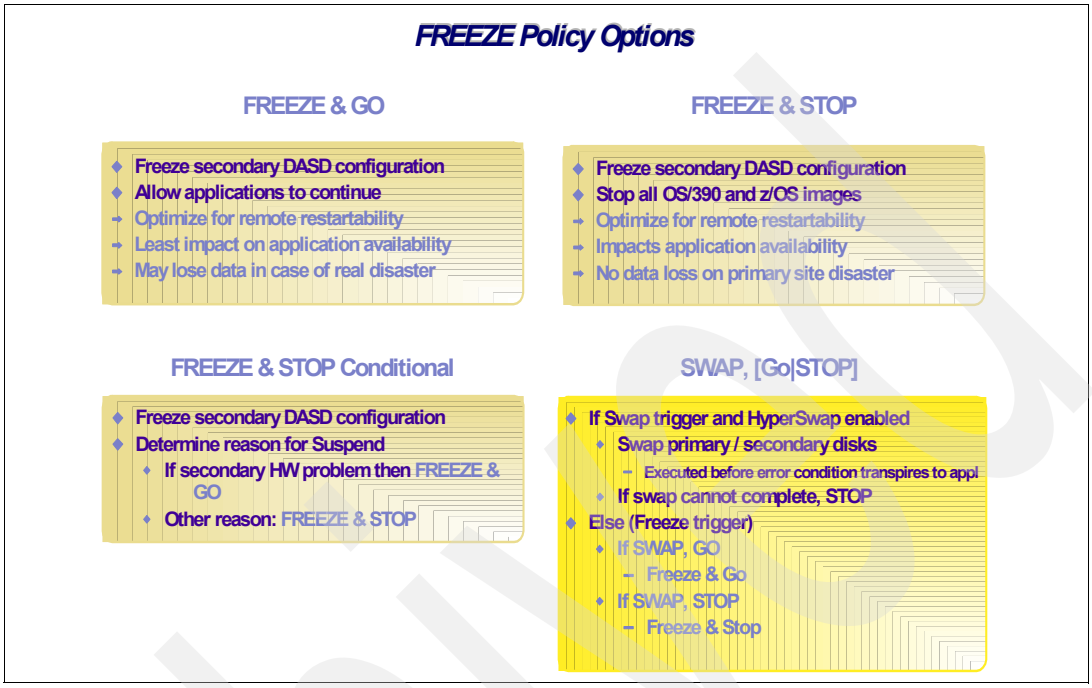


Figure 14-5 Freeze options

The freeze policy is activated when GDPS detects an imminent disaster at the local site which the “1K” system is monitoring, like, for example: failure in the DASD subsystem, failure in the z/OS images, failure in the CF.

GDPS reacts to two types of triggers: Freeze triggers and HyperSwap triggers. A Freeze trigger is an indication of a mirroring problem, and a HyperSwap trigger is an indicator of a primary disk failure. Your GDPS Freeze policy specifies what action(s) GDPS is to take in the event that a Freeze or HyperSwap triggering message is captured.

A brief description of the four policies follows.

Freeze and Go

Upon indication of failure, the secondary DASD is frozen. It can be restarted with time consistent data at the recovery site. Overcoming the rolling disaster is the first priority. The second priority is keeping the local site continuing, as it may not be known at that point if the disaster will materialize. This approach has the least impact on the local site. However, if a disaster does occur within the next few minutes or so, there will be updates at the primary site that are not reflected at the recovery site. The number of updates not propagated to the secondary is a function of the amount of time before the site fails.

Freeze and Stop

This policy also freezes all the secondary DASD at the first indication of failure. However, in order to avoid data loss, it stops the entire local environment, including all local z/OS images and theDB2 data sharing group. There will be no data loss at the secondary side. There will be no rolling disaster.

Freeze and Stop Conditional

This is a variation on Freeze and Stop. If the reason for the failure is a loss of connectivity to secondary DASD, then there is a now a consistent point for recovery. However, there will be no further consistency until the connectivity is restored. Since the local site is intact, the processing changes to Freeze and Go.

Swap, Go or Stop

The first parameter (SWAP) specifies that if HyperSwap is enabled, a HyperSwap is to be attempted in response of HyperSwap triggers. The HyperSwap function starts with a Freeze followed by swapping the primary / secondary disks. The second parameter (GO or STOP) indicates the action GDPS is to take if a Freeze trigger rather than a HyperSwap trigger, GO or STOP will cause GDPS to react the same as if you had specified Freeze&Go or Freeze&Stop, respectively.

When the disaster occurs

Figure 14-6 illustrates the functions that GDPS performs when an indication of a potential disaster occurs.

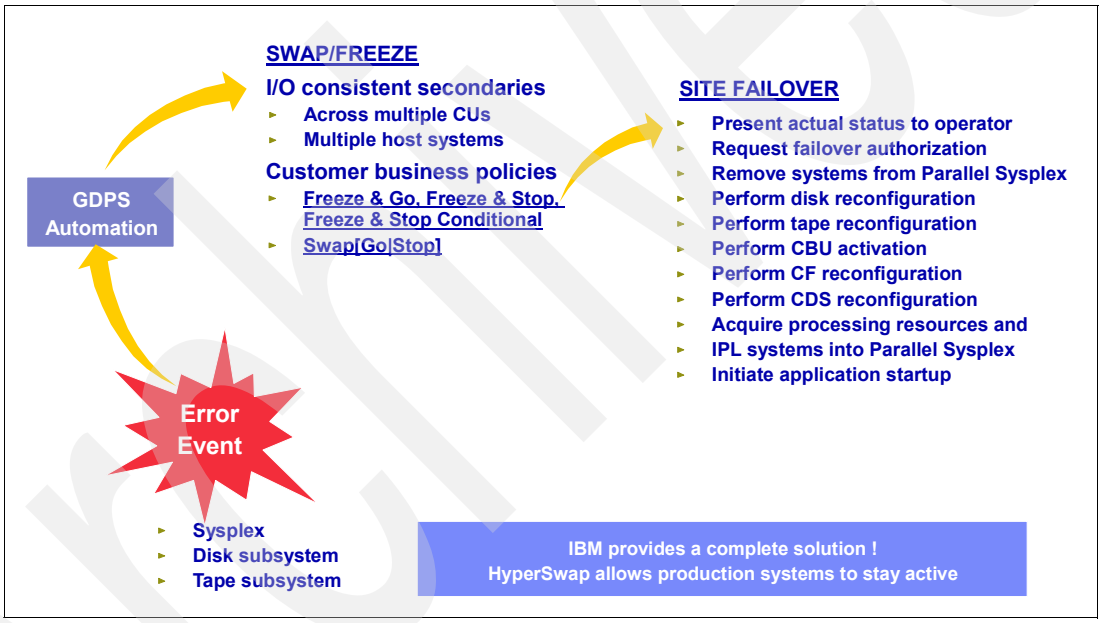


Figure 14-6 Cross-Site® reconfiguration

The error event occurs somewhere within the parallel sysplex as observed by “1K” at the recovery site. All secondary DASD is frozen, insuring a consistent point for a recovery. The situation management capability of GDPS is activated and will execute the site failover.

First, an operator confirmation is required to validate the situation. Then the Freeze policy is performed. Based on the policy, either z/OS images will be allowed to continue (Freeze and Go) or they will be stopped (Freeze and Stop).

In event of disaster, the z/OS images are removed fro the Parallel Sysplex. The secondary DASD is renamed to the volume serials that existed at the primary site.

All of the “upstream” systems on which DB2 relies, must be in place. The z/OS images are restarted and the network is switched. Following those activities DB2 is restarted.

The CF policy at the primary site cannot be used at the recovery site (serial numbers of the CFs are different at the minimum), and the structures in use at the primary site must be allocated in the secondary site's CF. A group restart for must be performed for data sharing members. Even if there were duplexed DB2 structures in the CF at the secondary site, they cannot be used. Since the CF cannot be synchronized with DASD in the current releases of z/OS, the CF's information will be more current than that on the frozen secondary DASD. Even if the policy is Freeze and Stop, it is possible that the first point of failure could be loss of connectivity to the CF and thus DASD could be ahead of the CF data at the secondary site.

14.2.3 After Failover to Recovery Site

Figure 14-7 depicts the situation where Site 1 has failed.

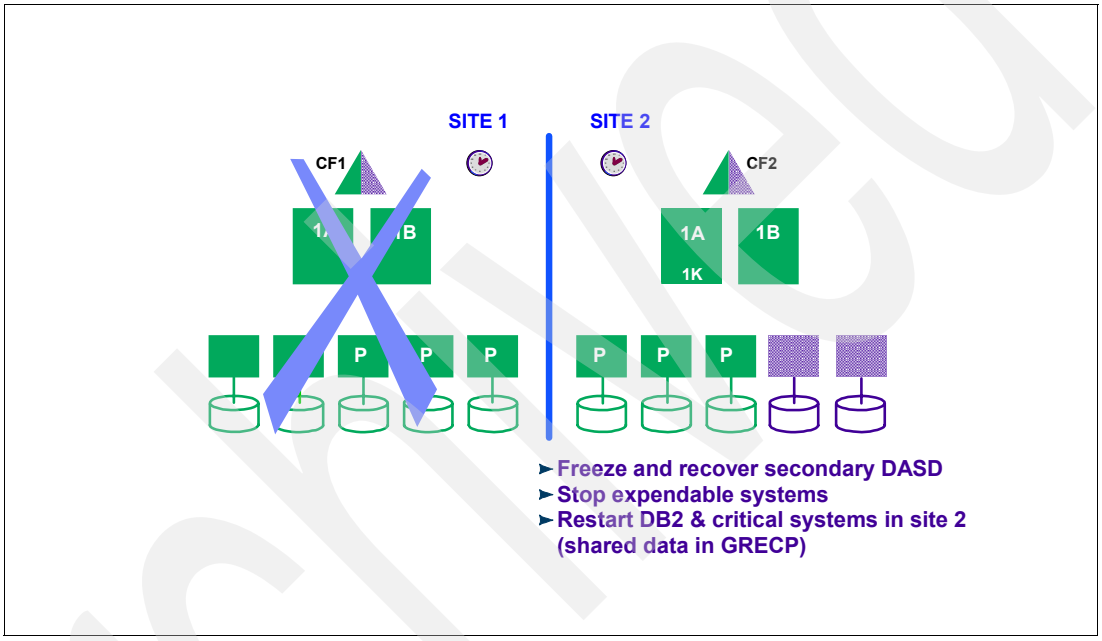


Figure 14-7 Site 1 failure

At the end of this scenario, GDPS has switched the workload to the secondary site by the activation of the correct Freeze Policy

- Site 1 has been marked inoperative with an "X"
- Site 2 is now active
- All DASD is now marked as "P", so it has been switched
- z/OS images "1A" and "1B" are now operative
- z/OS images that were shaded have now been replaced by "1A" and "1B"
- CF2 is being used for the DB2 GBPs, LOCK, and SCA structures (as well as others)
- After DB2 is started by GDPS, there will be DB2 table spaces and indexes that are in Group Buffer Pool Recovery Pending (GRECP) status. They are DB2 objects that were in any of the GBPs at the time that Site 1 was switched to Site 2. They were discovered during DB2 restart and when restart is complete, you must recover them

To remove objects from GRECP, you issue DB2 -START DATABASE commands as we describe in 21.6.5, "GRECP/LPL recovery recommendations" on page 428. The amount of time to recover those objects can take from a few minutes if there are hundreds of shared objects to perhaps an hour if there are thousands. It is impossible to predict how many

objects will be GRECP, because at any given point-in-time, the number can vary. DB2 imposes no data sharing overhead on objects that are not being shared. The shareability is dynamic, based on workload at the time.

14.2.4 Considerations about GDPS

Here are some considerations to keep in mind:

Advantages

- DB2 staff is minimally involved.
- DB2 is fully available.
- RPO is current.
- No DB2 complex procedures are needed.
- There are no infrequent and/or incremental image copy application issues.
- Current RTO - Normal restart of DB2 occurs.
- Elimination of a *Rolling Disaster* is a benefit.
- It can be used for planned outage.
- It can also be used for local LCU/LSS micro code failure.
- Automation and Freeze Policy are implemented easily.

Disadvantages

- Cost — A second site with full capacity with duplicate DASD (Sysplex timer and CF) is needed.
- For data sharing, all shared data sets are set to GRECP and recovery is required until CF Freeze. Outage can be from minutes to perhaps an hour if thousands of data sets are in GRECP.
- It must be within ESCON/FICON distance (40 KM - timer link limitation, soon to be 103 KM).
- It is necessary to IPL until HyperSwap switches secondary to primary non-disruptively.

  This implies a cloned workload split among two sites. A non-disruptive switch means that workload continues on the secondary site while images on the local site are IPL'd at the secondary site. This situation is uncommon.

14.3 Metro/Global Mirror solution

This chapter provides a scenario which has sometimes been called “three-site no data loss” or “three-site solution using Synchronous PPRC and XRC”. It can address the situation where there is a requirement for a secondary site across continental distance but there can be no data loss.

Global Mirror XRC provides most of the requirement, but due to its asynchronous nature, some data loss, though minute (perhaps less than 1/2 second) will occur. With Metro Mirror there is no data loss, but that capability is limited to metro distances. A combination of the two approaches can be employed to achieve the desired goal. This scenario appeals to institutions with fiduciary responsibility.

The technologies used in this solution are:

- Metro Mirror (PPRC Synchronous), which provides the synchronous mirroring capability at a near-by site (a few blocks away).
- Global Mirror XRC, which provides the asynchronous mirroring capability even at continental distances.
- GDPS, which provides all the automation required by the solution.

14.3.1 Scenario 1

Figure 14-8 describes the scenario. The assumption is that the primary site (Site-1a) is located at a very short distance from the computer site (Site-1), possibly across the street or in a campus building. As the figure shows, it can be just a DASD farm.

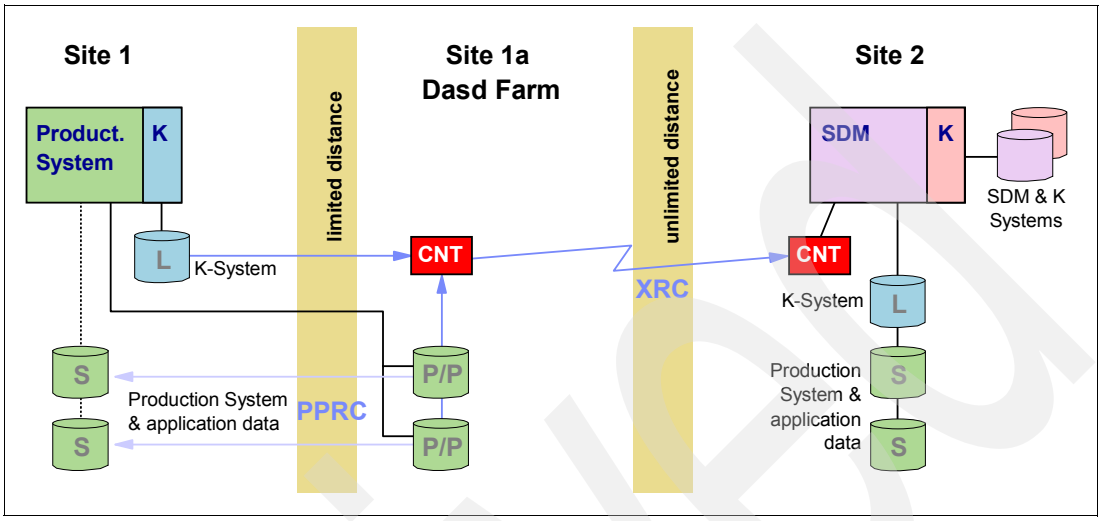


Figure 14-8 A three-site, continental distance solution with no data loss

The key point is that primary DASD is at Site-1a while the secondary is at Site-1. This aspect enables Site-1a to be a primary to an XRC secondary. A PPRC secondary cannot do this.

> **Note:** This configuration provides high availability to the local site. In the event of failure of one LSS/LCU or DASD subsystem, there will be no mass object recovery required.

There are two production LPARs at Site-1, “Product System” and K. At the secondary site (Site-2) the SDM runs on K. System K is the GDPS monitoring z/OS image. The vertical lines represent physical dividers between the three sites. The distance between the DASD farm (Site-1a) and the secondary site (Site-2) can be any distance.

There is PPRC between Site-1 and Site-1a and there is no local write penalty for the primary DASD from Site-1a to Site-2. Site 1a is connected via channel extenders to Site 2, the host disaster recovery site. The SDM runs at Site-2 and, pulls off all DASD I/O from Site-1a, forms consistency groups, and applies them to a third set of DASD. Site-2 normally lags Site-1.

GDPS is monitoring, among other things, the DASD in Site-1 and Site-1a. When there is a failure, GDPS will freeze the secondary DASD at Site-1.

In the configuration shown in Figure 14-8, one of the following scenarios could happen:

- If Site-1 is lost, the last DB2 writes have occurred to Site-1a. They are read by the SDM at Site-2 through the channel extenders and since no more updates happens at Site-1a, the last consistency group will be formed and applied to Site-2 secondary XRC disks. GDPS take care of restarting DB2 at Site-2 with no data loss.
- If you lose Site-1a, GDPS takes care of handling this situation. Either of the following two possibilities will occur:
  - GDPS will use HyperSwap to swap your DB2 I/Os to secondary PPRC devices.
  - GDPS will restart DB2 at Site-1after having recovered PPRC secondary disks.

However, in either cases, you loose the capability to restart you operation from Site-2, since the primary XRC disks in Site-1a are lost and data on secondary XRC disks at Site-2 remains “frozen” at the time of Site-1a failure.

- If you lose Site-2, your DB2 continues to run at Site-1 with no data loss across Sites-1 and Site-1a.

14.3.2 Scenario 2

As we saw in 14.3.1, “Scenario 1” on page 246, a Site-1a failure, although it doesn’t impact Site-1 production systems, can still compromise the capability to restart IT operations at Site-2, since XRC mirroring has been terminated at the Site-1a failure.

However, we think that you could build this solution to protect you from a disaster in Site-1 (your production site), which still has the same probability to occur even if another disaster strikes Site-1a. And, we also think that restarting with minimal data loss is better than not restarting at all.

To restore your restarting capability at Site-2, we need to introduce in the scenario another CNT device at Site-1 and to have the TLC connection from Site-2 switchable from Site-1a to Site-1 on request. Figure 14-9 shows the new scenario.

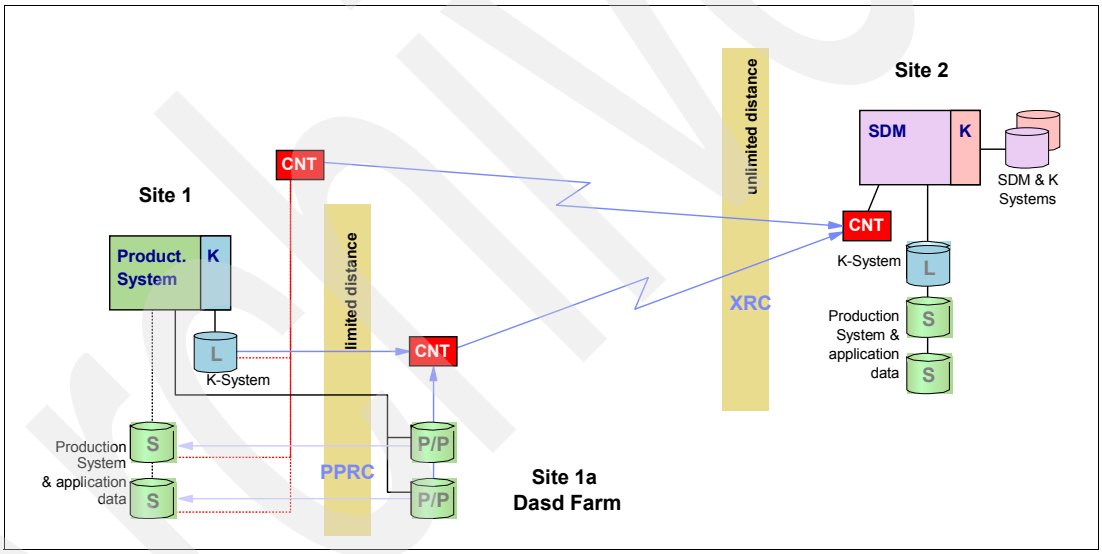


Figure 14-9 A three site-double failure solution

With such a configuration, if a disaster hits Site-1a, either of these situations will occur:

- GDPS will use HyperSwap to swap your application to secondary PPRC devices and production continues from Site-1 immediately without any disruption.
- GDPS restarts DB2 at Site-1after having recovered PPRC secondary disks.

In any case, we need to restore Site-2 mirroring capability, thus:

- XRC-TLC must be switched from Site-1a to Site-1.
- In Site-2, we must terminate XRC session active with Site-1a, and we activate another session with Site-1. However, all XRC pairs must be full copied since this is a new XRC session from Site-1 ESSs.

If a disaster hits your production site (Site-1) nothing is different from “Scenario 1” on page 246.

Part 4

Implementing disaster recovery scenarios

In this part of the book, we describe our implementation of six user oriented DR scenarios. For each scenario we describe our environment and configuration, including relevant DB2 settings, and document our findings, pointing out what we have learned. We used three laboratories for our tests, of very different environment complexities. Whenever possible, we add some recommendations.

Part 4 contains the following chapters:

- Chapter 15, “Set Log Suspend - FlashCopy - More Log - Restore System Log Only” on page 251
- Chapter 16, “FlashCopy Consistency Group and restart” on page 281
- Chapter 17, “PPRC - FlashCopy from secondary” on page 301
- Chapter 18, “XRC and restart” on page 335
- Chapter 19, “Local recovery: System PITR” on page 363
- Chapter 20, “Restart using tape dump of copy pools” on page 389

15

Set Log Suspend - FlashCopy - More Log - Restore System Log Only

In this chapter we describe the procedure related to providing a solution based on building a consistent point-in-time (PIT). We are copying the local system using FlashCopy, then restarting it at the recovery site and applying logs as they are available by using the new DB2 Utility RESTORE SYSTEM LOGONLY.

This solution requires DB2 for z/OS V8, but *not* z/OS DFSMS V1R5, *nor* SMS managed disks. Also, it is applicable to other disk units beside Enterprise Storage Server (ESS) as long as they provide a form of *instant copy.*

This chapter covers the following topics:

- Description of the scenarios
- Recovery procedures
- Test environment
- Preparing for disaster recovery
- Setting up the recovery site
- Considerations

15.1 Description of the scenarios

In this section we describe how to restore a system to an arbitrary point-in-time (PIT) using FlashCopy and RESTORE SYSTEM LOGONLY at the recovery site. It uses FlashCopy as a backup and the RESTORE SYSTEM LOGONLY performs log apply function from the point-in-time at which FlashCopy was taken. The SET LOG SUSPEND must be issued before the FlashCopy for this scenario to work. This solution can also be used for system level point-in-time recovery in a local site.

15.1.1 FlashCopy with DB2

Most 24x7 database operations, as well as all very large database (VLDBs) operations, require a mirroring solution. This requires the ability to split off a consistent mirror of a database while OLTP keeps on running on the primary (live) database. Recovery on the split-off database is acceptable for consistency. In fact, these days, no one can afford doing offline or online backups on a 1 TB live database. Backups and system copies can be done from a mirror image.

The SET LOG SUSPEND command will temporarily *freeze* all updates in a DB2 subsystem, while the logs and database can be copied using FlashCopy. This allows you to recover the DB2 subsystem to a point-in-time without having to experience an extended recovery outage, or without having to *stop* or *quiesce* the local system.

In Figure 15-1, the SET LOG SUSPEND command is issued and logging is suspended at RBA (or LRSN) “A”. All volumes can be dumped using FlashCopy in only a few minutes. The URs which are waiting on the log latch hold locks. Depending on your settings, some transactions may time out waiting on locks held by the log latch waiters.

Once the FlashCopy “Logical Complete” occurs, logging is resumed at RBA (or LRSN) “B” by SET LOG RESUME. You must then dump the copies to tape and send them to a recovery site backup system. The dumping to tape is not a trivial task and can take hours. Throughout the day, you also need to send secondary archive logs to the recovery site.

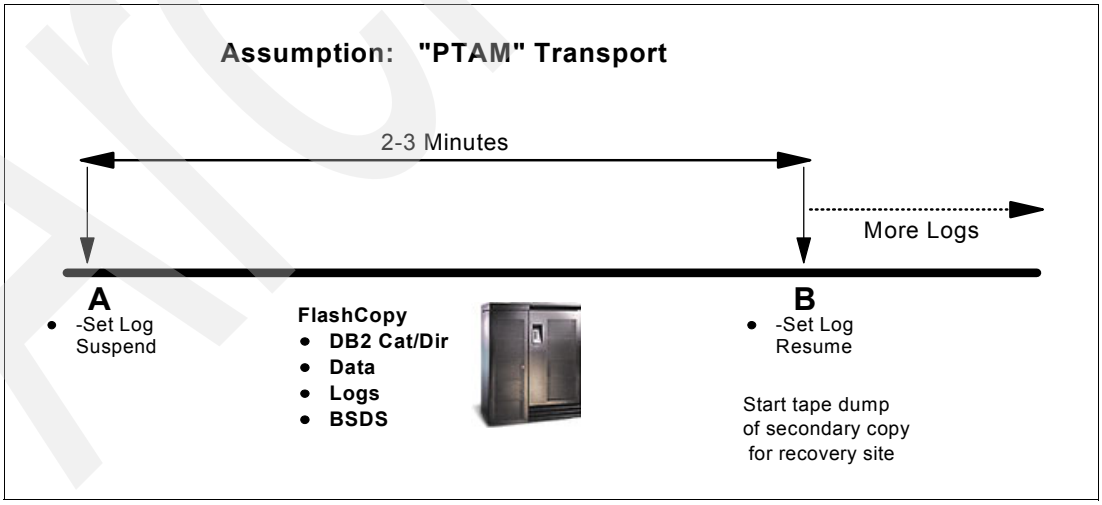


Figure 15-1 FlashCopy with SET LOG SUSPEND

The assumption is that we are using the very traditional Pickup Truck Access Method (PTAM) to transfer copies to the remote site.

For more information on the FlashCopy, refer to Chapter 5, “ESS FlashCopy” on page 67.

15.1.2 RESTORE SYSTEM LOGONLY

The primary focus of the RESTORE SYSTEM utility is to provide an easier way to recover an entire DB2 subsystem or data sharing group to any point-in-time, regardless of whether you have uncommitted units of work.

RESTORE SYSTEM LOGONLY applies log records to recover database objects to the specified PIT after the database volumes have been restored from FlashCopy. You can specify the SYSPITR (either ENDRBA or ENDLRSN) as a PIT that you want to recover to.

The assumed scenario is that we have made a FlashCopy at midnight, and have shipped copies and log archives as they are offloaded. At 3 pm the site disaster occurs. The example shows how it is desirable to apply approximately 15 hours of log compared to restoring the midnight FlashCopy and reprocess 15 hours of business transactions.

Because you are applying 15 more hours of log than what is reflected by the FlashCopy, you must perform a restart as described in 15.5, “Setting up the recovery site” on page 260. The system level restart point is “B”, which is 15 hours later that “A”.

If archive logs are not available, the alternative would be to just restore to A from the FlashCopy tapes, and then somehow reprocess all the transactions received during the 15 hours.

In Figure 15-2, you restore the DB2 subsystem from tape copies of the FlashCopy. This can take an hour or more depending on how many volumes are copied and how many tape/disk paths are available in parallel at the recovery site. Then the DB2 RESTORE SYSTEM LOGONLY will Log Apply 15 hours worth of log.

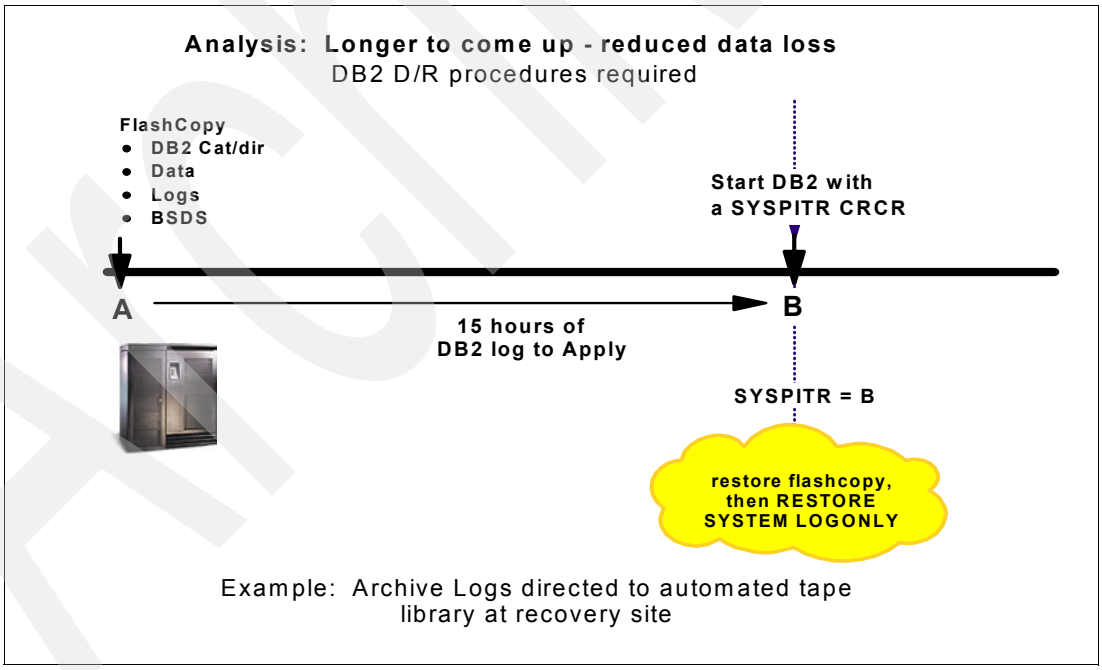


Figure 15-2 Restore FlashCopy - Forward Apply Logs

15.2 Recovery procedures

In this section, we describe the steps necessary to restore a DB2 subsystem and data sharing group to an arbitrary point-in-time at recovery site. You can also use this solution for the local site recovery. You may bypass step 5., “Recover the BSDS:” on page 254 at that time. For the detailed information on local site recovery, refer to Chapter 19, “Local recovery: System PITR” on page 363.

This solution uses the FlashCopy and RESTORE SYSTEM LOGONLY that provides the recovery of all database update activities between FlashCopy and PIT to which you want to recover.

15.2.1 Prepare for disaster recovery

For the detailed information, refer to 15.4, “Preparing for disaster recovery” on page 256.

1. Take a system backup:
   a. Execute SET LOG SUSPEND to stop update activity. If data sharing, issue it from each member.
   b. Use the FlashCopy function to copy all DB2 volumes (User DB, System DB, Active Logs, Archive LOGs, BSDS, Product Libraries, ICF catalogs).
   c. Execute SET LOG RESUME to resume update activity. If data sharing, issue it from each member.
   d. Use DFSMSdss to dump the FlashCopy to tape and then send these tapes to the recovery site.
2. Use more Active Logs.
3. Throughout the day, direct the second archive log to the automated tape library (ATL) at recovery site.

15.2.2 Set up for recovery site

For the detailed information, refer to 15.5, “Setting up the recovery site” on page 260.

1. Unallocate the ICF catalogs that reside on the target volumes.
2. Use DFSMSdss RESTORE to restore the FlashCopy data sets to disk.
3. Use the access method services IMPORT command to import the any ICF catalogs that did not exist previously at the recovery site.
4. Keep the current archive log.

Important: The assumption is that we have sent all archive logs on recovery site to apply more logs than the point-in-time at which FlashCopy was taken.

5. Recover the BSDS:

   If data sharing, the BSDS data sets on each data sharing member need to be restored.
   a. Use REPRO command to restore BSDS from the latest archive log tape.
   b. Use the change log inventory utility (DSNJU003) to register the last archive log data set in the archive log inventory of the BSDS just restored.

You must provide its starting (STARTRBA) and ending RBAs (ENDRBA) in the NEWLOG statement for the Change Log Inventory utility. If you are using data sharing, you must additionally provide its starting (STARTLRSN) and ending LRSNs (ENDLRSN).

c. Use the DSNJU003 to adjust the active logs (use the DELETE option to delete all active logs in the BSDS and NEWLOG statement to add the active log data sets to the BSDS. Do not specify a STARTRBA or ENDRBA in NEWLOG statement).

d. If you are using the DB2 distributed data facility, run the DSNJU003 to update the LOCATION and LUNAME in the BSDS.

e. Use the print log map utility (DSNJU004) to ensure that the BSDS correctly reflects the active and archive log data set inventories.

f. If you are using dual BSDSs, make a copy of the newly restored BSDS data set to the second BSDS data set.

6. Establish the SYSPITR CRCR:

   a. Determine the PIT to which you want to recover.

   b. Use DSNJU003 to create a SYSPITR CRCR record (CRESTART CREATE SYSPITR=logpoint).

      This is the log truncation point (that is the RBA or LRSN of the point which you just identified at step a). If data sharing, create a same PITR record for each active member.

   For the detailed information on how to find PITR, refer to 15.6.1, “How to find the SYSPITR RBA on the last archive log” on page 275.

   > **Restriction:** Before you run the RESTORE SYSTEM to recover system data, you must use the SYSPITR option of DSNJU003. You cannot specify any other option with CREATE, SYSPITR. You can run this option of the utility only **after** new function mode (NFM) is enabled.

7. Update the DSNZPARM parameter

   If data sharing, update it for all members. Use ARC2FRST=YES to specify that COPY2 archive log should be read first.

8. Restart DB2 with a SYSPITR CRCR:

   DB2 will start in system recover pending mode.

   If data sharing, delete all CF structures and restart all members.

9. Execute the RESTORE SYSTEM LOGONLY to recover to the current time (or to the time of the last log transmission from the local site).

   If data sharing, the utility only needs to be executed on one member. If the utility terminates and must be restarted it can only be restarted on the member on which it was originally executed.

   > **Restriction:** The RESTORE SYSTEM utility requires DB2 for z/OS V8. RESTORE SYSTEM LOGONLY can run in z/OS 1.3 using LOG SUSPEND / RESUME and backup volumes manually without BACKUP SYSTEM utility.

10. Recycle DB2 to reset system recover pending mode.

11. Execute the display command to check for active utilities or restricted objects:
    a. -DIS UTIL(*) and terminate any active utilities.
    b. -DIS DB(*) SP(*) LIMIT(*) RESTRICT and Recover all objects in Recovery Pending (RECP) or Rebuild Pending (RBDP) status.
12. Validate that recovery was successful.

15.3 Test environment

The tests were conducted in a *non-data sharing environment*, using the following hardware and software:

- Hardware:
  - zSeries 990 processor Model B16
  - 1 Logical Partition (LPAR) with 2 dedicated CPUs
  - 2 GB real memory
- Software:
  - z/OS V1R5, service level RSU0402
  - DB2 for z/OS V8
- Disks:
  - ESS Model 800 with FICON channels – FlashCopy V2

Figure 15-3 illustrates the disk of FlashCopy test environment.

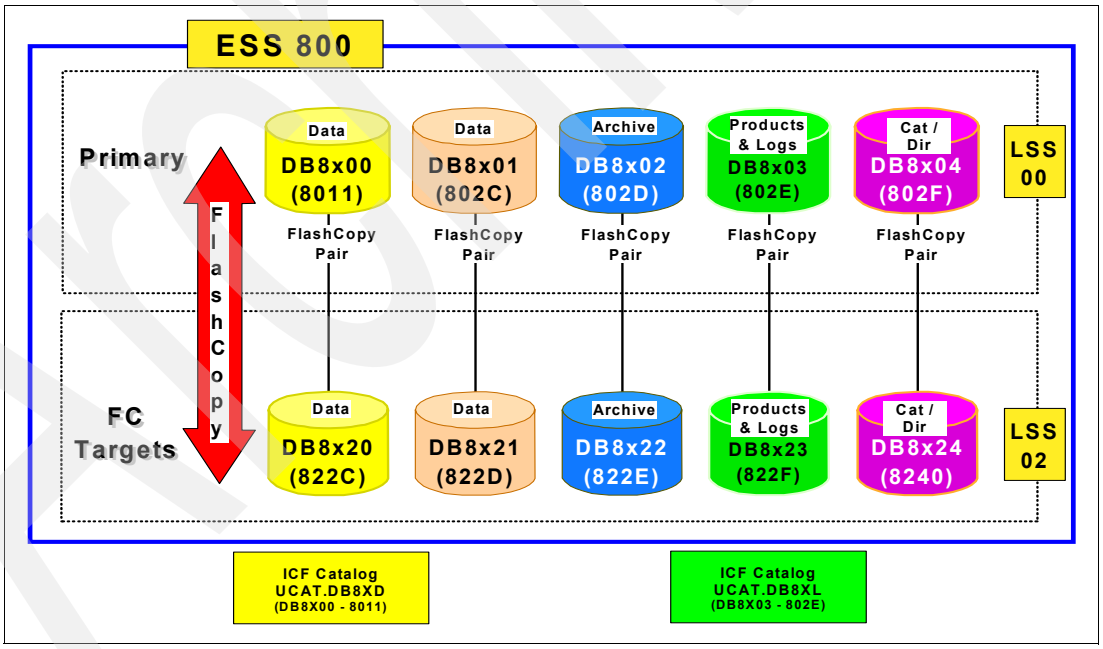


Figure 15-3 FlashCopy Test Environment

15.4 Preparing for disaster recovery

You can use the steps given in the following sections in order to prepare for disaster recovery at the local site.

15.4.1 -SET LOG SUSPEND

When the SET LOG SUSPEND is issued, DB2 suspends logging and update activity for the current DB2 subsystem until a subsequent SET LOG RESUME request is issued. A highlighted DSNJ372I message (as shown in Example 15-1) will be issued and remain on the console until update activity has been resumed.

This command has member scope. For data sharing, you must issue the command on each data sharing member and receive DSNJ372I before you begin the FlashCopy.

Example 15-1 -SET LOG SUSPEND

```
DSNJ372I  -DB8X DSNJC009 UPDATE ACTIVITY HAS BEEN SUSPENDED FOR DB8X
AT RBA 00005C65CE60, LRSN 00005C65CE60, PRIOR CHECKPOINT RBA
00005C657644
DSN9022I  -DB8X DSNJC001 '-SET LOG' NORMAL COMPLETION
```

Attention: The -SET LOG SUSPEND command updates the recovery base log point (RBLP) that is recorded in DBD01 to determine the starting point of the log apply phase. Therefore, if you take FlashCopy without a prior -SET LOG SUSPEND command, it may go back much farther in time for the start of the log apply phase than you expected.

15.4.2 FlashCopy

Here we discuss how to back up using FlashCopy functions. For more detailed information, refer to the redbook, *Implementing ESS Copy Services with IBM eServer zSeries*, SG24-5680.

FlashCopy can be invoked in many ways, but we use the DFSMSdss utility in this test scenario. For an example of how to invoke FlashCopy, see also “How to invoke FlashCopy” on page 70.

To create a tape backup of the secondary copy, you specify either DUMPCONDITIONING if the volumes are SMS-managed or NOCOPYVOLID if they are not SMS-managed.

Example 15-2 is issued to copy the source volume DB8X00 on the target volume DB8X20 using DUMPCONDITIONING.

Example 15-2 DFSMSdss COPY with DUMPCONDITIONING

```
//*******************************************************************
//*       COPY FULL W/ DUMPCONDITIONING                             *
//*******************************************************************
//CPYVOLID EXEC PGM=ADRDSSU
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  COPY FULL INDYNAM(DB8X00) OUTDYNAM(DB8X20) DUMPCONDITIONING
```

Since the keyword DUMPCONDITIONING is specified, DFSMSdss copies the source volume DB8X00 to the target volume DB8X20, but renames both the VTOC index and the VVDS to match the source volume DB8X00, as shown in Example 15-3. Although this means that all data is inaccessible, it will allow the dump of the target volume DB8X20, since it remains online.

The APAR OW48234 enhances the usability and capability of the DUMPCONDITIONING. For more information about the DFSMSdss COPY command and the DUMPCONDITIONING keyword, see 5.3.2, “DUMPCONDITIONING parameter” on page 72.

Example 15-3 Sample output of DUMPCONDITIONING

```
DSLIST - Data Sets on volume DB8X20                                Row 76 of 82
Command ===>                                                   Scroll ===> CSR

Command - Enter "/" to select action                 Message           Volume
-------------------------------------------------------------------------------
         DB8XD.DSNDBD.DSN8D81P.XDSPTXT1.J0001.A001                      DB8X20
         DB8XD.DSNDBD.DSN8D81P.XMAPRTBL.I0001.A001                      DB8X20
         DB8XD.DSNDBD.DSN8D81P.XOPTVAL1.J0001.A001                      DB8X20
         SYS1.VTOCIX.DB8X00                                             DB8X20
         SYS1.VVDS.VDB8X00                                              DB8X20
         UCAT.DB8XD                                                     DB8X20
         UCAT.DB8XD.CATINDEX                                            DB8X20
***************************** End of Data Set list ****************************
```

Once the FlashCopy “Logical Complete” occurs, the DFSMSdss job completes. The target volumes are available, and you can start dumping them to tape at that time as shown in 15.4.4, “Dump the FlashCopy to tape” on page 258. It does not wait until the copy is physically complete (which is performed by the ESS hardware). Once the copy is physically complete, the relationship between source and target is ended.

15.4.3 -SET LOG RESUME

When the SET LOG RESUME is issued, DB2 resumes logging and update activity, and deletes the DSNJ372I message from the console. See Example 15-4.

Example 15-4 -SET LOG RESUME

```
DSNJ373I  -DB8X DSNJC009 UPDATE ACTIVITY HAS BEEN RESUMED FOR DB8X
DSN9022I  -DB8X DSNJC001 '-SET LOG' NORMAL COMPLETION
```

15.4.4 Dump the FlashCopy to tape

For disaster recovery, you should dump to tape the copy to transport it to recovery site as shown in Example 15-5.

Example 15-5 DFSMSdss DUMP

```
//VDB8X00 EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X20
//OUTDS    DD  UNIT=3490,DISP=(,PASS),LABEL=4,
//             VOL=(,RETAIN,SER=(PRB000,PRB001,PRB002,PRB003)),
//             DSN=BACKUP.DB8X00
//SYSIN    DD  *
  DUMP INDDNAME(INVOL) OUTDDNAME(OUTDS) -
       FULL ALLDATA(*) ALLEXCP -
       CANCELERROR OPT(4) ADMIN
```

Because we use the *DUMPCONDITIONING* option as shown in Example 15-2, the DUMP FULL of the target device modifies the tape records so that it appears as if the tape was created directly from the source volume DB8X00, removing the requirement of having to clip the volser of the target volume DB8X20 following a subsequent restore.

15.4.5 More active logs

We perform the additional workload described below in order to provide additional log to apply.

Workloads suspended during FlashCopy

Example 15-6 lists active utilities during FlashCopy that are suspended by SET LOG SUSPEND command. They will be resumed by SET LOG RESUME command once the FlashCopy “Logical Complete” occurs.

Example 15-6 The utilities suspended during FlashCopy

```
DSNU105I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = LOAD.ORDER
                PROCESSING UTILITY STATEMENT 1
                UTILITY = LOAD
                PHASE = RELOAD   COUNT = 100557
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = ACTIVE
DSNU111I  -DB8X DSNUGDIS - SUBPHASE = SORTIN COUNT = 200192
DSNU111I  -DB8X DSNUGDIS - SUBPHASE = SORTIN COUNT = 99328
DSNU111I  -DB8X DSNUGDIS - SUBPHASE = BUILD COUNT = 0
DSNU111I  -DB8X DSNUGDIS - SUBPHASE = BUILD COUNT = 0
DSNU111I  -DB8X DSNUGDIS - SUBPHASE = SORTIN COUNT = 99840
DSNU111I  -DB8X DSNUGDIS - SUBPHASE = BUILD COUNT = 0
DSNU105I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = REORG01
                PROCESSING UTILITY STATEMENT 2
                UTILITY = REORG
                PHASE = SORTBLD   COUNT = 331658
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = ACTIVE
 DSNU111I  -DB8X DSNUGDIS - SUBPHASE = BUILD COUNT = 165829
 DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

Example 15-7 shows active threads during FlashCopy that are suspended by SET LOG SUSPEND command. We elected to issue the Display Thread to get an idea of the active units of work before the -SET LOG SUSPEND command. They will be resolved during the subsequent DB2 restart at the recovery site. It is not necessary for the successful execution of this scenario, but we found it helpful.

Example 15-7 The threads suspended during FlashCopy

```
DSNV401I  -DB8X DISPLAY THREAD REPORT FOLLOWS -
DSNV402I  -DB8X ACTIVE THREADS -
NAME     ST A   REQ ID           AUTHID   PLAN     ASID TOKEN
UTILITY  T      523 PAOLOR31     PAOLOR3  DSNUTIL  0029    26
UTILITY  T  *   372 PAOLOR31     PAOLOR3  DSNUTIL  0029    34
UTILITY  T  *   241 PAOLOR30     PAOLOR3  DSNUTIL  0027    11
UTILITY  T        5 PAOLOR30     PAOLOR3  DSNUTIL  0027    12
UTILITY  T        6 PAOLOR30     PAOLOR3  DSNUTIL  0027    13
UTILITY  T        5 PAOLOR30     PAOLOR3  DSNUTIL  0027    14
UTILITY  T        6 PAOLOR30     PAOLOR3  DSNUTIL  0027    15
UTILITY  T        5 PAOLOR30     PAOLOR3  DSNUTIL  0027    16
UTILITY  T        6 PAOLOR30     PAOLOR3  DSNUTIL  0027    17
```

```
TSO      T  *     3 PAOLOR3      PAOLOR3           047F    41
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I  -DB8X DSNVDT '-DIS THD' NORMAL COMPLETION
```

Workloads performed after FlashCopy

We created and populated tables as follows because we want to see if DB2 could handle any creates and LOG NO events during forward log recovery using RESTORE SYSTEM LOGONLY:

- Execute DDL in an existing database:
  - Create another table space (PAOLODB.TSNATION), a table (PAOLOR3.NATION) and two indexes in an existing database.
  - Load data into the table (PAOLOR3.NATION) using LOAD LOG NO.
  - Back up the table space (PAOLODB.TSNATION) using COPY SHRLEVEL REFERENCE.
- Use the REORG utility with the LOG NO COPYDDN attribute for a table space (PAOLODB.TSSUPLY).
- Execute DML to insert rows into one table.
- Use the LOAD utility with the LOG YES attribute for a table space (PAOLODB.TSLINEI).

15.5 Setting up the recovery site

In case of disaster, you can follow the steps described in the next section at the recovery site.

15.5.1 Restore the FlashCopy data sets to disk

Before restoring the dump, you must unallocate the ICF catalogs that reside on the volumes where the dump will be restored. We use the MODIFY CATALOG command as shown in Example 15-8.

Example 15-8 MODIFY CATALOG command to unallocate ICF catalogs

```
/F CATALOG,UNALLOCATE(UCAT.DB8XL)
/F CATALOG,UNALLOCATE(UCAT.DB8XD)
```

Attention: The assumption is that the volumes at the recovery site have been initialized with the same volume serial numbers (VOLID) as the ones used at the local site. If not, you should specify the COPYVOLID parameter.

We used the same volume serial numbers at the recovery site. In that case, the RESTORE command can be used without the COPYVOLID parameter, as the VOLID is the same on disk as the one on the tape. Example 15-9 shows sample JCL statements to restore a tape dump of a target volume without the COPYVOLID parameter.

Example 15-9 DFSMSdss RESTORE

```
//VDB8X00  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X00
//INDS     DD  UNIT=3490,DISP=OLD,LABEL=4,
//             VOL=(,RETAIN,SER=(PRB000,PRB001,PRB002,PRB003)),
//             SER=(PRB000,PRB001,PRB002,PRB003)),
//             DSN=BACKUP.DB8X00
//SYSIN    DD  *
  RESTORE INDDNAME(INDS) OUTDDNAME(OUTVOL) -
          FULL PURGE -
          CANCELERROR ADMIN
```

The PURGE parameter is required in the restore job because the VVDS name on the target volume does not match the target volume serial number (as a result of the DUMPCONDITIONING parameter that was specified for the full volume COPY).

15.5.2 Keep the current archive log

We use dual archive logging. The second archive log (in most cases, on tape) is intended as a backup, or can be sent to the recovery site in preparation for disaster recovery.

Tip: To make recovering from the COPY2 archive log data set faster at the recovery site, use the ARC2FRST (DSNZPARM parameter) to specify that the COPY2 archive log should be read first. Otherwise, DB2 always attempts to read the primary copy of the archive log data set first.

We restored all the second archive logs to disk as shown in Example 15-10. It eliminates multiple tape mounts and speeds up recovery.

Example 15-10 Restore the second archive logs to disk

```
DSLIST - Data Sets Matching DB8XU                              Row 85 of 181
Command ===>                                                 Scroll ===> CSR

Command - Enter "/" to select action                Message           Volume
-------------------------------------------------------------------------------
         DB8XU.ARCHLOG1.D2004198.T1412140.A0000043                     DB8X02+
         DB8XU.ARCHLOG1.D2004198.T1412140.B0000043                     DB8X02+
         DB8XU.ARCHLOG1.D2004198.T1422270.A0000044                     DB8X02+
         DB8XU.ARCHLOG1.D2004198.T1422270.B0000044                     DB8X02+


DSLIST - Data Sets Matching DB8XU                              Row 173 of 181
Command ===>                                                 Scroll ===> CSR

Command - Enter "/" to select action                Message           Volume
-------------------------------------------------------------------------------
         DB8XU.ARCHLOG2.D2004198.T1412140.A0000043                     DB8X02+
         DB8XU.ARCHLOG2.D2004198.T1412140.B0000043                     DB8X02+
         DB8XU.ARCHLOG2.D2004198.T1422270.A0000044                     DB8X02+
         DB8XU.ARCHLOG1.D2004198.T1422270.B0000044                     DB8X02+
         DB8XU.ARCHLOG2.D2004198.T1427133.A0000045                     DB8X02
         DB8XU.ARCHLOG2.D2004198.T1429419.A0000046                     DB8X02
         DB8XU.ARCHLOG2.D2004198.T1432116.A0000047                     DB8X02
         DB8XU.ARCHLOG2.D2004198.T1434413.A0000048                     DB8X02
         DB8XU.ARCHLOG2.D2004198.T1434413.B0000048                     DB8X02
```

15.5.3 Recover the BSDS

This scenario bases recovery on the latest available archive log and assumes that it has arrived at the recovery site.

You must determine the fully qualified name and RBA of the archive log at the recovery site. At the local site, your automation can schedule search for the message DSNJ139I LOG OFFLOAD TASK ENDED. Once the message is issued, you can release the batch job to run your DSNJU004 report. The DSNJU004 report tape is sent to the recovery site with the archive log tape.

If data sharing, the BSDS data sets on each data sharing member need to be restored.

REPRO to restore BSDS from the last archive log

We copied the second archive logs to disk before REPRO as shown in Example 15-10. In Example 15-11, we restore BSDS from the last archive log file.

Example 15-11 Repro to restore BSDS from the last archive log

```
//REPRO    EXEC PGM=IDCAMS
//INPUT    DD   DISP=SHR,DSN=DB8XU.ARCHLOG2.D2004198.T1434413.B0000048
//OUTPUT   DD   DISP=SHR,DSN=DB8XU.BSDS01
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 REPRO INFILE(INPUT) OUTFILE(OUTPUT) REUSE
```

Attention: If you do not use REUSE option for REPRO, it will be in error as follows:

```
IDC3302I  ACTION ERROR ON DB8XU.BSDS01
IDC3308I ** DUPLICATE RECORD - KEY FOLLOWS:
000000  00000000
```

Register the last archive log in the BSDS using DSNJU003

Example 15-12 lists the RBA you will need to register in the DSNJU003 job. We found it in SYSLOG at local site. If you send DSNJU004 report to the recovery site, you can find it on the print log map utility output at the recovery site.

Example 15-12 Find the last archive log in syslog

```
DSNJ003I  -DB8X DSNJOFF3 FULL ARCHIVE LOG VOLUME 796
DSNAME=DB8XU.ARCHLOG1.D2004198.T1434413.A0000048,
STARTRBA=000063240000, ENDRBA=0000653FFFFF, STARTTIME=BB86A525B02C,
ENDTIME=BB86A5B40F14, UNIT=3390, COPY1VOL=DB8X02, VOLSPAN=00,
CATLG=YES
DSNJ003I  -DB8X DSNJOFF3 FULL ARCHIVE LOG VOLUME 797
DSNAME=DB8XU.ARCHLOG2.D2004198.T1434413.A0000048,
STARTRBA=000063240000, ENDRBA=0000653FFFFF, STARTTIME=BB86A525B02C,
ENDTIME=BB86A5B40F14, UNIT=3390, COPY2VOL=DB8X02, VOLSPAN=00,
CATLG=YES
DSNJ139I  -DB8X LOG OFFLOAD TASK ENDED
```

The BSDS that you restored from the last archive log does not contains information about the archive log. You should register the last archive log in BSDS as shown in Example 15-13. If data sharing, you must also provide its starting (STARTLRSN) and ending LRSNs (ENDLRSN) in the NEWLOG statement for the Change Log Inventory utility.

Example 15-13 Register the last archive log in BSDS

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 NEWLOG DSNAME=DB8XU.ARCHLOG2.D2004198.T1434413.A0000048,
 COPY2VOL=DB8X02,
 STARTRBA=000063240000,ENDRBA=0000653FFFFF,UNIT=3390,CATALOG=YES
```

Adjust the active logs using DSNJU003

The BSDS contains information about the active logs at your local site. You want to delete the old active logs and add the new active logs as shown in Example 15-14.

Example 15-14 Register the active logs in BSDS

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
   DELETE DSNAME=DB8XU.LOGCOPY1.DS01
   DELETE DSNAME=DB8XU.LOGCOPY1.DS02
   DELETE DSNAME=DB8XU.LOGCOPY1.DS03
   DELETE DSNAME=DB8XU.LOGCOPY2.DS01
   DELETE DSNAME=DB8XU.LOGCOPY2.DS02
   DELETE DSNAME=DB8XU.LOGCOPY2.DS03
   NEWLOG DSNAME=DB8XU.LOGCOPY1.DS01,COPY1
   NEWLOG DSNAME=DB8XU.LOGCOPY1.DS02,COPY1
   NEWLOG DSNAME=DB8XU.LOGCOPY1.DS03,COPY1
   NEWLOG DSNAME=DB8XU.LOGCOPY2.DS01,COPY2
   NEWLOG DSNAME=DB8XU.LOGCOPY2.DS02,COPY2
   NEWLOG DSNAME=DB8XU.LOGCOPY2.DS03,COPY2
```

Use DSNJU004 to list the BSDS

Example 15-15 lists the information you updated in the BSDS.

Example 15-15 DSNJU004 output

```
DSNJCNVB CONVERSION PROGRAM HAS NOT RUN   DDNAME=SYSUT1
   LOG MAP OF BSDS DATA SET COPY 1, DSN=DB8XU.BSDS01
   LTIME INDICATES LOCAL TIME, ALL OTHER TIMES ARE GMT.
         DATA SHARING MODE IS OFF
         SYSTEM TIMESTAMP   - DATE=2004.198  LTIME=14:34:27.57
         UTILITY TIMESTAMP  - DATE=2004.198  LTIME=15:00:55.08
         VSAM CATALOG NAME=DB8XD
         HIGHEST RBA WRITTEN       0000653FE401  2004.198  18:34:41.3
         HIGHEST RBA OFFLOADED     00006323FFFF
         RBA WHEN CONVERTED TO V4  000000000000

ACTIVE LOG COPY 1 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATIO
 --------------------  --------------------  --------  ----- -------------------
                 EMPTY DATA SET              2004.198  15:00 DSN=DB8XU.LOGCOPY1.DS01
 0000.000  00:00:00.0  0000.000  00:00:00.0  PASSWORD=(NULL) STATUS=NEW, REUSABLE
                 EMPTY DATA SET              2004.198  15:00 DSN=DB8XU.LOGCOPY1.DS02
```

```
 0000.000  00:00:00.0  0000.000  00:00:00.0  PASSWORD=(NULL) STATUS=NEW, REUSABLE
                 EMPTY DATA SET              2004.198  15:00 DSN=DB8XU.LOGCOPY1.DS03
 0000.000  00:00:00.0  ........  ..........  PASSWORD=(NULL) STATUS=NEW, REUSABLE


ARCHIVE LOG COPY 2 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
   000061080000           00006323FFFF       2004.198  14:32 DSN=DB8XU.ARCHLOG2.D2004198.T1432116.A0000047
 2004.198  18:29:41.8  2004.198  18:32:11.5                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
   000063240000           0000653FFFFF       2004.198  14:59 DSN=DB8XU.ARCHLOG2.D2004198.T1434413.A0000048
 0000.000  00:00:00.0  0000.000  00:00:00.0                  PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED


                          CHECKPOINT QUEUE
                      19:01:08 JULY 16, 2004
      TIME OF CHECKPOINT       18:34:41 JULY 16, 2004
      BEGIN CHECKPOINT RBA            000065413000
      END CHECKPOINT RBA              0000654213A6
      TIME OF CHECKPOINT       18:32:11 JULY 16, 2004
      BEGIN CHECKPOINT RBA            000063249000
      END CHECKPOINT RBA              000063254452
      TIME OF CHECKPOINT       18:29:41 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00006108D000
      END CHECKPOINT RBA              000061094554
      TIME OF CHECKPOINT       18:27:13 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005EED308A
      END CHECKPOINT RBA              00005EEE6906
      TIME OF CHECKPOINT       18:22:26 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005CD105D0
      END CHECKPOINT RBA              00005CD1C0B8
      TIME OF CHECKPOINT       18:12:24 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005C657644
      END CHECKPOINT RBA              00005C65BCB8
      TIME OF CHECKPOINT       18:12:13 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005AB728FF
      END CHECKPOINT RBA              00005AC78236
      TIME OF CHECKPOINT       18:09:09 JULY 16, 2004
      BEGIN CHECKPOINT RBA            000058988000
      END CHECKPOINT RBA              0000589998B0
```

Make a copy of the restored BSDS to the second BSDS

If the boot strap data sets do not match, you must copy your updated BSDS to the non-updated BSDS as shown in Example 15-16.

Example 15-16 Repro to make the second BSDS from the restored BSDS

```
//REPRO    EXEC PGM=IDCAMS
//INPUT    DD   DISP=OLD,DSN=DB8XU.BSDS01
//OUTPUT   DD   DISP=OLD,DSN=DB8XU.BSDS02
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 REPRO INFILE(INPUT) OUTFILE(OUTPUT) REUSE
```

15.5.4 DSNJU003 to create a SYSPITR CRCR

The new “SYSPITR” option of DSNJU003 establishes a log truncation point in preparation for running the RESTORE SYSTEM utility. Each member of a data sharing group must have a SYSPITR CRCR created with the same log truncation point (LRSN).

It is your responsibility to find the point-in-time (PIT) that you want to go back to, and that RBA is specified in the SYSPITR conditional restart record. We suggest that you use either *the RBA of last valid log record* or *the RBA of the latest checkpoint END* as a SYSPITR. You can get an additional logdata applied by finding the RBA of last valid log record as opposed to using the RBA of the latest Checkpoint END as shown in Figure 15-4.

For the detailed information on determining the SYSPITR RBA, refer to 15.6.1, “How to find the SYSPITR RBA on the last archive log” on page 275.

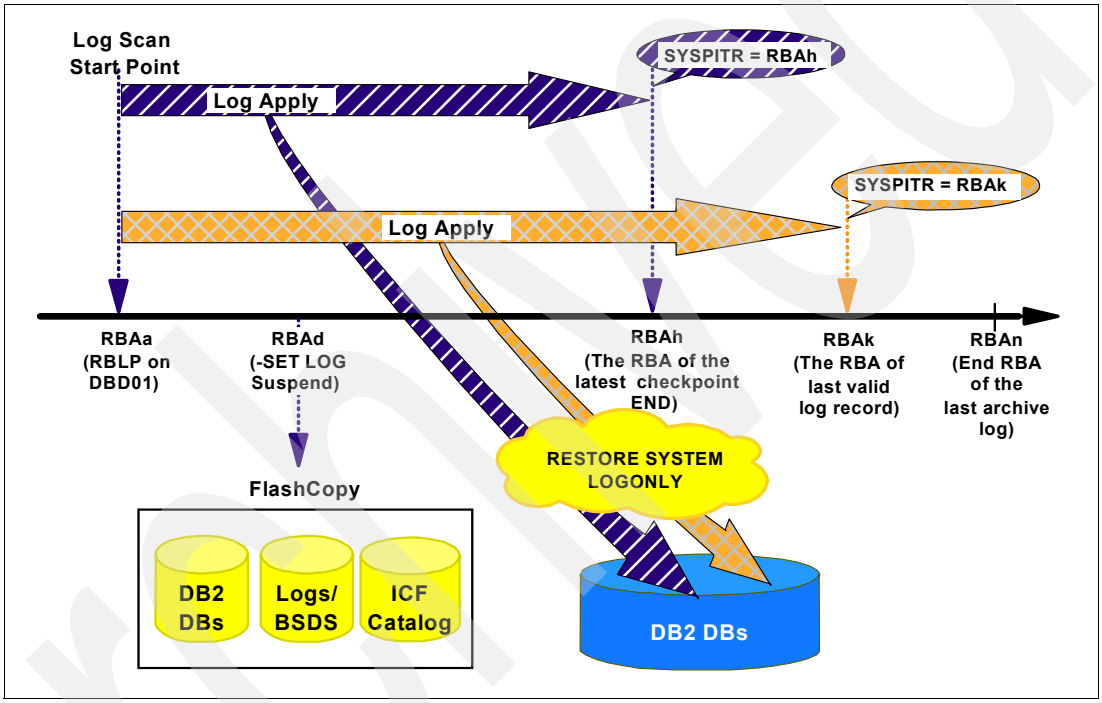


Figure 15-4 Establish the SYSPITR CRCR

In Figure 15-4, the RBAa (RBLP) is ‘000058988000’X, RBAd (-SET LOG SUSPEND) is ‘00005C65CE60’X, RBAh (the latest checkpoint END) is ‘000063254452’X, RBAk (the last valid log record) is ‘0000653FE401’X, and RBAn (the last archive log END) is ‘0000653FFFFF’X.

In Example 15-17, we use the RBA of the last valid log record *‘0000653FE401’X* found in Example 15-36 on page 277 and Example 15-38 on page 279 as a SYSPITR.

Example 15-17 DSNJU003 to create a SYSPITR CRCR

```
//BSDS1   EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DB8XU.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DB8XU.BSDS02
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 CRESTART CREATE,SYSPITR=0000653FE401
```

Example 15-18 shows CRCR inserted from the previous DSNJU003.

Example 15-18 Restart Control Record

```
CONDITIONAL RESTART CONTROL RECORD
                      19:16:48 JULY 16, 2004
 **** ACTIVE CRCR RECORD ****
 CRCR IDENTIFIER  0001
      USE COUNT   0
      RECORD STATUS
          CRCR ACTIVE
          CRCR NOT USED
      PROCESSING STATUS
          FORWARD = YES
          BACKOUT = YES
      SYSPITR SYSTEM LEVEL RECOVERY MODE RESTART
      STARTRBA                        NOT SPECIFIED
      ENDRBA                          NOT SPECIFIED
      ENDLRSN                         0000653FE401
      EARLIEST REQUESTED RBA          000000000000
      FIRST LOG RECORD RBA            000000000000
      ORIGINAL CHECKPOINT RBA         000000000000
      NEW CHECKPOINT RBA (CHKPTRBA)   000063249000
      END CHECKPOINT RBA              000063254452
      CRCR CREATED             19:16:08 JULY 16, 2004
      TIME OF CHECKPOINT       18:32:11 JULY 16, 2004
      RESTART PROGRESS                 STARTED     ENDED
                                      =======     =====
         CURRENT STATUS REBUILD         NO         NO
         FORWARD RECOVERY PHASE         NO         NO
         BACKOUT RECOVERY PHASE         NO         NO
```

15.5.5 Restart DB2 with a SYSPITR CRCR

A DB2 system goes into System Recover Pending mode after a conditional restart is performed, using a special type of conditional restart record called a “SYSPITR” as shown in Example 15-19.

The System Recover Pending mode means that:

- Only the RESTORE SYSTEM utility is allowed to execute.
- The DB2 data remains unavailable until the RESTORE SYSTEM utility has completed.
- DB2 has to be recycled to reset the System Recover Pending mode.

In a data sharing environment, each non-dormant member needs to be restarted with the SYSPITR CRCR, with all members specifying the same log truncation LRSN. After the conditional restart of all members is complete, you can run the RESTORE SYSTEM utility on (only) one of the members. This utility has to start and complete on the same member. It cannot be restarted on a different member. After RESTORE SYSTEM successfully completes, you must bring down all members to reset the System Recover Pending mode.

Example 15-19 Restart DB2 with a SYSPITR CRCR

```
$HASP373 DB8XMSTR STARTED
 IEF403I DB8XMSTR - STARTED - TIME=15.22.18 - ASID=0477 - SC64
 DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE NAME IS DSNZPARM
 S DB8XIRLM
 DSNY001I  -DB8X SUBSYSTEM STARTING
 DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.198 14:34:27.57
```

```
*264 DSNJ245I  -DB8X CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT RBA 0000653FE401.
REPLY Y TO CONTINUE, N
          TO CANCEL
 R 264,Y
 DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  326
 DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS01,
 STARTRBA=0000653FF000,ENDRBA=0000675BEFFF
 DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  327
 DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS01,
 STARTRBA=0000653FF000,ENDRBA=0000675BEFFF
 DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  328
 STARTRBA=0000653FF000
 S DB8XDBM1
 S DB8XDIST
 DSNR001I  -DB8X RESTART INITIATED
 DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=000063249000
 DSNR004I  -DB8X RESTART...UR STATUS COUNTS  341
 IN COMMIT=0, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
 DSNR007I  -DB8X RESTART...STATUS TABLE  342
 T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY TIME
 - -------- ------------ -------- -------- - ------------ --- -----
 B UTILITY  PAOLOR3L     PAOLOR3  DSNUTIL  F 000064F4FD61 198 14:34:20
 DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  343
 RECOVERY
 IN COMMIT=0, INDOUBT=0
 DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  344
 RECOVERY
 INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
*DSNR050I  -DB8X DSNRRPRC DB2 STARTED IN SYSTEM RECOVER PENDING MODE
 DSNR002I  -DB8X RESTART COMPLETED
 DSNY014I  -DB8X DSNYSTRT DB2 WAS STARTED WITH ACCESS(MAINT)
 -DB8XRECOVER POSTPONED
 DSN9022I  -DB8X DSNYASCP 'STA DB2' NORMAL COMPLETION
 DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
 DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
```

Example 15-20 shows the information changed during system level point-in-time recovery. The information changed includes the highest RBA written, the RBA of active log, the status of the last archive log now changed to “truncated”, and CRCR.

Example 15-20 DSNJU004 after DB2 restart

```
LOG MAP OF BSDS DATA SET COPY 1, DSN=DB8XU.BSDS01
LTIME INDICATES LOCAL TIME, ALL OTHER TIMES ARE GMT.
      DATA SHARING MODE IS OFF
      SYSTEM TIMESTAMP   - DATE=2004.198  LTIME=15:22:21.68
      UTILITY TIMESTAMP  - DATE=2004.198  LTIME=15:16:08.67
      VSAM CATALOG NAME=DB8XD
      HIGHEST RBA WRITTEN       0000653FEFFF  1900.001  00:00:00.0
      HIGHEST RBA OFFLOADED     0000653FEFFF
      RBA WHEN CONVERTED TO V4  000000000000
ACTIVE LOG COPY 1 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
                 EMPTY DATA SET              2004.198  15:00 DSN=DB8XU.LOGCOPY1.DS02
 0000.000  00:00:00.0  0000.000  00:00:00.0  PASSWORD=(NULL) STATUS=NEW, REUSABLE
                 EMPTY DATA SET              2004.198  15:00 DSN=DB8XU.LOGCOPY1.DS03
 0000.000  00:00:00.0  0000.000  00:00:00.0  PASSWORD=(NULL) STATUS=NEW, REUSABLE
   0000653FF000           0000675BEFFF       2004.198  15:00 DSN=DB8XU.LOGCOPY1.DS01
 2004.198  18:34:40.8  ........  ..........  PASSWORD=(NULL) STATUS=NOTREUSABLE
```

```
ARCHIVE LOG COPY 2 DATA SETS
    START RBA/TIME         END RBA/TIME        DATE    LTIME DATA SET INFORMATIO
 --------------------  --------------------  --------  ----- --------------------
  000061080000           00006323FFFF       2004.198  14:32 DSN=DB8XU.ARCHLOG2.D2004198.T1432116.A0000047
2004.198  18:29:41.8  2004.198  18:32:11.5                   PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED
  000063240000           0000653FEFFF       2004.198  14:59 DSN=DB8XU.ARCHLOG2.D2004198.T1434413.A0000048
0000.000  00:00:00.0  0000.000  00:00:00.0                   PASSWORD=(NULL) VOL=DB8X02 UNIT=3390
                                                             CATALOGUED STATUS=TRUNCATED
                CONDITIONAL RESTART CONTROL RECORD
                     19:27:29 JULY 16, 2004
**** ACTIVE CRCR RECORD ****
     NO CRCR RECORDS ARE ACTIVE
****
CRCR IDENTIFIER  0001
     USE COUNT   1
     RECORD STATUS
         CRCR NOT ACTIVE
         SUCCESSFUL RESTART
     PROCESSING STATUS
         FORWARD = YES
         BACKOUT = YES
     SYSPITR SYSTEM LEVEL RECOVERY MODE RESTART
     STARTRBA                        NOT SPECIFIED
     ENDRBA                          NOT SPECIFIED
     ENDLRSN                         0000653FE401
     TRUNCATION RBA (FROM ENDLRSN)   0000653FF3A8
     EARLIEST REQUESTED RBA          000000000000
     FIRST LOG RECORD RBA            000000000000
     ORIGINAL CHECKPOINT RBA         000065413000
     NEW CHECKPOINT RBA (CHKPTRBA)   000063249000
     END CHECKPOINT RBA              000063254452
     CRCR CREATED             19:16:08 JULY 16, 2004
     BEGIN RESTART            19:22:21 JULY 16, 2004
     END RESTART              19:23:20 JULY 16, 2004
     TIME OF CHECKPOINT       18:32:11 JULY 16, 2004
     RESTART PROGRESS                 STARTED     ENDED
                                       =======     =====
         CURRENT STATUS REBUILD         YES         YES
         FORWARD RECOVERY PHASE         YES         YES
         BACKOUT RECOVERY PHASE         YES         YES
```

15.5.6 RESTORE SYSTEM LOGONLY

If a method other than the BACKUP SYSTEM utility was used to copy the system, restore the data manually and use RESTORE SYSTEM LOGONLY. This option can be used with z/OS V1R3 and above. In this scenario, we copy the system by using FlashCopy and SET LOG SUSPEND / SET LOG RESUME.

The RESTORE SYSTEM utility recovers a DB2 system to an arbitrary point-in-time. RESTORE SYSTEM automatically handles any creates, drops, and LOG NO events that might have occurred between the backup and the recovery point-in-time.

You also have to make sure that all archive logs, between the time of the system level backup (in this case, FlashCopy) and the PIT that you want to recover to, are available to the RESTORE SYSTEM utility.

DSN1PRNT of DBD01

You can retrieve RBLP in the DBD01 header page using DSN1PRNT utility as shown in Example 15-21.

Example 15-21 DSN1PRNT to retrieve RBLP in DBD01

```
//PRINT   EXEC PGM=DSN1PRNT,PARM='PRINT(0),FORMAT'
//STEPLIB  DD DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=DB8XD.DSNDBC.DSNDB01.DBD01.I0001.A001
```

Example 15-22 lists the RBLP that is updated by SET LOG SUSPEND command. RESTORE SYSTEM utility uses it to determine the log scan starting point.

Example 15-22 DSN1PRNT output sample in DBD01

```
DSN1999I START OF DSN1PRNT FOR JOB PAOLOR3P PRINT
DSN1989I DSN1PRNT IS PROCESSED WITH THE FOLLOWING OPTIONS:
          4K/NO IMAGECOPY/NUMPARTS =   0/   FORMAT/NO EXPAND/NO SWONLY/   PRINT/NO VALUE/     /   /
          DSSIZE=   /PIECESIZ=    /       /
DSN1998I INPUT  DSNAME = DB8XD.DSNDBC.DSNDB01.DBD01.I0001.A001       , VSAM


PAGE: # 00000000 ----------------------------------------------------------------
HEADER PAGE:  PGCOMB='00'X  PGLOGRBA='000000003945'X  PGNUM='00000000'X  PGFLAGS='18'X
              HPGOBID='0001001F'X  HPGHPREF='00001680'X  HPGCATRL='D3'X  HPGREL='D3'X  HPGZLD='00'X
              HPGCATV='50'X  HPGTORBA='000000000000'X  HPGTSTMP='00000000000000000000'X
              HPGSSNM='....'  HPGFOID='0010'X  HPGPGSZ='1000'X  HPGSGSZ='0000'X  HPGPARTN='0000'X
              HPGZ3PNO='000000'X  HPGZNUMP='00'X  HPGTBLC='0002'X  HPGROID='0000'X
              HPGZ4PNO='00000000'X  HPGMAXL='0000'X  HPGNUMCO='0000'X  HPGFLAGS='2810'X
              HPGCONTM='00000000000000000000'X  HPGSGNAM='........'  HPGVCATN='........'
              HPGRBRBA='00005AB728FF'X  HPGLEVEL='00005C5A4773'X  HPGPLEVL='00005494DE4F'X
              HPGCLRSN='000000000060'X  HPGSCCSI='0025'X  HPGDCCSI='0000'X  HPGMCCSI='0000'X
              HPGFLAG2='00'X  HPGEPOCH='0000'X  HPGRBLP='000058988000'X  HPGDNUMB='1F'X
              HPGDNUMC='0007'X  HPGDFSG='00000000'X  HPGDLSG='00000000'X  FOEND='E'


DSN1994I DSN1PRNT COMPLETED SUCCESSFULLY,  00000001 PAGES PROCESSED
```

RESTORE SYSTEM LOGONLY

Restriction: You can only run RESTORE SYSTEM utility at this point that DB2 started in Access Maint and System Recover Pending Mode. Only access allowed is INSTALL SYSADM authority.

The RESTORE SYSTEM utility can only be executed when the system is in System Recover Pending mode. Example 15-23 is the sample control statement of RESTORE SYSTEM using LOGONLY.

Example 15-23 RESTORE SYSTEM LOGONLY

```
//STEP1   EXEC DSNUPROC,PARM='DB8X,DSNTEL'
//SYSIN    DD  *
  RESTORE SYSTEM LOGONLY
```

The RESTORE SYSTEM LOGONLY performs log apply function only. It uses the recovery base log point (RBLP) that is recorded in DBD01 to determine the starting point of the log apply phase and it also uses FLA to recover objects in parallel as shown in Example 15-24.

Attention: During our project, early versions of APAR PQ93156 and PQ89297 were applied in order to activate FLA during RESTORE.

Example 15-24 RESTORE SYSTEM output using LOGONLY

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNTEL
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RESTORE SYSTEM LOGONLY
DSNU1604I -DB8X DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT =
X'000058988000'.
DSNU1629I -DB8X DSNUVARL - DB2 PUT ONE OR MORE OBJECTS INTO THE RECOVER-PENDING STATE, THE
REBUILD-PENDING STATE, OR THE LOGICAL PAGE LIST DURING THE LOG APPLY PHASE.
DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME =  00:01:03.
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

Recycle DB2 to reset system recover pending mode

The DB2 system and DB2 data sharing members will need to be restarted after the RESTORE SYSTEM utility has been executed. This will reset the system recover pending mode as shown in Example 15-25.

Example 15-25 start DB2 to remove access maintenance mode

```
DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE NAME IS DSNZPARM
S DB8XIRLM
DSNY001I  -DB8X SUBSYSTEM STARTING
DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.198 17:12:55.33
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  603
DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS01,
STARTRBA=0000653FF000,ENDRBA=0000675BEFFF
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  604
DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS01,
STARTRBA=0000653FF000,ENDRBA=0000675BEFFF
DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  605
STARTRBA=000065519000
S DB8XDBM1
S DB8XDIST
DSNR001I  -DB8X RESTART INITIATED
DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=0000655157CC
DSNR004I  -DB8X RESTART...UR STATUS COUNTS  618
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  619
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  620
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR002I  -DB8X RESTART COMPLETED
-DB8XRECOVER POSTPONED
DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
```

15.5.7 Recover all objects in RECP or RBDP status

As mentioned before, the RESTORE SYSTEM utility uses the recovery base log point (RBLP) that is recorded in DBD01 to determine the starting point of the log apply phase after the data is restored. The log apply phase uses the fast log apply (FLA) function to recover objects in parallel.

Attention: The RESTORE SYSTEM utility handles CREATEs and LOG NO events, but LOG NO marks them in RECP or RBDP. Even for LOAD LOG YES operation, DB2 will not log index updates. All indexes involved in LOAD LOG YES or LOG NO operation will be marked in RBDP/RECP/PSRBD state. CREATE INDEX will not log index changes. For an index that was created between the backup and the PIT recovery point, it will be marked in RBDP/RECP/PSRBD if the target table was not empty at the time the CREATE INDEX was issued.

When the consistency for LOG NO utilities is established during the logapply phase of RESTORE SYSTEM, a log record that represents the open of a table space or index space with recovery(no) is encountered. In this case, table spaces will have to be put in RECP state, and index spaces will have to be put in either RECP or RBDP state, depending on their COPY attribute.

You should decide what you use to recover all objects in RECP. There are two choices. The one is RECOVER utility and the other is REPAIR with SET NORCVRPEND. The RECOVER is better but you should send the image copies to recovery site that were made after FlashCopy. And be careful using REPAIR. Improper use can damage the data even further. And then all indexes in RBDP should be rebuilt from the table data.

Display and terminate any active utilities

Example 15-26 displays a utility in progress. You must terminate it since you cannot restart it.

Example 15-26 -DIS UTIL(*) output

```
DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = LOAD.LINEITEM
                PROCESSING UTILITY STATEMENT 1
                UTILITY = LOAD
                PHASE = RELOAD   COUNT = 1725359
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = STOPPED
DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

After terminating the utility, there is no active utility, as shown in Example 15-27.

Example 15-27 Display any active utilities

```
DSNU112I  -DB8X DSNUGDIS - NO AUTHORIZED UTILITY FOUND FOR UTILID = *
DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

Display restricted objects

Example 15-28 lists the objects that DB2 was not able to recover as part of RESTORE SYSTEM LOGONLY. They need to be recovered or rebuilt.

Example 15-28 -DIS DB(*) SP(*) RESTRICT output

```
DSNT360I  -DB8X ************************************
DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                *    RESTRICTED
DSNT360I  -DB8X ************************************
DSNT362I  -DB8X     DATABASE = PAOLODB  STATUS = RW
                   DBD LENGTH = 20180
DSNT397I  -DB8X
```

```
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
TSLINEI  TS         RW,RECP,UTUT
TSNATION TS         RW,RECP
TSSUPLY  TS         RW,RECP
PXC#CKNK IX         RW,RBDP
PXL#OKSD IX         RW,RBDP
PXN#NKNN IX         RW,RBDP
PXO#OKOD IX         RW,RBDP
PXP#PKPT IX         RW,RBDP
PXS#SKNK IX         RW,RBDP
SXL#PKSK IX         RW,RBDP
SXP@SZPT IX         RW,RBDP
SXRI#CUS IX         RW,RBDP
SXRI#ITM IX         RW,RBDP
SXRI#NAT IX         RW,RBDP
SXRI#ORD IX         RW,RBDP
SXRI#PRT IX         RW,RBDP
SXRI#SUP IX         RW,RBDP
UXC@NKCK IX         RW,RBDP
UXO#CLOK IX         RW,RBDP
UXO$CKOK IX         RW,RBDP
******* DISPLAY OF DATABASE PAOLODB  ENDED      **********************
DSN9022I  -DB8X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

Recover all objects in RECP or RBDP status

A LOAD utility was running against a table space PAOLODB.TSLINEI when the last archive log was offloading and this table space has been put in RECP by the restart processing. Example 15-29 shows recovery of this table space and all indexes in RBDP that reference it.

Example 15-29 Recover objects used by utility in progress

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR3.RECOV
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RECOVER TABLESPACE PAOLODB.TSLINEI
DSNU510I  -DB8X DSNUCALA - NO GOOD FULL IMAGE COPY DATA SET FOR RECOVERY OF
                     TABLESPACE PAOLODB.TSLINEI
DSNU513I  -DB8X DSNUCALA - RECOVER UTILITY LOG APPLY RANGE IS RBA 00005DA2DDFE LRSN
00005DA2DDFE TO RBA 00005DA2FBE6 LRSN 00005DA2FBE6
DSNU513I  -DB8X DSNUCALA - RECOVER UTILITY LOG APPLY RANGE IS RBA 00005DA30482 LRSN
00005DA30482 TO RBA 000065538599 LRSN 000065538599
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005DF56000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005E469000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005E97C3FF
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005EE8F000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005F3A847F
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005F8BB000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00005FDCE0C7
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 0000602E006F
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 0000607F3077
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000060D065FE
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00006121D3F7
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00006172F000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000061C42219
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000062155000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000062668000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000062B7B000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 00006308E000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 0000635A517A
```

```
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000063AB8000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000063FCB000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 0000644DE61B
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 0000649F1000
DSNU579I  -DB8X DSNUCACL - RECOVER UTILITY LOG APPLY AT LOGPOINT 000064F0337C
DSNU1510I   DSNUCBDR - LOG APPLY PHASE COMPLETE, ELAPSED TIME = 00:00:10
DSNU500I    DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:10
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0

DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNTEA
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  REBUILD INDEX(PAOLOR3."PXL#OKSDRFSKEPDC",
PAOLOR3."SXL#PKSKOKEPDSQN", PAOLOR3."SXRI#ITM")
DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 7
DSNU555I  -DB8X DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=5195586
DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:16
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=1731862 FOR INDEX
PAOLOR3.SXRI#ITM
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=1731862 FOR INDEX
PAOLOR3.PXL#OKSDRFSKEPDC
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=1731862 FOR INDEX
PAOLOR3.SXL#PKSKOKEPDSQN
DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 3
DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:12
```

The RESTORE SYSTEM utility handles LOG NO events and marks in RECP or RBDP for the objects. A table space PAOLODB.TSSUPLY has been put in RECP and its indexes in RBDP because we ran REORG with LOG NO after FlashCopy. Example 15-30 shows the recovery of a table space in RECP and its indexes in RBLP.

Example 15-30 Recover objects used by LOG NO event

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR3.RECOV
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RECOVER TABLESPACE PAOLODB.TSSUPLY
DSNU515I    DSNUCBAL - THE IMAGE COPY DATA SET PAOLOR3.TSSUPLY WITH DATE=200407
                       IS PARTICIPATING IN RECOVERY OF TABLESPACE PAOLODB.TSSUP
DSNU504I    DSNUCBMD - MERGE STATISTICS FOR TABLESPACE PAOLODB.TSSUPLY   -
                       NUMBER OF COPIES=1
                       NUMBER OF PAGES MERGED=388
                       ELAPSED TIME=00:00:01
DSNU1511I -DB8X DSNUCALA - FAST LOG APPLY WAS NOT USED FOR RECOVERY
DSNU1510I   DSNUCBDR - LOG APPLY PHASE COMPLETE, ELAPSED TIME = 00:00:00
DSNU500I    DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:01
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0

DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNTEA
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  REBUILD INDEX(PAOLOR3."PXS#SKNK", PAOLOR3."SXRI#SUP")
DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 5
DSNU555I  -DB8X DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=20000
DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:00
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=10000 FOR INDEX
PAOLOR3.PXS#SKNK
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=10000 FOR INDEX
PAOLOR3.SXRI#SUP
DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 2
DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:00
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

The RESTORE SYSTEM utility also handles CREATEs and DROPs. A table space PAOLODB.TSNATION was created, loaded with LOG NO and followed by COPY utility after FlashCopy. The table space was restored by the restart processing but it and its indexes set in RECP/RBDP because of LOAD with LOG NO. Example 15-31 shows the recovery of a table space in RECP and its indexes in RBLP.

Example 15-31 Recover objects created and loaded with LOG NO after FlashCopy

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR3.RECOV
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RECOVER TABLESPACE PAOLODB.TSNATION
DSNU515I    DSNUCBAL - THE IMAGE COPY DATA SET DB8X8.SYSCOPY.PAOLODB.TSNATION WITH
DATE=20040718 AND TIME=190351
                       IS PARTICIPATING IN RECOVERY OF TABLESPACE PAOLODB.TSNATION
DSNU504I    DSNUCBMD - MERGE STATISTICS FOR TABLESPACE PAOLODB.TSNATION -
                       NUMBER OF COPIES=1
                       NUMBER OF PAGES MERGED=3
                       ELAPSED TIME=00:00:00
DSNU1511I -DB8X DSNUCALA - FAST LOG APPLY WAS NOT USED FOR RECOVERY
DSNU1510I   DSNUCBDR - LOG APPLY PHASE COMPLETE, ELAPSED TIME = 00:00:00
DSNU500I    DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:00
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0

DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNTEA
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  REBUILD INDEX(PAOLOR3.PXNKNNNR, PAOLOR3.SXRI2NAT)
DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 5
DSNU555I  -DB8X DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=50
DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:00
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=25 FOR INDEX
PAOLOR3.PXNKNNNR
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=25 FOR INDEX
PAOLOR3.SXRI2NAT
DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 2
DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:00
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Example 15-32 rebuilds all indexes in RBDP that were used by the workload (as shown in Example 15-6 on page 259) suspended during FlashCopy.

Example 15-32 Rebuild all indexes used by workload suspended during FlashCopy

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DSNTEA
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  REBUILD INDEX(PAOLOR3.PXP#PKPTPN, PAOLOR3.SXP@SZPTPKMF,
PAOLOR3.SXRI#PRT)
DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 7
DSNU555I  -DB8X DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=497487
DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:02
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=165829 FOR INDEX
PAOLOR3.SXRI#PRT
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=165829 FOR INDEX
PAOLOR3.PXP#PKPTPN
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=165829 FOR INDEX
PAOLOR3.SXP@SZPTPKMF
DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 3
DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:01
DSNU050I    DSNUGUTC -  REBUILD INDEX(PAOLOR3.PXC#CKNKMS, PAOLOR3.SXRI#CUS,
PAOLOR3.UXC@NKCK)
DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 7
```

```
DSNU555I  -DB8X DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=450000
DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:01
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=150000 FOR INDEX
PAOLOR3.SXRI#CUS
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=150000 FOR INDEX
PAOLOR3.PXC#CKNKMS
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=150000 FOR INDEX
PAOLOR3.UXC@NKCK
DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 3
DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:01
DSNU050I    DSNUGUTC -  REBUILD INDEX(PAOLOR3.PXO#OKODCKSPOP, PAOLOR3.UXO#CLOKOD,
PAOLOR3.UXO$CKOKODSP,PAOLOR3.SXRI#ORD)
DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 9
DSNU555I  -DB8X DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=1346076
DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:04
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=336519 FOR INDEX
PAOLOR3.SXRI#ORD
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=336519 FOR INDEX
PAOLOR3.UXO$CKOKODSP
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=336519 FOR INDEX
PAOLOR3.UXO#CLOKOD
DSNU394I  -DB8X DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=336519 FOR INDEX
PAOLOR3.PXO#OKODCKSPOP
DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 4
DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:15
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

Validate that recovery was successful

All restricted objects are recovered successfully and there is no restriction for all databases as shown in Example 15-33. You can now verify your table data.

Example 15-33 Validate that recovery was successful

```
DSNT365I  -DB8X    NO DATABASES FOUND
DSN9022I  -DB8X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

15.6 Considerations

In this section, we describe special considerations that must be taken into account when using this solution.

15.6.1 How to find the SYSPITR RBA on the last archive log

When doing a conditional restart for system level point-in-time recovery, with DB2 V8, you always use the SYSPITR keyword for the CRESTART record using Change Log Inventory. You must specify an exact RBA that you want to use during a conditional restart as an RBA for the SYSPITR conditional restart record.

Note that this is contrasted to the standard conditional restart where, the *log truncation RBA* must be rounded to the next higher CI boundary (x"000"). For a data sharing environment, the exact LRSN was specified as the “log truncation” point for all members.

Important: It is your responsibility to find the point-in-time (PIT) that you want to go back to, and that LRSN-RBA is specified in the SYSPITR conditional restart record. You can choose one of the following LRSN-RBA values as a SYSPITR value:

1. The RBA of the latest Checkpoint END:
   a. Do a Print log map on the BSDS for the last archive you have.
   b. Find the Checkpoint Queue.
   c. Find the latest complete checkpoint on the last archive log (that is usually the second one in the queue).
   d. Find its END CHECKPOINT RBA.
2. The RBA of last valid log record:
   a. You may run DSN1LOGP using either RBASTART(xxxxxx) SUMMARY(ONLY) or RBASTART(yyyyyy) SUMMARY(NO) DBID(FFFF).
      i. Find RBA of the latest checkpoint BEGIN for xxxxxx.
      ii. Find RBA of the latest checkpoint END for yyyyyy.
   b. The run DSN1LOGP with the just found RBA as the STARTRBA.
   c. The output from DSN1LOGP will tell you the RBA of last valid log record.

Find the RBA of the latest Checkpoint END as a SYSPITR

You can find the RBA that should represent the latest complete checkpoint on the last archive log volume, and this corresponds to the second entry in the checkpoint queue as shown in Example 15-34.

If you want system level point-in-time recovery to the RBA of the latest Checkpoint END, you can find "*END CHECKPOINT RBA = 000063254452*" in this case.

Example 15-34 Find RBA of the latest checkpoint

```
CHECKPOINT QUEUE
                    19:01:08 JULY 16, 2004
      TIME OF CHECKPOINT       18:34:41 JULY 16, 2004
      BEGIN CHECKPOINT RBA            000065413000
      END CHECKPOINT RBA              0000654213A6
      TIME OF CHECKPOINT       18:32:11 JULY 16, 2004
      BEGIN CHECKPOINT RBA            000063249000
      END CHECKPOINT RBA              000063254452
      TIME OF CHECKPOINT       18:29:41 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00006108D000
      END CHECKPOINT RBA              000061094554
      TIME OF CHECKPOINT       18:27:13 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005EED308A
      END CHECKPOINT RBA              00005EEE6906
      TIME OF CHECKPOINT       18:22:26 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005CD105D0
      END CHECKPOINT RBA              00005CD1C0B8
      TIME OF CHECKPOINT       18:12:24 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005C657644
      END CHECKPOINT RBA              00005C65BCB8
      TIME OF CHECKPOINT       18:12:13 JULY 16, 2004
      BEGIN CHECKPOINT RBA            00005AB728FF
      END CHECKPOINT RBA              00005AC78236
      TIME OF CHECKPOINT       18:09:09 JULY 16, 2004
      BEGIN CHECKPOINT RBA            000058988000
      END CHECKPOINT RBA              0000589998B0
```

Find the RBA of last valid log record as a SYSPITR

When you want to find the RBA of last valid log record as a SYSPITR, you can run DSN1LOGP using either RBASTART(xxxxxx) SUMMARY(ONLY) or RBASTART(yyyyyy) SUMMARY(NO) DBID(FFFF). The latter option is nicer since the printed report is very small. However, the first one shows you URs in process at the SYPITR point, which might be useful.

DSN1LOGP using RBASTART(xxxxxx) SUMMARY(ONLY)

Example 15-35 is a sample of DSN1LOGP using RBASTART(xxxxxx) SUMMARY(ONLY) that xxxxxx is the RBA of the latest complete checkpoint BEGIN on the archive log as shown in Example 15-34.

Example 15-35 DSN1LOGP using RBASTART(xxxxxx) SUMMARY(ONLY)

```
//BSDS01   EXEC PGM=DSN1LOGP
//ARCHIVE  DD   DISP=OLD,DSN=DB8XU.ARCHLOG2.D2004198.T1434413.A0000048
//SYSPRINT DD   SYSOUT=*
//SYSSUMRY DD   SYSOUT=*
//SYSIN    DD   *
 RBASTART (000063249000) SUMMARY(ONLY)
```

Example 15-36 tells you the RBA of the last valid log record as a SYSPITR. You can find "*LAST LOG RBA ENCOUNTERED 0000653FE401*" in this case. The DSN1LOGP report also shows any activity that was ongoing during the archive processing. You should review any active threads and decide how you are going to deal with them.

Example 15-36 DSN1LOGP to find the RBA of last valid log record

```
DSN1212I DSN1LGRD FIRST LOG  RBA ENCOUNTERED 000063249000

===============================================================================
DSN1150I SUMMARY OF COMPLETED EVENTS

DSN1153I DSN1LSUM CHECKPOINT
               STARTRBA=000063249000  ENDRBA=000063254452  STARTLRSN=BB86A525B31F
ENDLRSN=BB86A525B687
               DATE=04.198  TIME=14:32:11

DSN1151I DSN1LPRT UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:31:52  DISP=COMMITTED           INFO=
             STARTRBA=000062DF968F  ENDRBA=00006338B4F5  STARTLRSN=000000000000
ENDLRSN=BB86A52AB6AB
             NID=* LUWID=*
             COORDINATOR=* PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI
                  DATABASE=0001=DSNDB01   PAGE SET=00AF=SYSUTILX

DSN1151I DSN1LPRT UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:32:17  DISP=COMMITTED           INFO=COMPLETE
             STARTRBA=00006338B529  ENDRBA=000063919A0D  STARTLRSN=BB86A52B2B96
ENDLRSN=BB86A54223B2
             NID=* LUWID=USIBMSC.SCPDB8X.BB86A3B06A55.0001
             COORDINATOR=* PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI
                  DATABASE=0001=DSNDB01   PAGE SET=00AF=SYSUTILX

DSN1151I DSN1LPRT UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:32:41  DISP=COMMITTED           INFO=COMPLETE
```

```
             STARTRBA=000063919A41  ENDRBA=000063EA772D  STARTLRSN=BB86A5429692
ENDLRSN=BB86A559914F
             NID=* LUWID=USIBMSC.SCPDB8X.BB86A3B06A55.0001
             COORDINATOR=* PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI
                  DATABASE=0001=DSNDB01   PAGE SET=00AF=SYSUTILX

DSN1151I DSN1LPRT UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:33:06  DISP=COMMITTED           INFO=COMPLETE
             STARTRBA=000063EA7761  ENDRBA=0000644354F5  STARTLRSN=BB86A55A07B2
ENDLRSN=BB86A5711ED7
             NID=* LUWID=USIBMSC.SCPDB8X.BB86A3B06A55.0001
             COORDINATOR=* PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI
                  DATABASE=0001=DSNDB01   PAGE SET=00AF=SYSUTILX

DSN1151I DSN1LPRT UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:33:31  DISP=COMMITTED           INFO=COMPLETE
             STARTRBA=000064435529  ENDRBA=0000649C2C81  STARTLRSN=BB86A57192FA
ENDLRSN=BB86A5889F65
             NID=* LUWID=USIBMSC.SCPDB8X.BB86A3B06A55.0001
             COORDINATOR=* PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI
                  DATABASE=0001=DSNDB01   PAGE SET=00AF=SYSUTILX

DSN1151I DSN1LPRT UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:33:55  DISP=COMMITTED           INFO=COMPLETE
             STARTRBA=0000649C3312  ENDRBA=000064F4FD2D  STARTLRSN=BB86A589124A
ENDLRSN=BB86A59FF5F1
             NID=* LUWID=USIBMSC.SCPDB8X.BB86A3B06A55.0001
             COORDINATOR=* PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI
                  DATABASE=0001=DSNDB01   PAGE SET=00AF=SYSUTILX

 DSN1213I DSN1LGRD LAST LOG  RBA ENCOUNTERED 0000653FE401
 DSN1214I NUMBER OF LOG RECORDS READ 0000000000009274
===============================================================================

DSN1157I RESTART SUMMARY

DSN1153I DSN1LSIT CHECKPOINT
              STARTRBA=000063249000  ENDRBA=000063254452  STARTLRSN=BB86A525B31F
ENDLRSN=BB86A525B687
               DATE=04.198  TIME=14:32:11

DSN1162I DSN1LPRT  UR  CONNID=UTILITY   CORRID=PAOLOR3L      AUTHID=PAOLOR3   PLAN=DSNUTIL
             START DATE=04.198 TIME=14:34:20  DISP=INFLIGHT            INFO=COMPLETE
             STARTRBA=000064F4FD61  STARTLRSN=BB86A5A06A6D  NID=*
              LUWID=USIBMSC.SCPDB8X.BB86A3B06A55.0001  COORDINATOR=*
              PARTICIPANTS=*
             DATA MODIFIED:
                  DATABASE=0109=PAOLODB   PAGE SET=0012=TSLINEI

DSN1160I DATABASE WRITES PENDING:
         DATABASE=0001=DSNDB01    PAGE SET=00AF=SYSUTILX   START=00006108D000
         DATABASE=0109=PAOLODB    PAGE SET=0012=TSLINEI    START=00006108D000
```

DSN1LOGP using RBASTART(yyyyyy) SUMMARY(NO) DBID(FFFF)

Example 15-37 is a sample of DSN1LOGP using RBASTART(yyyyyy) SUMMARY(NO) DBID(FFFF) that yyyyyy is the RBA of the latest complete checkpoint END on the archive log as shown in Example 15-34 on page 276.

Example 15-37 DSN1LOGP using RBASTART(yyyyyy) SUMMARY(NO) DBID(FFFF)

```
//BSDS01   EXEC PGM=DSN1LOGP
//ARCHIVE  DD   DISP=OLD,DSN=DB8XU.ARCHLOG2.D2004198.T1434413.A0000048
//SYSPRINT DD   SYSOUT=*
//SYSSUMRY DD   SYSOUT=*
//SYSIN    DD   *
 RBASTART (000063254452) SUMMARY(NO) DBID(FFFF)
```

Example 15-38 only tells you the RBA of the last valid log record as a SYSPITR. You can find "*LAST LOG RBA ENCOUNTERED 0000653FE401*" very easily in this case.

Example 15-38 DSN1LOGP to find the RBA of last valid log record easily

```
DSN1212I DSN1LGRD FIRST LOG  RBA ENCOUNTERED 000063254452
DSN1213I DSN1LGRD LAST LOG  RBA ENCOUNTERED 0000653FE401
DSN1214I NUMBER OF LOG RECORDS READ 0000000000009085
```

15.6.2 Considerations

The RESTORE SYSTEM LOGONLY eliminates complex procedures and reduces recovery skills needed for disaster recovery. It has a short outage at the primary to suspend and resume logging.

It eliminates the complexity for FlashCopy of a restore with log apply, since it supports one step recovery. This solution starts at RBLP and processes all data set restoration up to the PIT you want to recover. It is much faster than the individual "RECOVER xxx LOGONLY" and does not require image copies, unless REORG / LOAD LOG(NO) have occurred since RBLP.

But it does need duplicate disk units for FlashCopy and sufficient tape drives to dump secondaries for transport.

16

FlashCopy Consistency Group and restart

In this chapter we describe a procedure to create a consistency point of the DB2 subsystem using FlashCopy Consistency Group, a function available with FlashCopy V2. This procedure works without issuing the SET LOG SUSPEND, and it does not require DB2 V8. The Consistency Group option creates a group of volumes which will issue an Extended Long Busy (ELB) to any application which attempts a write. As a result, the FlashCopy completes with full data consistency and the application does not need to be quiesced.

When you restart the DB2 subsystem using FlashCopy Consistency Group, logs do not need to be applied.

This scenario can only be used for a normal crash and restart, and not for a log apply, since RBLP is not set automatically, unless automation is used to do a Set Log Suspend followed by a Set Log Resume to get an RBA value. We will compare it to Chapter 15, “Set Log Suspend - FlashCopy - More Log - Restore System Log Only” on page 251.

The information presented in this chapter is complemented by:

- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries,* SG24-5680-4
- *IBM TotalStorage Enterprise Storage Server Web Interface User’s Guide*, SC26-7448
- *z/OS DFSMS Advanced Copy Services*, SC35-0428

This chapter covers the following topics:

- Overview
- The procedure
- The test environment
- Backup using FlashCopy Consistency Group
- Workload at local site
- Dumping the FlashCopy
- DB2 restart at recovery site
- Considerations

16.1 Overview

The FlashCopy Consistency Group is a facility to create a multi-volume copy effectively in an instant. The copy captures all dependent writes up to a point-in-time. There is no need to quiesce applications or application write processing.

The Consistency Group Copy can be viewed as a *time-consistent copy*. The resulting group of target volumes has the same level of data currency and data consistency that the group of source volumes would have had if the host writing to the source volumes had suddenly lost power and stopped processing. Therefore, the copy can also be thought of as being *power failure consisten*t. For an example of how a FlashCopy Consistency Group works, see also 5.5, “FlashCopy Consistency Group” on page 77.

A FlashCopy Consistency Group provides a means for quicker recovery than a power-fail consistent copy, because quiescing the I/O could be disruptive if performed more than once a day. FlashCopy Consistency Group offers the following potential benefits compared to a standard FlashCopy operation with DB2 Log Suspension:

- A time-consistent copy can be made without a requirement to temporarily stop applications.
- No operator intervention is required.
- There is very little disruption to host applications.
- Copies can potentially be made more frequently.
- The amount of lost data might be reduced in the event of a site failure.

  Compared to Set Log Suspend, described in Chapter 17, “PPRC - FlashCopy from secondary” on page 301, this statement it true only if more copies can be made. For a DR scenario, the Consistency Group must be dumped to tape. For instance, if you obtain four Consistency Groups a day, but use three for local site recovery (dumping only one to tape for off-site use), then there is no improvement over the Set Log suspend methodology.

For more detailed information on FlashCopy Consistency Group, refer to Chapter 10, “FlashCopy Consistency Group” on page 139.

> **Attention:** FlashCopy consistency group can only be invoked using the ESS Copy Services Web User Interface in support of the FlashCopy V2 function.

As shown in Figure 16-1, the backup using FlashCopy Consistency Group is issued and the Extended Long Busy (ELB) condition is invoked at RBA/LRSN **A**. All volumes will be dumped using FlashCopy in only a few minutes.

The ELB condition on each source volume is involved in the consistency group creation; it holds all the I/O operations directed to the disk until it is released at RBA/LRSN **B**. This condition also holds all the dependent I/O operations on other disks not already processed by the FlashCopy function.

When all FlashCopy relationships are established and the ELB condition is reset, you should dump the copies to tape and send them to a recovery site. This is not a trivial task and can take hours.

In Figure 16-1, showing the use of FlashCopy Consistency Group, PTAM means Pickup Truck Access Method.

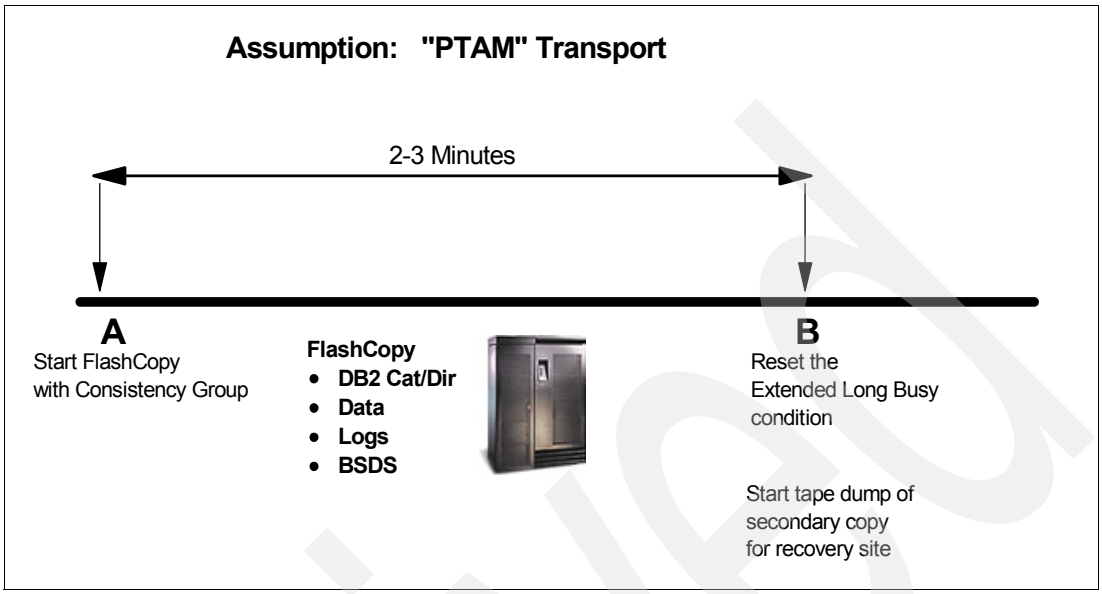


Figure 16-1 FlashCopy Consistency Group

As shown in Figure 16-2, you now restore the FlashCopy data sets to disk. This will take more than a few minutes, depending on how many volumes are copied and how many tape/disk paths are available at the recovery site.

You then just issue the START DB2 command at restart point **B**. A normal restart is performed. The goal of this scenario is that it looks exactly like a DB2 local crash. All inflight and inabort URs will back out at restart, and indoubt URs will remain until the coordinator is restarted. When restart is complete, new work can start.

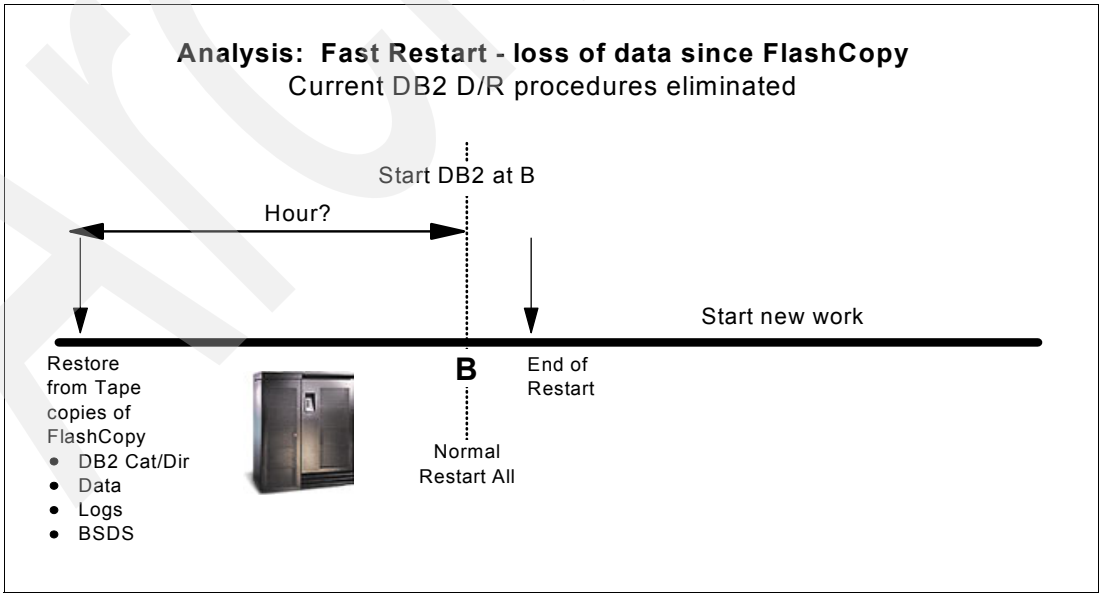


Figure 16-2 Restore FlashCopy - Restart DB2

16.2 The procedure

In this section, we provide an overview of the steps necessary to back up and restore a DB2 subsystem and data sharing group to a consistency point that the FlashCopy Consistency Group function supports at the recovery site. We will describe the steps in detail in the next sections.

16.2.1 Backup using FlashCopy Consistency Group

To use the FlashCopy Consistency Group function, you should first define tasks to establish the FlashCopy pairs, and then define tasks to resume I/O activity after the Consistency Group is created. The defined tasks can be executed from the ESS Copy Services Web User Interface or from Command Line Interface (CLI). For the detailed information, refer to 16.4, “Backup using FlashCopy Consistency Group” on page 286.

A FlashCopy Consistency Group can span multiple volumes, multiple LSSs, even multiple ESSs. To create *a Consistency Point*, do the following steps:

1. Define FlashCopy Consistency Group Creation Task:
   – This creates the Consistency Group that establishes the FlashCopy pairs.
2. Define the FlashCopy Consistency Group created task:
   – This resumes the volumes I/O activity after the Consistency Group creation.
3. Create a consistent copy by running the previously defined tasks:
   a. Run the *FlashCopy Consistency Group Creation* task, which also sets the ELB condition.
   b. Wait for all FlashCopy relationships to be established.
   c. Run the *FlashCopy Consistency Group Created* task, which resets the ELB condition.

16.2.2 Dump the FlashCopy

The FlashCopy Consistency Group works only from ESS Copy Services Web Interface and produces an exact copy of the source disk, including disk label. This means that all FlashCopy target disks are varied in offline after FlashCopy.

To dump the FlashCopy target disks in offline to tape, do the following steps:

For more information, refer to 16.6, “Dumping the FlashCopy” on page 292.

1. Execute ICKDSF REFORMAT DUMPCOND(SET) to change the label of FlashCopy target disks.
2. Vary all FlashCopy target disks online.
3. Use DFSMSdss to dump the FlashCopy to tape and then send it to the remote site.

16.2.3 Restore and restart at the recovery site

Now, you can restore DB2 system to a consistency point that the FlashCopy Consistency Group created. For more information, refer to 16.7, “DB2 restart at recovery site” on page 294.

1. Unallocate the ICF catalogs that reside on the target volumes.
2. Use DFSMSdss to restore the tape dump of FlashCopy to disk.

3. Use the access method services IMPORT command to import the any ICF catalogs that did not exist previously at the recovery site.
4. If using DDF, update LOCATION and LUNAME in BSDS using the change log inventory utility (DSNJU003). If data sharing, do it for all members.
5. Start DB2. If data sharing, delete all CF structures and restart all members.
6. Recover all objects in restricted status (GRECP in case of data sharing).

16.3 The test environment

The tests were conducted in a *non-data sharing environment*, using the following hardware and software:

- Hardware:
  - zSeries 990 processor Model B16
  - 1 Logical Partition (LPAR with 2 dedicated CPUs
  - 2 GB real memory
- Software:
  - z/OS V1R5, service level RSU0402
  - DB2 for z/OS V8
- Disks:
  - ESS Model 800 with FICON channels – FlashCopy V2

Figure 16-3 illustrates the disk of a FlashCopy Consistency Group test environment.

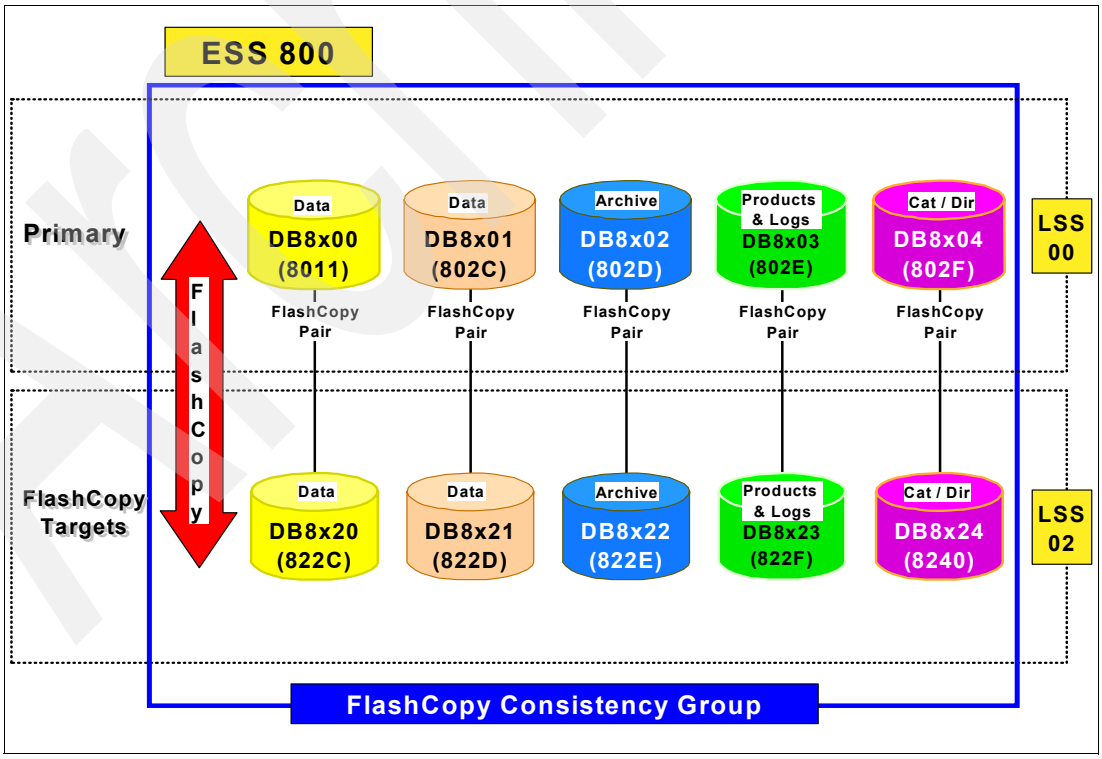


Figure 16-3 FlashCopy Consistency Group Test Environment

16.4 Backup using FlashCopy Consistency Group

In this section we discuss FlashCopy Consistency Group available with the ESS Copy Services Web User Interface in support of the FlashCopy V2 function, and present examples showing how to implement it.

The FlashCopy Consistency Group creates a consistency point for the DB2 subsystem, and it can only be invoked using the ESS Copy Services Web User Interface (WUI), which is documented in *IBM TotalStorage Enterprise Storge Server Web Interface User's Guide,* SC26-7448.

16.4.1 Define the FlashCopy Consistency Group Creation Task

In Figure 16-4 we show the pairing of volumes:

1. In the ESS Copy Services Volumes panel, select the FlashCopy source volume on the left column, *11*.
2. Then right-click to select the corresponding FlashCopy target volume on the right column, *2c*.

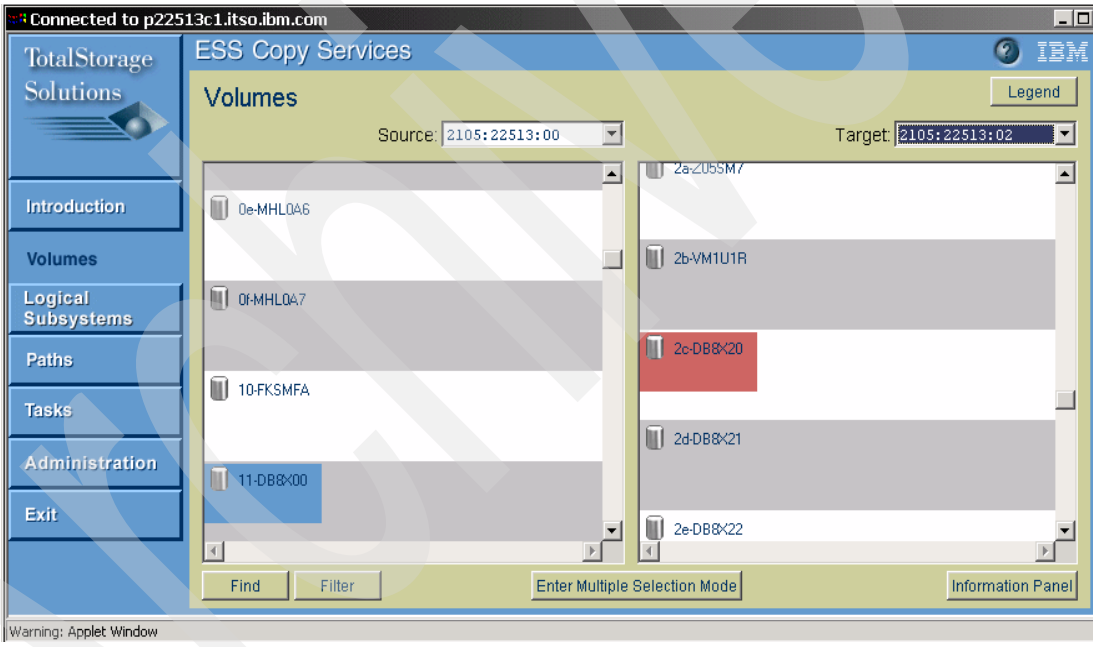


Figure 16-4 ESS Copy Services - Volumes

3. Right-click again to bring up the Select Task Type panel.

4. Select the **Establish FlashCopy pair** option as shown in Example 16-5, see and click **Next**.

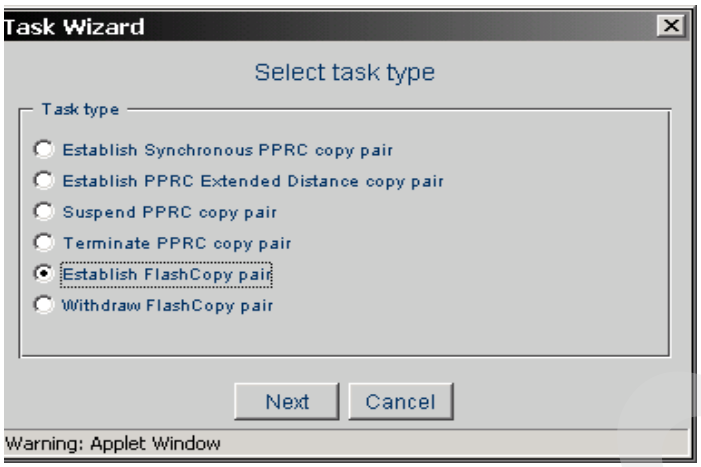


Figure 16-5 Establishing the FlashCopy pair

5. At the Select Copy Options panel, select the **Freeze FlashCopy Consistency Group** option as shown in Figure 16-6 and click **Next**.

Figure 16-6 Specifying FlashCopy Consistency Group

6. Enter the name and description of the task in the Define Task panel (see Figure 16-7), then select **Save** to save the defined task.

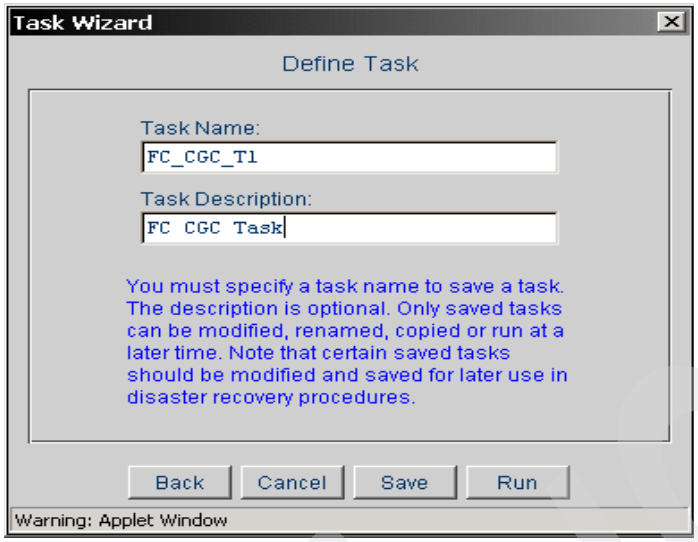


Figure 16-7 Task definition

7. Repeat steps 1 on page 286 through step 6 on page 288 for each source volume that will be part of the Consistency Group.
8. When the establish tasks have been created for all volumes in the Consistency Group, click **Tasks** in the navigation frame to go the ESS Copy Services Tasks panel.
9. Click to highlight the task defined in step 6.
10. Press the Ctrl key and click to highlight each task defined in step 7 in Figure 16-8.

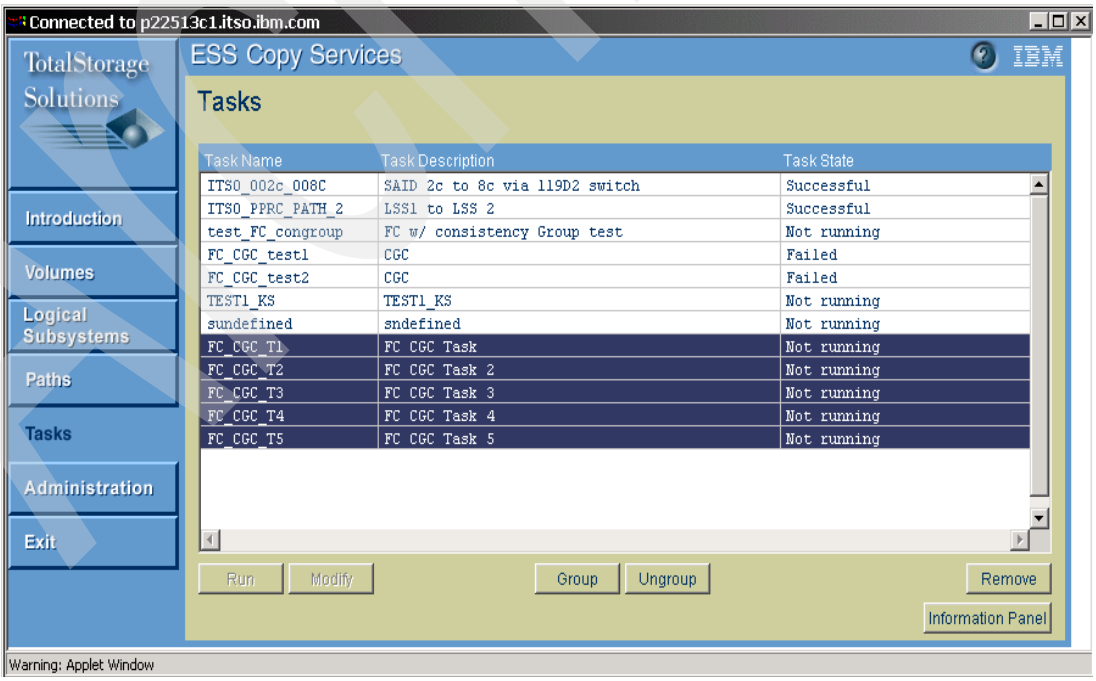


Figure 16-8 List of tasks

When all required tasks have been highlighted, click **Group** to define a composite task that comprises all highlighted tasks. We will refer to this task as the Freeze_CG_task. See Figure 16-9.

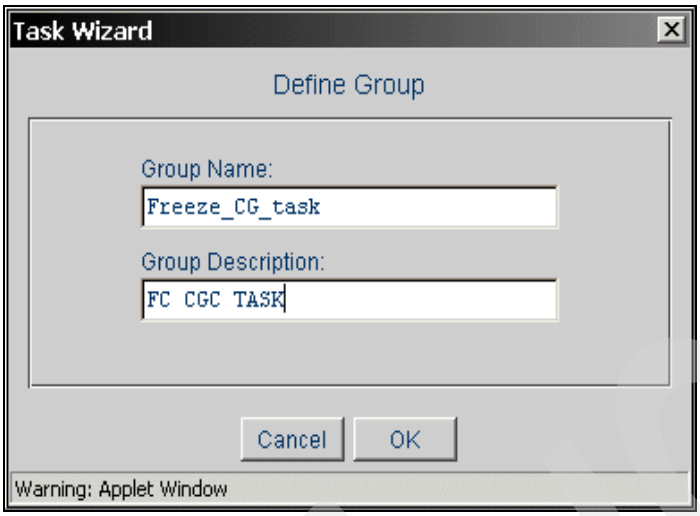


Figure 16-9 Tasks grouping

Tip: In the previous procedures, if you are going to select several volume pairs, you have the alternative of initially clicking the **Enter Multiple Selection Mode** button and then iterate through step 1 on page 286 and step 2 on page 286 until you select all the volumes. Once all the volumes have been selected, you then proceed with steps 3 on page 286 through step 6 on page 288, with the procedure ending at step 6 on page 288.

16.4.2 Define the FlashCopy Consistency Group Created Task

Next we show you how to do this operation:

1. From the ESS Copy Services Logical Subsystems (LSS) panel (see Figure 16-10), click to select an LSS which contains source volumes from the FlashCopy Consistency Group defined.
2. Right-click the source LSS (do not select target LSS).

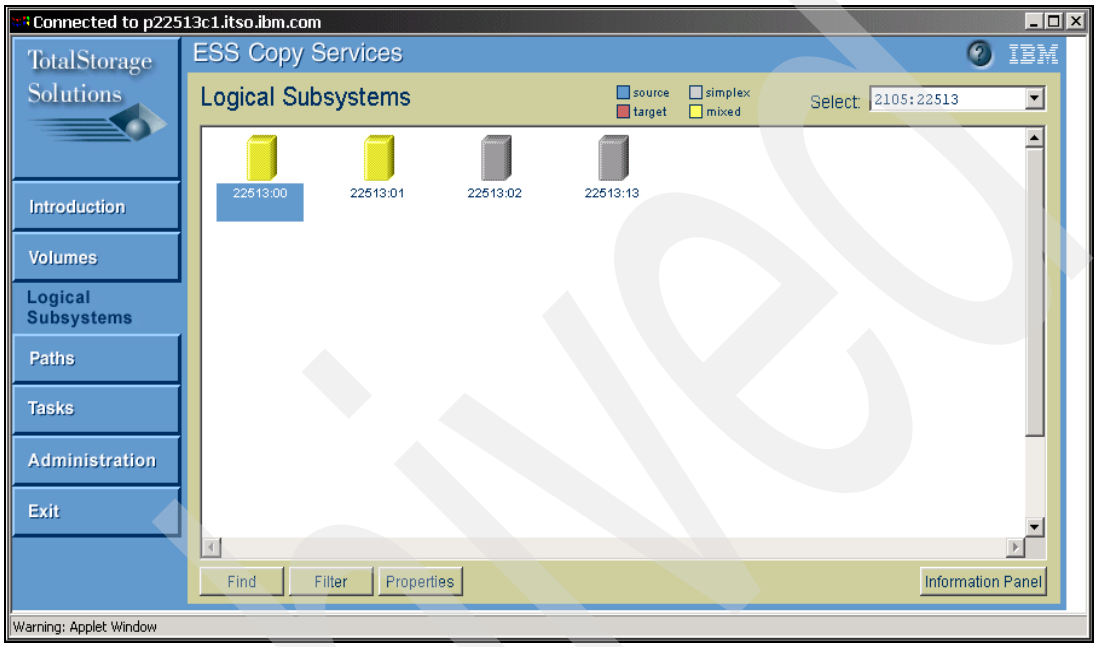


Figure 16-10 LSS panel

3. Right-click again to bring up the Select Task Type panel.
4. Select **Consistency Created** (as shown in Figure 16-11) and click **Next**.

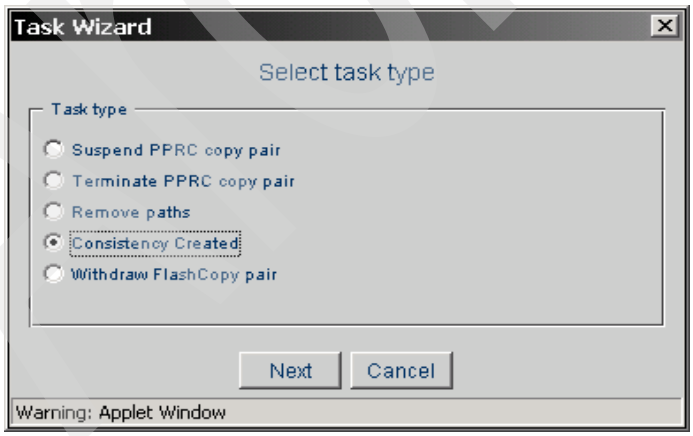


Figure 16-11 Defining Consistency Created task

From the Select Copy Options panel, select **FlashCopy Consistency Group (One LSS selected)** as shown in Figure 16-12, and click **Next**.

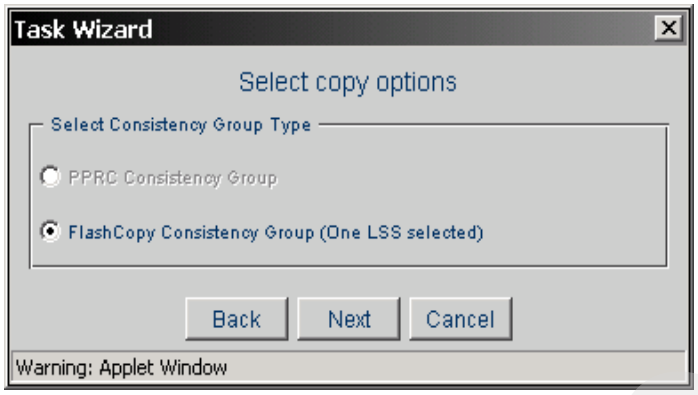


Figure 16-12 Select Consistency Group Type

5. Enter the name and description of the task at the Define Task panel, and select **Save** to save the defined task (In our test, we used one LSS for source volumes in the Consistency Group. So we skipped step 7 through 11. We refer to this task as the CG_Created_task).
6. Repeat steps 1 on page 290 through step 6 for each LSS that contains source volumes in the Consistency Group.
7. When tasks have been created for all volumes in the Consistency Group, click **Tasks** in the navigation frame to go the ESS Copy Services Tasks panel.
8. Click to highlight the task defined in step 6.
9. Press the Ctrl key and click to highlight each task defined in step 7.
10. When all required tasks have been highlighted, click **Group** to define a composite task that comprises all highlighted tasks. We will refer to this task as the CG_Created_task_G.

16.4.3 Create consistent copy

To create a consistent copy of all volumes in the Consistency Group, do the following steps:

1. Run the Freeze_CG_task. The freeze option in the defined task, see Figure 16-13, causes the volumes to remain in a long-busy condition until the condition is reset (see the step 2) or the time-out value expires (default is two minutes).

Notes:

- After you perform a FlashCopy with the freeze option, write activity will go to zero.

  The ELB status is confirmed by IEA494I messages, and a display DA OJOB from SDSF will show the jobs' I/O going to zero. See 17.3.4, "FlashCopy with Consistency Group" on page 312, for examples.
- If you click on the source volume you selected and then click the **Information Panel,** it will display that the volume is **Extended Long Busy for FlashCopy Consistency Groups**.

2. Wait for all FlashCopy relationships to be established.
3. Run the CG_Created task. *This task resets the long-busy condition*, making the volumes available for normal I/O activity.

There is no message informing you that the ELB status is reset; you will see it from the I/O activity being resumed (Figure 16-13).

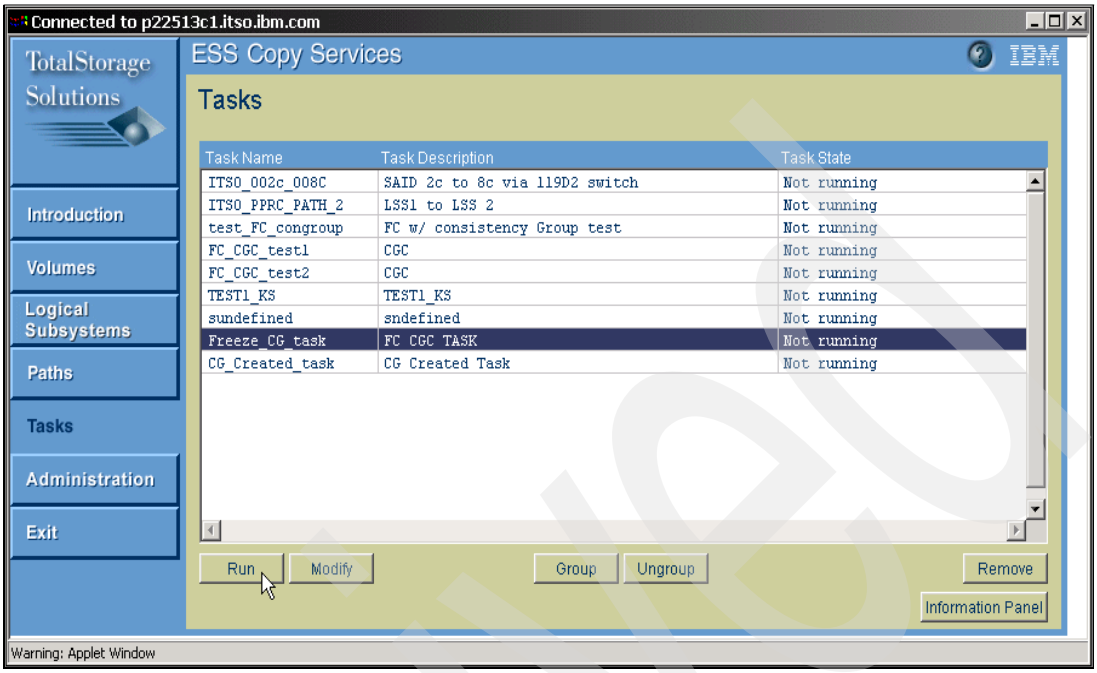


Figure 16-13 Run the Freeze option task

16.5 Workload at local site

Example 16-1 lists the workload active during FlashCopy at local site. When you perform a FlashCopy with Consistency Group (the freeze option), they all go to zero write activity. They will be available for normal I/O activity by resetting the long busy condition.

Example 16-1 -DIS THD(*)

```
NAME     ST A   REQ ID          AUTHID   PLAN     ASID TOKEN
UTILITY  T  *   341 PAOLOR33    PAOLOR3  DSNUTIL  0029  1392
UTILITY  T        7 PAOLOR33    PAOLOR3  DSNUTIL  0029  1393
UTILITY  T       42 PAOLOR31    PAOLOR3  DSNUTIL  0027  1387
UTILITY  T  *     7 PAOLOR31    PAOLOR3  DSNUTIL  0027  1394
UTILITY  T  *     7 PAOLOR31    PAOLOR3  DSNUTIL  0027  1395
UTILITY  T  *     7 PAOLOR31    PAOLOR3  DSNUTIL  0027  1397
TSO      T  *     3 PAOLOR3     PAOLOR3           0063  1400
TSO      T       16 PAOLOR2     PAOLOR2  DSNESPCS 0074  1342
TSO      T  *     3 PAOLOR2     PAOLOR2           0074  1401
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I  -DB8X DSNVDT '-DIS THD' NORMAL COMPLETION
```

16.6 Dumping the FlashCopy

One thing to remember is that FlashCopy Consistency Group works only from the ESS Copy Services Web Interface and produces an exact copy of the source disk, including disk label. This means that it will not be possible to bring the target volumes online to the same LPAR in order to dump them on tape.

To do this, we must first change the label of the FlashCopy target disk to be different from the source one, using ICKDSF REFORMAT command. And then you can dump it to tape as described in the following sections.

ICKDSF using REFORMAT DUMPCOND(SET)

Example 16-2 is issued to change the label of FlashCopy target disk to DB8X20 using DUMPCOND(SET). It will allow the dump of the target volume DB8X20, since it is varied online.

Example 16-2 ICKDSF using REFORMAT DUMPCOND(SET)

```
//*****************************************************************
//**   VERIFY : ORIGINAL VOLUME SERIAL                            **
//**   VOLID  : TARGET   VOLUME SERIAL                            **
//*****************************************************************
//CLIPVOL  EXEC PGM=ICKDSF,REGION=4096K,PARM=NOREPLYU
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
     REFORMAT UNIT(822C) VERIFY(DB8X00) VOLID(DB8X20) DUMPCOND(SET)
/*
```

Vary the FlashCopy target disks online

You should online all FlashCopy target disks to allow the dump to tape as shown in Example 16-3.

Example 16-3 Vary all FlashCopy target disks online

```
V 822C,ONLINE
V 822D,ONLINE
V 822E,ONLINE
V 822F,ONLINE
V 8240,ONLINE
```

Since the DUMPCOND(SET) is specified, it changes the volume label DB8X00 to DB8X20, but remains both the VTOC index and the VVDS to match the source volume DB8X00, as shown in Example 16-4. All data is inaccessible at that time.

Example 16-4 Sample output of DUMPCONDITIONING

```
DSLIST - Data Sets on volume DB8X20                              Row 76 of 82
Command ===>                                                 Scroll ===> CSR

Command - Enter "/" to select action                 Message          Volume
-------------------------------------------------------------------------------
         DB8XD.DSNDBD.DSN8D81P.XDSPTXT1.J0001.A001                    DB8X20
         DB8XD.DSNDBD.DSN8D81P.XMAPRTBL.I0001.A001                    DB8X20
         DB8XD.DSNDBD.DSN8D81P.XOPTVAL1.J0001.A001                    DB8X20
         SYS1.VTOCIX.DB8X00                                           DB8X20
         SYS1.VVDS.VDB8X00                                            DB8X20
         UCAT.DB8XD                                                   DB8X20
         UCAT.DB8XD.CATINDEX                                          DB8X20
***************************** End of Data Set list ****************************
```

Use DFSMSdss to dump the FlashCopy to tape

For disaster recovery, you should dump to tape the copy to send it to the recovery site, as shown in Example 16-5.

Example 16-5 DFSMSdss DUMP

```
//VDB8X00 EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X20
//OUTDS    DD  UNIT=3490,DISP=(,PASS),LABEL=4,
//             VOL=(,RETAIN,SER=(PRB000,PRB001,PRB002,PRB003)),
//             DSN=BACKUP.DB8X00
//SYSIN    DD  *
  DUMP INDDNAME(INVOL) OUTDDNAME(OUTDS) -
       FULL ALLDATA(*) ALLEXCP -
       CANCELERROR OPT(4) ADMIN
```

Because we use the REFORMAT DUMPCOND(SET) as shown in Example 16-2 on page 293, the DUMP FULL of the target device modifies the tape records so that it appears as if the tape was created directly from the source volume, DB8X00. We do not have to clip the volume serial of the target volume DB8X20 following a subsequent restore.

16.7 DB2 restart at recovery site

When restarting the DB2 system, the current status rebuild phase reads all records from the most recent system checkpoint to the end of log and determines the status of all URs and page sets.

DB2 can recover any UR in status incommit, inflight, inabort, and postponed abort without asking the transaction coordinator (CICS or IMS) during forward and backward log recovery. The URs in indoubt status are automatically resolved by asking the coordinator as soon as CICS and/or IMS reconnect to DB2. You should analyze the CICS/IMS/DB2 logs to determine the correct action for unresolved URs in case automatic resolution fails. The manual resolution of indoubt URs is required only in this case.

16.7.1 Restore the backup of FlashCopy

Before restoring the dump, we must unallocate the ICF catalogs that reside on the volumes where the dump will be restored. We use the MODIFY CATALOG command as shown in Example 16-6.

Example 16-6 MODIFY CATALOG command to unallocate ICF catalogs

```
/F CATALOG,UNALLOCATE(UCAT.DB8XL)
/F CATALOG,UNALLOCATE(UCAT.DB8XD)
```

Attention: The assumption is that the volumes at the recovery site have been initialized with the same volume serial numbers (VOLID) as the ones used at the local site. If not, you should specify the COPYVOLID parameter.

We used the same volume serial numbers at recovery site. In this case, the RESTORE command can be used without the COPYVOLID parameter, as the VOLID is the same on disk as the one on the tape. Example 16-7 shows sample JCL statements to restore a tape dump of a target volume without the COPYVOLID parameter.

Example 16-7 DFSMSdss RESTORE

```
//VDB8X00  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X00
//INDS     DD  UNIT=3490,DISP=OLD,LABEL=4,
//             VOL=(,RETAIN,SER=(PRB000,PRB001,PRB002,PRB003)),
//             SER=(PRB000,PRB001,PRB002,PRB003)),
//             DSN=BACKUP.DB8X00
//SYSIN    DD  *
  RESTORE INDDNAME(INDS) OUTDDNAME(OUTVOL) -
          FULL PURGE -
          CANCELERROR ADMIN
```

The PURGE parameter is required in the restore job because the VVDS name on the target volume does not match the target volume serial number (as a result of the DUMPCONDITIONING parameter that was specified for the full volume COPY).

16.7.2 DB2 restart

When restarting DB2 system, there are an In-commit UR and an Inflight UR on the current status rebuild phase because of active threads frozen during the FlashCopy, and they are automatically resolved on forward and backward log recovery phase, as shown in Example 16-8.

Example 16-8 -DB8X STA DB2

```
DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE
NAME DSNZPARM
S DB8XIRLM
DSNY001I  -DB8X SUBSYSTEM STARTING
IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  783
IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  784
IEC161I DB8XU.BSDS01.INDEX,UCAT.DB8XL
IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  785
IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  786
IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  787
IEC161I DB8XU.BSDS02.INDEX,UCAT.DB8XL
IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  788
IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.190 17:28:43.41
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  790
DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS02,
STARTRBA=00000EC40000,ENDRBA=000010DFFFFF
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  791
DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS02,
STARTRBA=00000EC40000,ENDRBA=000010DFFFFF
DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  792
STARTRBA=000010BED000
S DB8XDBM1
S DB8XDIST
DSNR001I  -DB8X RESTART INITIATED
DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=00000EC40E2E
DSNR004I  -DB8X RESTART...UR STATUS COUNTS  805
IN COMMIT=1, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
DSNR007I  -DB8X RESTART...STATUS TABLE  806
```

```
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
B TSO      PAOLOR2      PAOLOR2  DSNESPCS F 0000104D51B1 190 17:27:55
B UTILITY  PAOLOR33     PAOLOR3  DSNUTIL  C 000010BEAAB7 190 17:28:51
DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  808
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR018I  -DB8X RESTART...BACKWARD RECOVERY PROCESSED  809
 FROM RBA 0000104D5348 TO RBA 0000104D51B1
DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  810
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR002I  -DB8X RESTART COMPLETED
-DB8XRECOVER POSTPONED
DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
DSNL003I  -DB8X DDF IS STARTING
DSNL519I  -DB8X DSNLILNR TCP/IP SERVICES AVAILABLE  816
           FOR DOMAIN wtsc64.itso.ibm.com AND PORT 38080
DSNL004I  -DB8X DDF START COMPLETE  817
           LOCATION  DB8X
           LU        USIBMSC.SCPDB8X
           GENERICLU -NONE
           DOMAIN    wtsc64.itso.ibm.com
           TCPPORT   38080
           RESPORT   38081
DSN9022I  -DB8X DSNYASCP 'STA DB2' NORMAL COMPLETION
DSNL519I  -DB8X DSNLIRSY TCP/IP SERVICES AVAILABLE  819
           FOR DOMAIN wtsc64.itso.ibm.com AND PORT 38081
```

16.7.3 Recover all objects in the restricted status

The following steps are recommended to recover all objects in the restricted status:

- Execute the display command to check for active utilities or restricted objects.
- Terminate any active utilities.
- Recover all objects in the restricted status.
- Validate that all restricted objects are recovered successfully.

Check for active utilities or restricted objects

Attention: The active utilities and restricted status may vary with the active workload suspended while executing FlashCopy. The job followed also can vary, and it depends on the customer's environment.

Example 16-9 is a output of the LIST UTIL(*) command to check for active utilities.

Example 16-9 -DIS UTIL(*)

```
DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = DSNTEX
                PROCESSING UTILITY STATEMENT 1
                UTILITY = REORG
                PHASE = SORTBLD   COUNT = 0
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = STOPPED
```

```
 DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                 MEMBER =
                 UTILID = DSNTEY
                 PROCESSING UTILITY STATEMENT 6
                 UTILITY = LOAD
                 PHASE = BUILD   COUNT = 0
                 NUMBER OF OBJECTS IN LIST = 1
                 LAST OBJECT STARTED = 1
                 STATUS = STOPPED
 DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

Example 16-10 shows the restricted status of all databases after DB2 restart at the recovery site. The restricted status may vary depending on which utility was processing and on which phase the utility was in during FlashCopy.

Example 16-10 -DIS DB(*) SP(*) RESTRICT LIMIT(*)

```
DSNT360I  -DB8X ************************************
 DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                 *    RESTRICTED
 DSNT360I  -DB8X ************************************
 DSNT362I  -DB8X     DATABASE = DRTESTD2  STATUS = RW
                    DBD LENGTH = 24218
 DSNT397I  -DB8X
 NAME     TYPE PART  STATUS           PHYERRLO PHYERRHI CATALOG  PIECE
 -------- ---- ----- ----------------- -------- -------- -------- -----
 DSN8S81D TS         RW,CHKP
 DSN8S81P TS         RW,CHKP,UTUT
 XEMP17ZR IX         RW,RBDP,UTUT
 XEMPPROJ IX         RW,UTUT
 ******* DISPLAY OF DATABASE DRTESTD2 ENDED      **********************
 DSNT360I  -DB8X ************************************
 DSNT362I  -DB8X     DATABASE = DRTESTDB  STATUS = RW
                    DBD LENGTH = 24218
 DSNT397I  -DB8X
 NAME     TYPE PART  STATUS           PHYERRLO PHYERRHI CATALOG  PIECE
 -------- ---- ----- ----------------- -------- -------- -------- -----
 DSN8S81D TS         RW,UTUT
 XDEPT1   IX         RW,RBDP,UTUT
 XDEPT2   IX         RW,RBDP,UTUT
 XDEPT3   IX         RW,RBDP,UTUT
 ******* DISPLAY OF DATABASE DRTESTDB ENDED      **********************
 DSN9022I  -DB8X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

Terminate active utilities

Example 16-11 terminates active utilities instead of restarting them, because we don't usually have all work data sets at the recovery site.

Example 16-11 `-TERM UTIL(DSNTE*)`

```
DSNU166I  -DB8X DSNUGTER - LOAD UTILITY,
                UTILID = DSNTEX NOT EXECUTING,
                CLEANUP COMPLETE
DSNU166I  -DB8X DSNUGTER - LOAD UTILITY,
                UTILID = DSNTEY NOT EXECUTING,
                CLEANUP COMPLETE
DSN9022I  -DB8X DSNUGCCC '-TERM UTIL' NORMAL COMPLETION
```

Recover the restricted objects

Example 16-12 is a output of the restricted status changed after terminating utilities as shown in Example 16-11. You should recover the remaining restricted objects.

Example 16-12 -DIS DATABASE(*) SPACE(*) RESTRICT LIMIT(*)

```
DSNT360I  -DB8X ***********************************
 DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                 *    RESTRICTED
 DSNT360I  -DB8X ***********************************
 DSNT362I  -DB8X     DATABASE = DRTESTD2  STATUS = RW
                    DBD LENGTH = 24218
 DSNT397I  -DB8X
 NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
 -------- ---- ----- ----------------- -------- -------- -------- -----
 DSN8S81D TS         RW,CHKP
 DSN8S81P TS         RW,CHKP
 XEMP17ZR IX         RW,RBDP
 ******* DISPLAY OF DATABASE DRTESTD2 ENDED      **********************
 DSNT360I  -DB8X ***********************************
 DSNT362I  -DB8X     DATABASE = DRTESTDB  STATUS = RW
                    DBD LENGTH = 24218
 DSNT397I  -DB8X
 NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
 -------- ---- ----- ----------------- -------- -------- -------- -----
 XDEPT1   IX         RW,RBDP
 XDEPT2   IX         RW,RBDP
 XDEPT3   IX         RW,RBDP
******* DISPLAY OF DATABASE DRTESTDB ENDED      **********************
```

Recover RBDP status

Example 16-13 performs REBUILD INDEX to recover RBDP of indexes as shown in Example 16-12, and it reconstructs indexes from the table that they reference.

Example 16-13 REBUILD INDEX

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEZ'
//SYSIN    DD  *
REBUILD INDEX (DRTEST2.XEMPPROJACT2)
REBUILD INDEX (DRTEST.XDEPT1,DRTEST.XDEPT2,DRTEST.XDEPT3)
```

Recover CHKP status

The LOAD utility places table spaces in CHKP proactively when there are any RI relationships. When LOAD is interrupted, as has occurred during our test, CHKP remains, whether or not there is any actual condition. At the recovery site, this presents us with a dilemma, as we want to access this table space as quickly as possible. We might be inclined to accept risk in order to reach our availability objective.

Prior to the disaster, you should decide how you will resolve the CHKP conditions, as shown in Example 16-14 and Example 16-15. There are two choices: CHECK DATA and REPAIR with SET NOCHECKPEND:

- CHECK DATA will fully resolve the condition, but for large objects, this may take a significant amount of time to resolve for large table spaces with many RI relationships.
- REPAIR with SET NOCHECKPEND is very quick, but does not resolve the underlying situation. However, in the interests of making data available as soon as possible, you may choose this option.

Next we illustrate both ways to resolve this situation.

The CHECK DATA checks table spaces for violations of referential and table check constraints, and reports information about violations that are detected. Example 16-14 is a sample JCL that we use.

Example 16-14 CHECK DATA

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEY'
//SYSUT1   DD  UNIT=VIO,SPACE=(4000,(20,20),,,ROUND)
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTOUT  DD  UNIT=VIO,SPACE=(4000,(20,20),,,ROUND)
//DSNTRACE DD  SYSOUT=*
//SYSERR   DD  UNIT=VIO,SPACE=(4000,(20,20),,,ROUND)
//SYSIN    DD  *

  CHECK DATA TABLESPACE DRTESTD2.DSN8S81D
             TABLESPACE DRTESTD2.DSN8S81E
             TABLESPACE DRTESTD2.DSN8S81P
       SCOPE ALL
       DELETE YES
       FOR EXCEPTION IN DRTEST2.DEPT       USE DRTEST2.EDEPT
                     IN DRTEST2.EMP        USE DRTEST2.EEMP
                     IN DRTEST2.PROJ       USE DRTEST2.EPROJ
                     IN DRTEST2.ACT        USE DRTEST2.EACT
                     IN DRTEST2.PROJACT    USE DRTEST2.EPROJACT
                     IN DRTEST2.EMPPROJACT USE DRTEST2.EEPA
     EXCEPTIONS 50
     SORTDEVT SYSDA SORTNUM 4
```

Example 16-15 lists REPAIR with SET NOCHECKPEND, which is intended as a means of replacing invalid data with valid data.

Example 16-15 REPAIR SET NOCHECKPEND

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEB'
 //SYSIN    DD  *
 REPAIR SET TABLESPACE DRTESTD2.DSN8S81D NOCHECKPEND
        SET TABLESPACE DRTESTD2.DSN8S81P NOCHECKPEND
```

Validate that all restricted objects are recovered successfully

All restricted objects are recovered successfully and there is no restriction for all databases, as shown in Example 16-16. You can now query and verify your table data.

Example 16-16 -DIS DB(*) SPACE(*) RESTRICT LIMIT(*)

```
DSNT365I  -DB8X    NO DATABASES FOUND
DSN9022I  -DB8X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

16.8 Considerations

This solution is easy to implement for DB2 staff (storage staff establishes) and eliminates complex procedures and need for deep recovery skills for disaster recovery. There is no infrastructure needed at the recovery site.

But it needs duplicate DASD for FlashCopy and sufficient tape drive to dump secondaries for transport. It has short outage during FlashCopy Consistency Group Creation.

Advantages

Time Consistent copy can be obtained without stopping applications. It can be obtained without suspending and resuming DB2 logging. As a result, it may be possible to make more frequent copies.

Disadvantages

If CG Created fails, no notification is given — suspended writes resume. Cannot be used to apply additional log using Restore System Logonly (RBLP is not updated) and 32 KB page writes can be split (if not a 32 KB CI Size).

Issues

Set Log Suspend provides a cleaner point by flushing buffers, updating BSDS with high written RBA and RBLP (that provides options for applying more log if you have it available off-site.)

17

PPRC - FlashCopy from secondary

In this chapter we describe an extension to the procedures for remote site recovery using Peer-to-Peer Remote Copy (PPRC) of the ESS storage system. We assume that additional copies of the secondary set of DASD are needed, and explore different ways of procuring this consistent additional copy. A likely reason would be to provide a backup copy before a restart is attempted; another reason is to provide user managed versions for protection against data corruption.

The primary objective of our tests is explore ways to guarantee that a consistent set of DASD is provided, even with a large number of volumes to copy.

When planning to use PPRC and restart DB2, the following publications should be referenced to complement the information presented in this chapter:

- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries,* SG24-5680-4
- *IBM TotalStorage Enterprise Storage Server Web Interface User's Guide*, SC26-7448
- *z/OS DFSMS Advanced Copy Services,* SC35-0428

This chapter covers the following topics:

- Overview of the test scenarios
- Configuring the PPRC environment
- Scenario 1: Using FlashCopy Consistency Group
- Scenario 2: Using DB2 Set Log Suspend
- Scenario 3: Using PPRC Consistency Group
- Considerations

17.1 Overview of the test scenarios

In this test scenario, we copy full volumes of an entire DB2 subsystem, including DB2 data, catalog, and logs, to the remote site volumes using PPRC. We then also make an additional full copy from the remote site volumes to a third set of volumes using FlashCopy to be used in case of failure of the remote site restart. Finally, we restart DB2 from the third (FlashCopy target) volumes.

We will explore three flavors of this scenario, the difference of each test being the method used to create the Consistency Group of the recovery data:

1. FlashCopy Consistency Group
2. DB2 Set Log Suspend and FlashCopy
3. PPRC Consistency Group

The assumption is that a third set of ESS volumes are available for FlashCopy, in addition to the PPRC source (primary) volumes and the target (secondary) volumes. Depending on your environment, it is possible that you cannot afford to allocate the third set of volumes, or that you need to bring tape dump images to the third site because of the distance limitations of PPRC synchronous copy (max 103 km). In this case you might choose test scenario B or C and use DFSMSdss to dump PPRC target volumes to tape instead of using FlashCopy.

Figure 17-1 provides a pictorial overview of the test procedures.

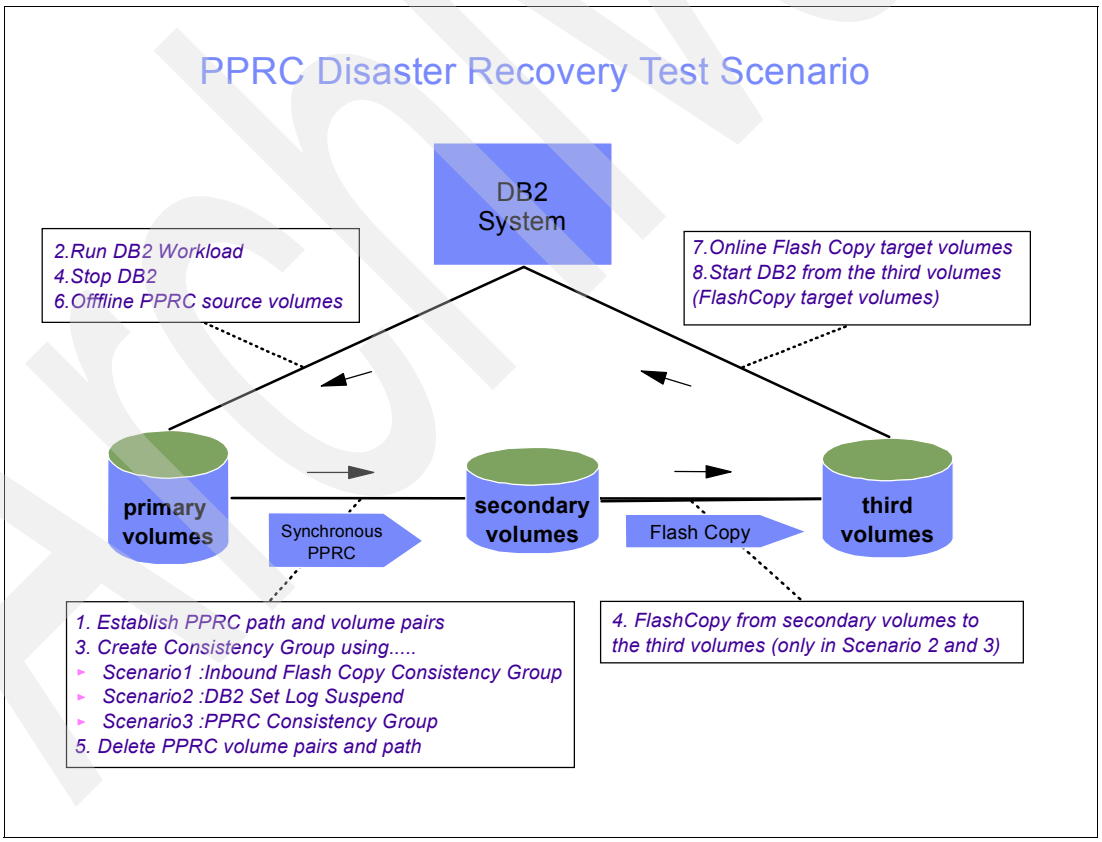


Figure 17-1 Test scenario overview

17.1.1 Testing environment

Tests were conducted in a *non-data sharing environment*, using the following hardware and software:

- Hardware:
  - zSeries 990 processor Model B16
  - 1 Logical Partition (LPAR) with 2 dedicated CPUs
  - 2 GB real memory
- Disks:
  - ESS Model 800 with FICON channels
  - FlashCopy V2
  - PPRC V2
- Software:
  - z/OS V1R5, service level RSU0402
  - DB2 for z/OS V8

Figure 17-2 shows our PPRC configuration.

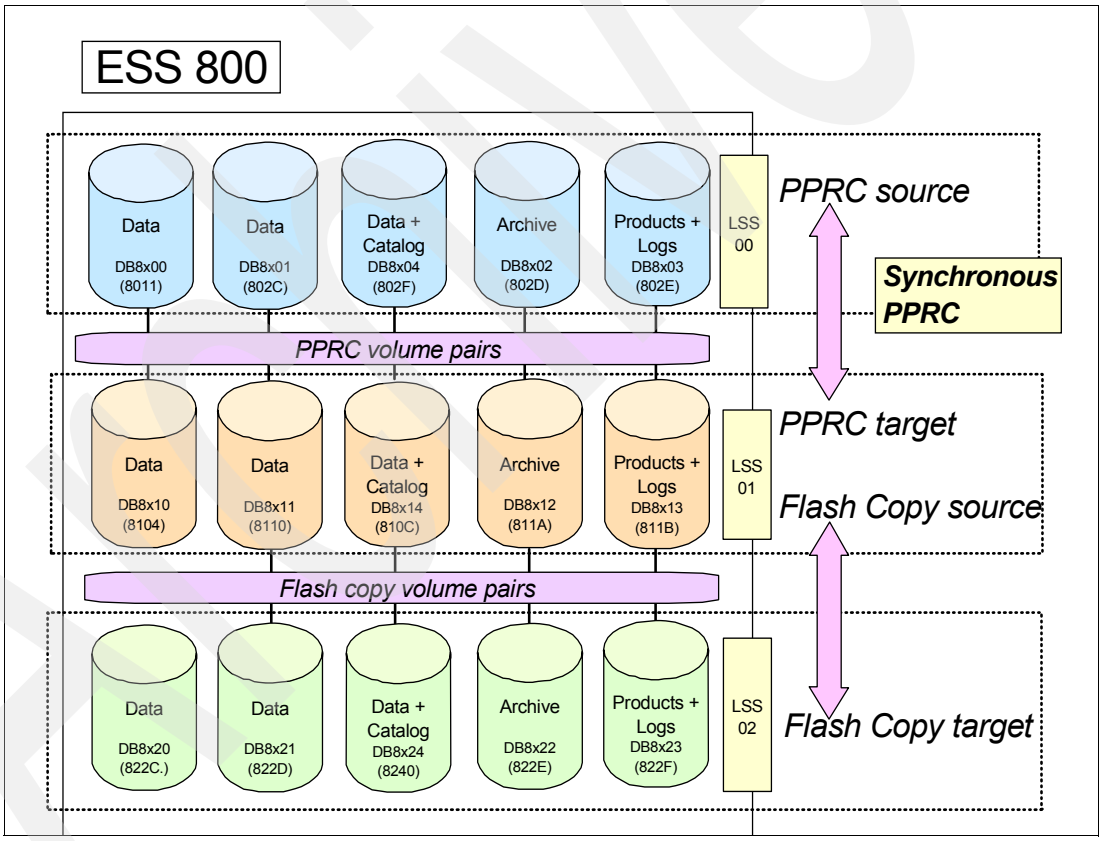


Figure 17-2 PPRC test environment

17.2 Configuring the PPRC environment

In this section we show the steps to configure our PPRC environment.

17.2.1 Tasks for PPRC testing

The ESS related tasks for PPRC testing are summarized in Table 17-1.

In our scenario we used only the ESS Web Interface for PPRC operations. If you use the TSO interface for such commands, refer to *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries,* SG24-5680-4. For a detailed description of the TSO commands refer to *z/OS DFSMS Advanced Copy Services,* SC35-0428.

Table 17-1 Tasks for PPRC testing

No.	Task Name	Description
1	PP_patng_00_01	Establish path lss00 to lss01 without consistency group
2	PP_patcg_00_01	Establish path lss00 to lss01 with consistency group
3	PP_pairs_critN	Establish pairs and critical volume=no
4	PP_rmpath_00_01	Remove path lss00 to lss01 without force option
5	PP_rmpatfo_00_01	Remove path lss00 to lss01 with force option
6	PP_stop_pairs_p	Delete pairs in primary (source) LSS
7	PP_stop_pairs_t	Delete pairs in target LSS
8	PP_susp_pairs	Suspend pairs in target LSS
9	PP_CG_created	PPRC consistency group created
10	PP_freeze	Freeze PPRC session lss00 to lss01
11	PP_pairs_resync	Establish pairs and Resync ‘copy out-of sync cyl only’
12	FC_01_02_ib_CG	FlashCopy inbound from PPRC source with CG
13	FC_01_02_ib_NC	FlashCopy inbound from PPRC source non-CG
14	FC_01_02_NC	FlashCopy secondary to third non-CG
15	FC_CG_created	FlashCopy consistency group created

17.2.2 Define task to Establish PPRC Path

1. In the ESS Copy Services of ESS Web Interface, Open the **Path** panel. Click the PPRC source LSS and target LSS, and source SAID, and right-click another source SAID, and right-click again as shown in Figure 17-3.

   In the test environment we created two paths for PPRC communication so that in case one switch port is down, PPRC is still available; it contributes to availability and performance if each Logical Subsystem is located in a different ESS.

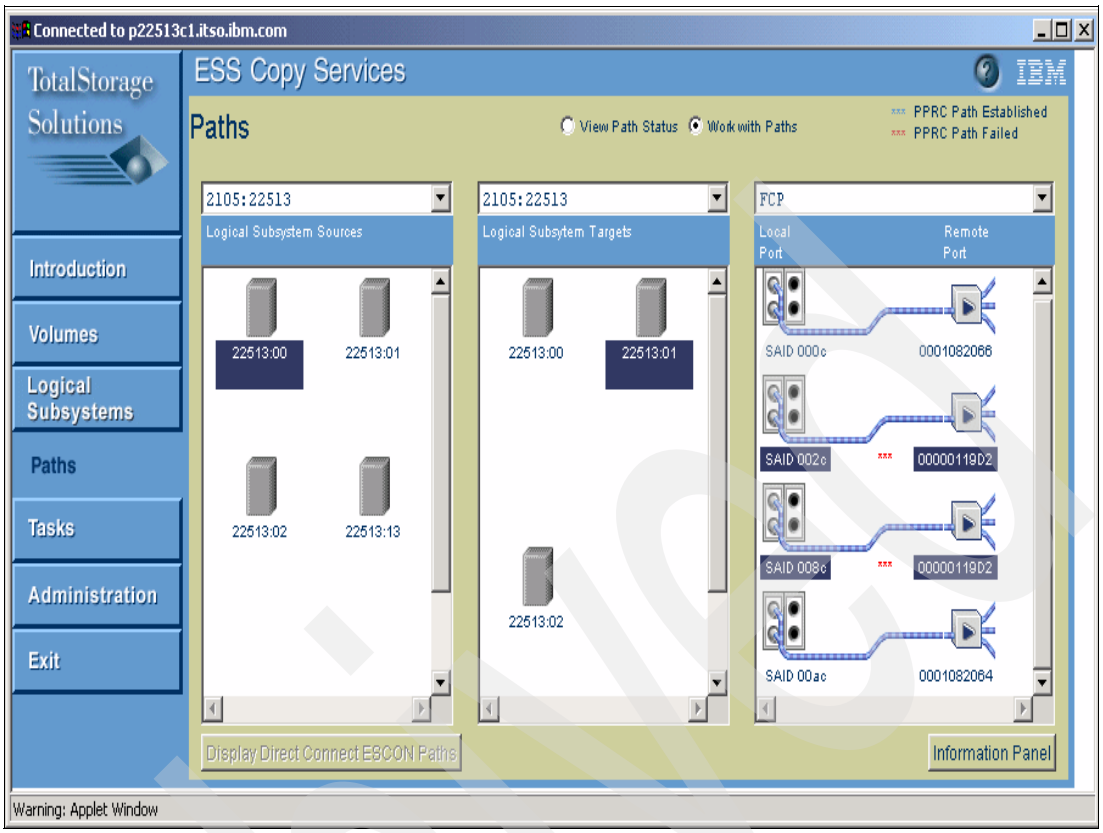


Figure 17-3 Select path

2. Select **Establish paths** and **Next** as shown in Figure 17-4.

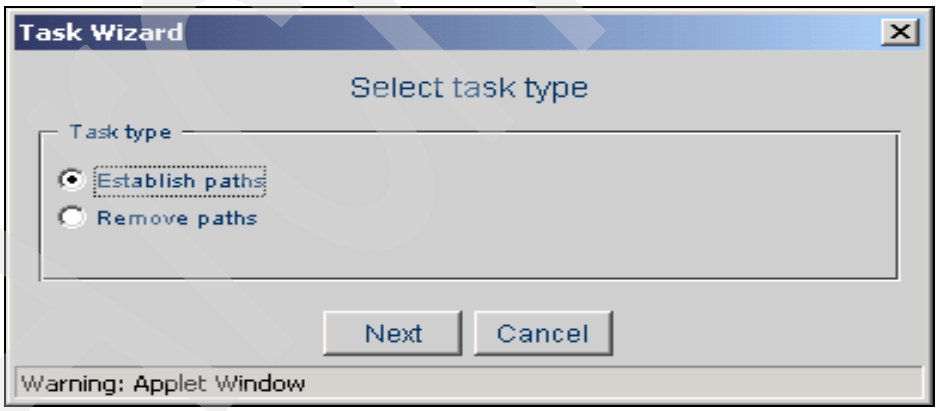


Figure 17-4 Select tasks

3. Select the SAID connections as shown in Figure 17-5.

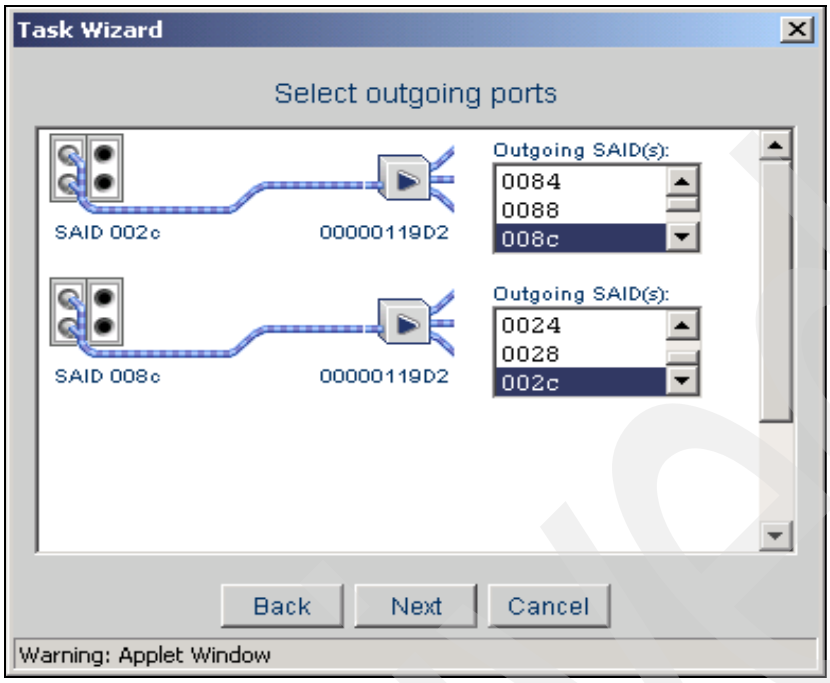


Figure 17-5 Select outgoing path

4. Specify whether you use PPRC Consistency group. See Figure 17-6.

   PPRC consistency group is a set of PPRC volume pairs that have the same primary and secondary LSS, and consistency group is created when a path is established between primary and secondary PPRC.

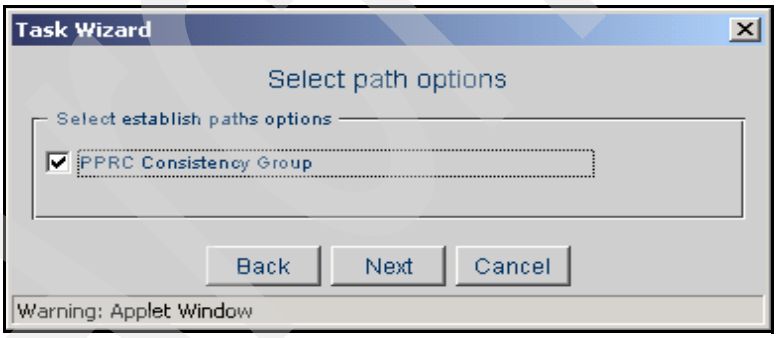


Figure 17-6 Select path options

   – When PPRC consistency group is activated:

     If an error occurs which affects any of these volume, ESS causes the error volume to enter suspended mode and additionally to enter a long busy state, so that the application cannot update the volume.

   – Without PPRC consistency group:

     If an error occurs which affects any of these volume, ESS causes the error volume to enter suspended mode, but still allows updates to the volume.

5. Specify an arbitrary **Task Name** and **Description** and click **Save**. See the example shown in Figure 17-7.

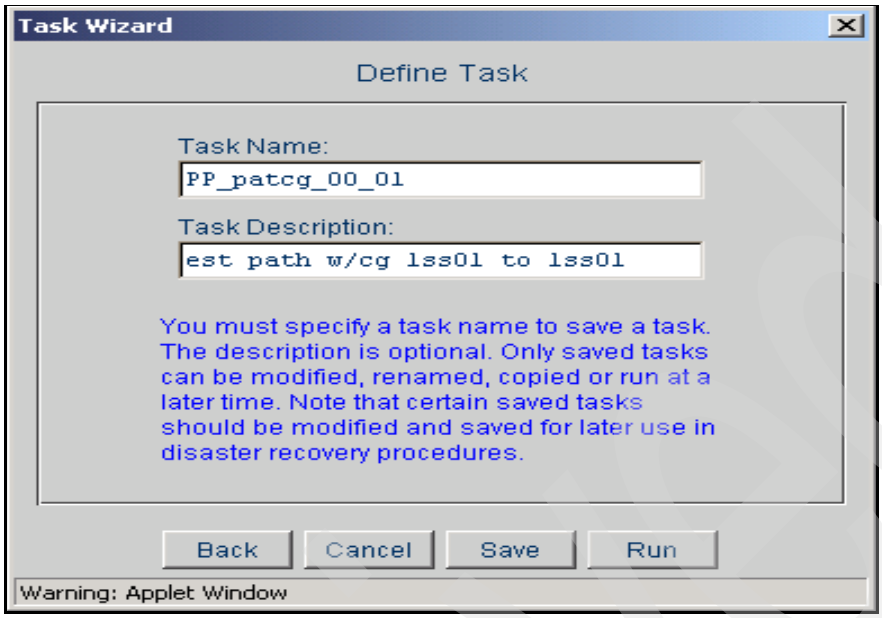


Figure 17-7 Define task

17.2.3 Define task to establish PPRC volume pair

1. In order to define a PPRC volume pair, you open the volume panel and specify at the top of the panel source and target LSS. All the related volumes are listed. Click **Enter Multiple Selection Mode** to specify multiple volume pairs. Click a source volume and right-click a related target volume and repeat the same process for each volume pair. See Figure 17-8.

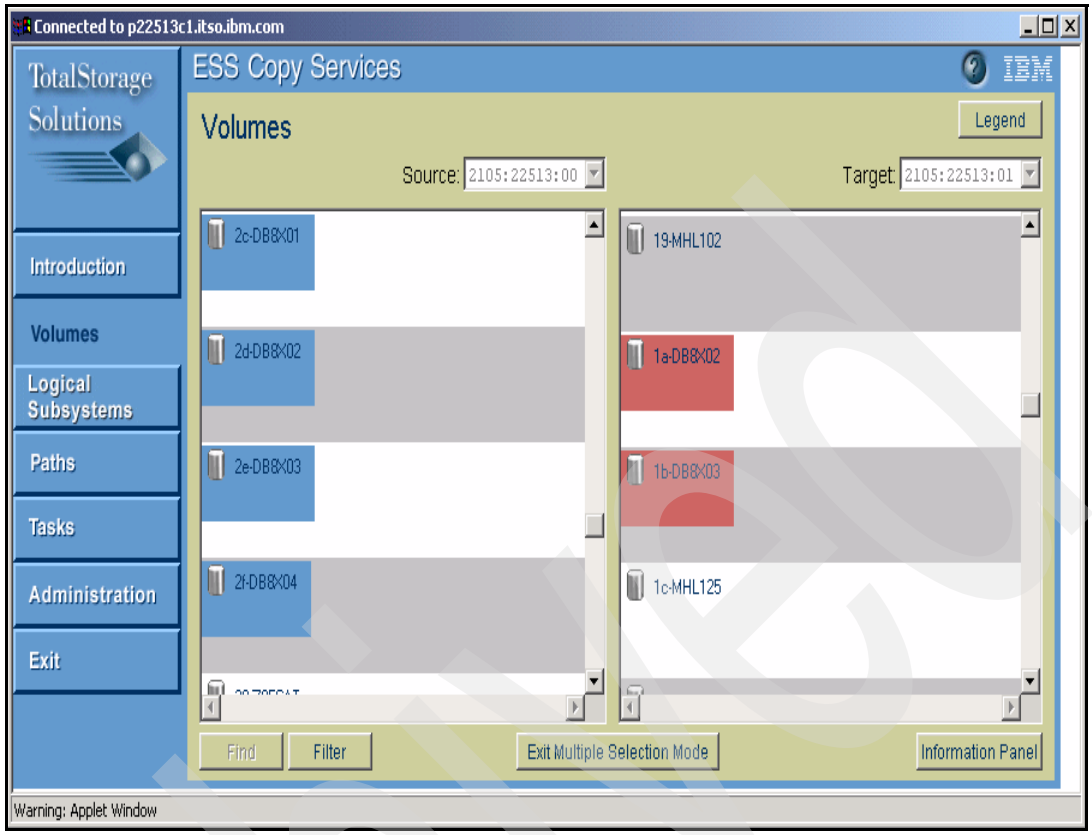


Figure 17-8 Select Source and Target volumes

2. After the last target volume is selected, right-click again and select **Establish Synchronous PPRC copy pair**. See Figure 17-9.

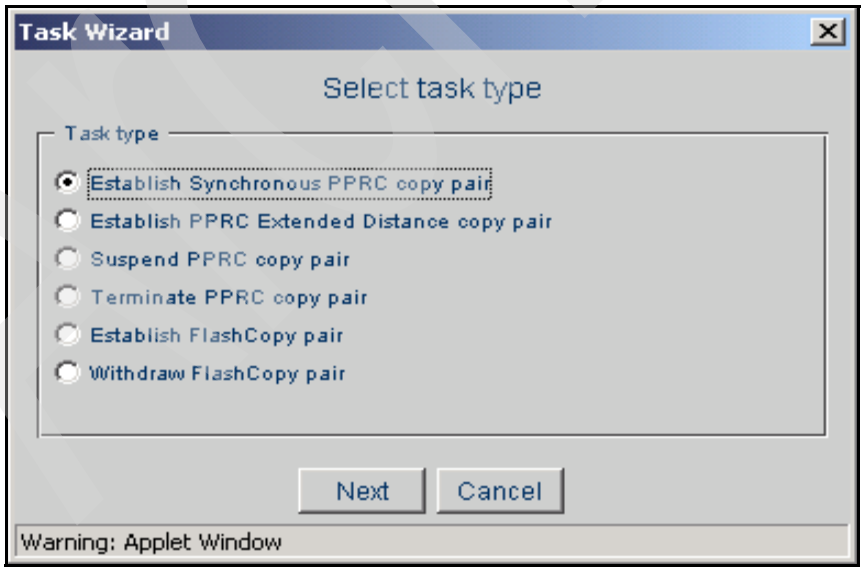


Figure 17-9 Establish synchronous PPRC copy pair

3. Select **Copy Entire Volume** and check **Permit establish if target is online**. Specify this option if you want to establish a PPRC pair even if target volume is online (Figure 17-10).

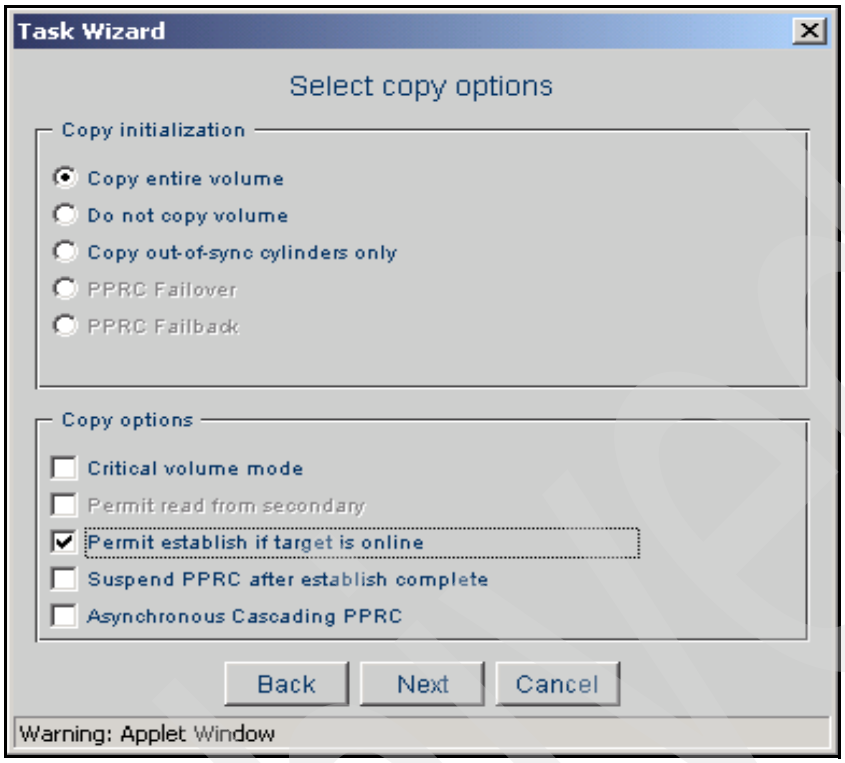


Figure 17-10 Copy option

4. Specify an arbitrary **Task Name** and **Description** and click **Save**. See the example shown in Figure 17-11.

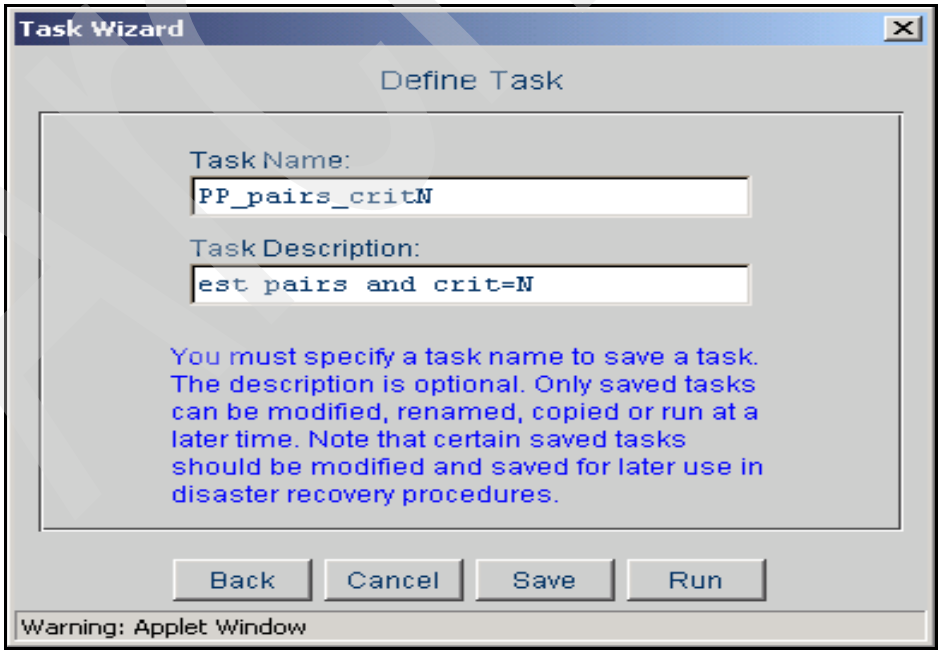


Figure 17-11 Define task

17.3 Scenario 1: Using FlashCopy Consistency Group

In this scenario we tested the possibility of using the FlashCopy Consistency Group feature to maintain consistency of the remote site copy. With the PPRC source and target volume pairs synchronized, we run a DB2 workload. We then run FlashCopy with Consistency Group from the PPRC remote volumes to the third set of volumes while the workload is active. And then we restart from the third set of volumes (FlashCopy target volumes).

The procedure is as follows:

1. Establish PPRC path lss00 to lss01 without consistency group (Task No.1*).
2. Establish pairs and critical volume=no (Task No.3*).
3. Start the DB2 workload JCL.
4. FlashCopy inbound from PPRC source volume with Consistency Group (Task No.12*).
5. FlashCopy Consistency Group Created (Task No.15*).
6. Stop the DB2 subsystem.
7. Delete pairs in the primary/source LSS (Task No.6*).
8. Remove path lss00 to lss01 without the force option (Task No.4*).
9. Switch volumes from the source PPRC to the FlashCopy target:
   a. Volume offline source PPRC volumes.
   b. Volume online source PPRC FlashCopy target volumes.
10. Restart DB2 and see what happens.

*The tasks refer to Table 17-1 on page 302.

We now describe each task.

17.3.1 Establish PPRC path

To establish and start PPRC path, run Task No.1 ‘PP_patng_00_01’. Make sure that in the task panel, the task state is displayed as *successful*.

- For how to define the task, refer to 17.2.2, “Define task to Establish PPRC Path" on page 304.

17.3.2 Establish PPRC volume pair

To establish and start the PPRC volume pair, run Task No.2 ‘PP_pairs_critN’. It will take several minutes for the PPRC pairs to be synchronized. In the Volumes panel we can see the status of volume initialization. While synchronization is going on, the volume icons are as shown in Figure 17-12.

- For details on how to define the task, refer to 17.2.3, “Define task to establish PPRC volume pair" on page 307.

Tip: The PPRC and FlashCopy volume status shown graphically in Figure 17-12 is obtained by using a filter on these volumes. To filter, you just select “filter” on the volume panel in the ESS Web Interface and select *PPRC, FlashCopy, Source,* and *Target*, depending on the function you want to see.

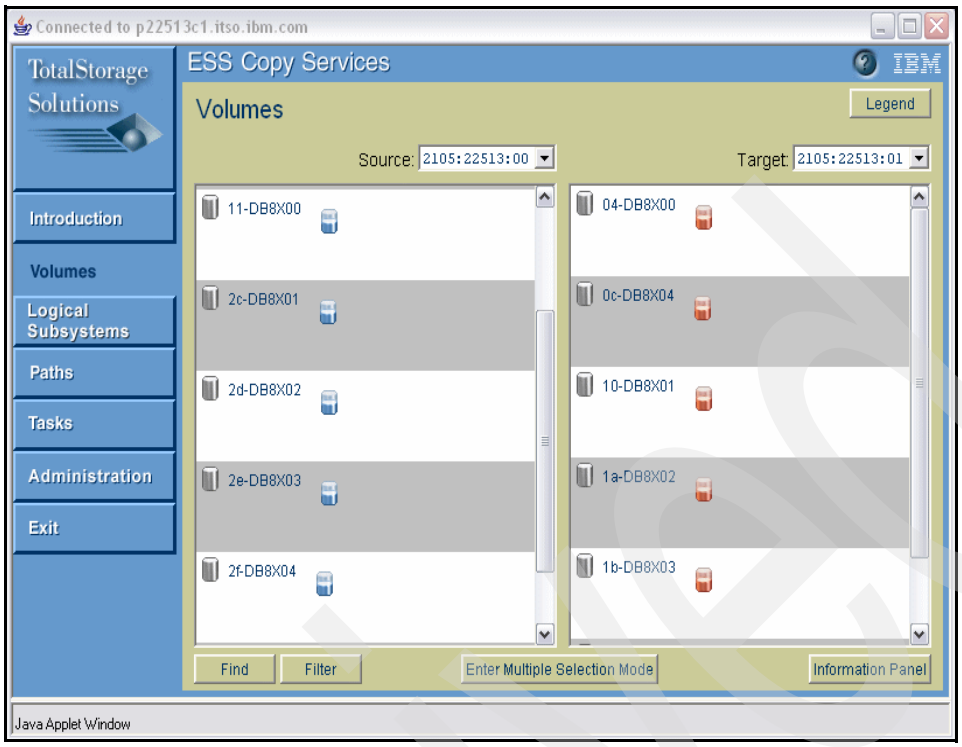


Figure 17-12 Synchronizing volume pair

11. You can make sure that the synchronization is completed when the icons of the volumes have changed as shown in Figure 17-13.

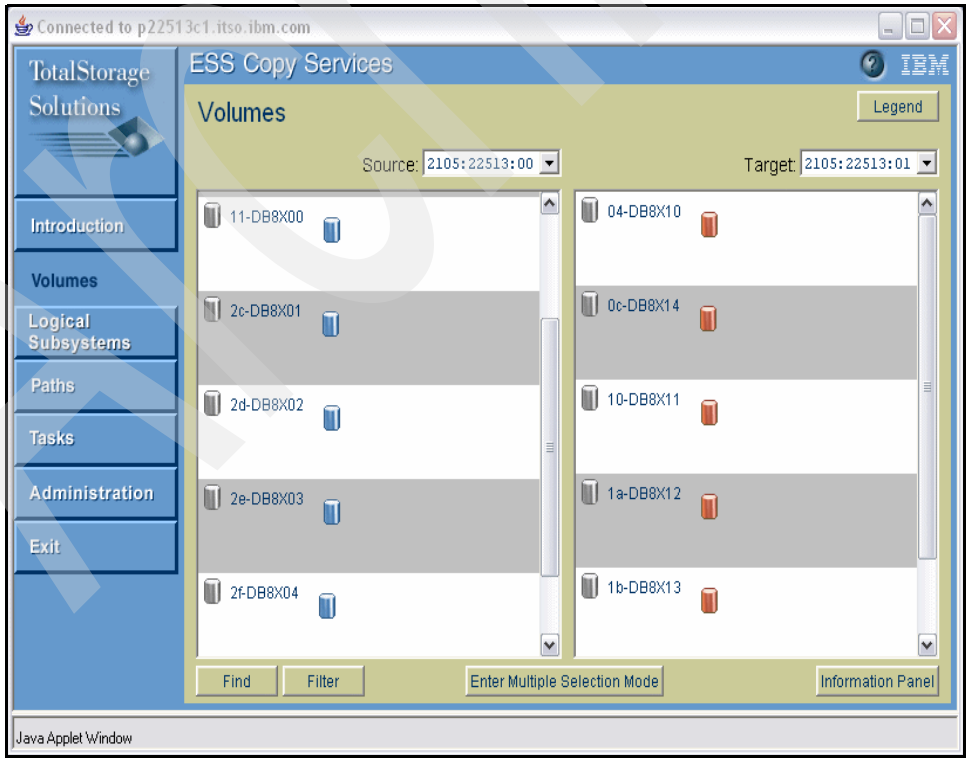


Figure 17-13 Volume pair synchronized

17.3.3 Start DB2 workload JCL

We now submit the DB2 workload JCL. We use two jobs, and each job updates a different database with several tasks which include Create schema, Load data, Copy, Reorg, etc. You should wait until this workload reaches the Load data step or Reorg, and as soon as either of these steps begins, run the FlashCopy task as described in 17.3.4, “FlashCopy with Consistency Group".

17.3.4 FlashCopy with Consistency Group

1. Run Task No.12 ’FC_01_02_ib_CG’. This task will do a FlashCopy from the PPRC target (FlashCopy source) to the FlashCopy target with Consistency Group using inbound command from PPRC source LSS. In this test scenario, although we create a Consistency Group using FlashCopy instead of the PPRC function, the FlashCopy command is issued by the PPRC primary via PPRC communication.
   - Via Consistency Group FlashCopy, all the PPRC source volumes will enter Extended Long Busy status. Updates from the workload are not allowed in this status. In the MVS log you can check the messages as shown in Figure 17-1.

Example 17-1 Extended Long Busy State in MVSLOG

```
IEA494I 8011,DB8X00,PPRC PAIR SUSPENDED,SSID=8910,CCA=11,EXTENDED LONG BUSY STATE
IEA494I 802F,DB8X04,PPRC PAIR SUSPENDED,SSID=8910,CCA=2F,EXTENDED LONG BUSY STATE
IEA494I 802C,DB8X01,PPRC PAIR SUSPENDED,SSID=8910,CCA=2C,EXTENDED LONG BUSY STATE
IEA494I 802D,DB8X02,PPRC PAIR SUSPENDED,SSID=8910,CCA=2D,EXTENDED LONG BUSY STATE
IEA494I 802E,DB8X03,PPRC PAIR SUSPENDED,SSID=8910,CCA=2E,EXTENDED LONG BUSY STATE
```

Note: There are two ways to reset the Extended Long Busy Status. One is to wait until the time-out interval expires (default value is set to 2 minutes). The other way is that after the FlashCopy is logically completed, you run the Consistency Created task which closes the time-out window.

   - At the same time, the DB2 workload job is suspended. We can see in the SDSF monitor that there is no I/O activity since FlashCopy is issued. See Figure 17-2.

Example 17-2 SDSF DA OJOB during FlashCopy”

```
SDSF DA SC64  SC64     PAG    0 SIO     4 CPU  23/ 21  LINE 1-2 (2)
COMMAND INPUT ===> DA OJOB                                SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     PAOLOR51 PH01S18  DSNUPROC JOB13577 PAOLOR5  A NS  F5  939   0.00   0.00
     PAOLOR53 PH01S13  DSNUPROC JOB13578 PAOLOR5  A NS  F5  16T   0.00   0.00
```

2. Make sure that the FlashCopy with Consistency Group is logically completed. You can check the volume status via the Web Interface as shown in Figure 17-14.

Note: In the Web Interface image of Figure 17-14, if the lightning signs appear next to all of both source and target volumes, it means that the FlashCopy is logically completed.

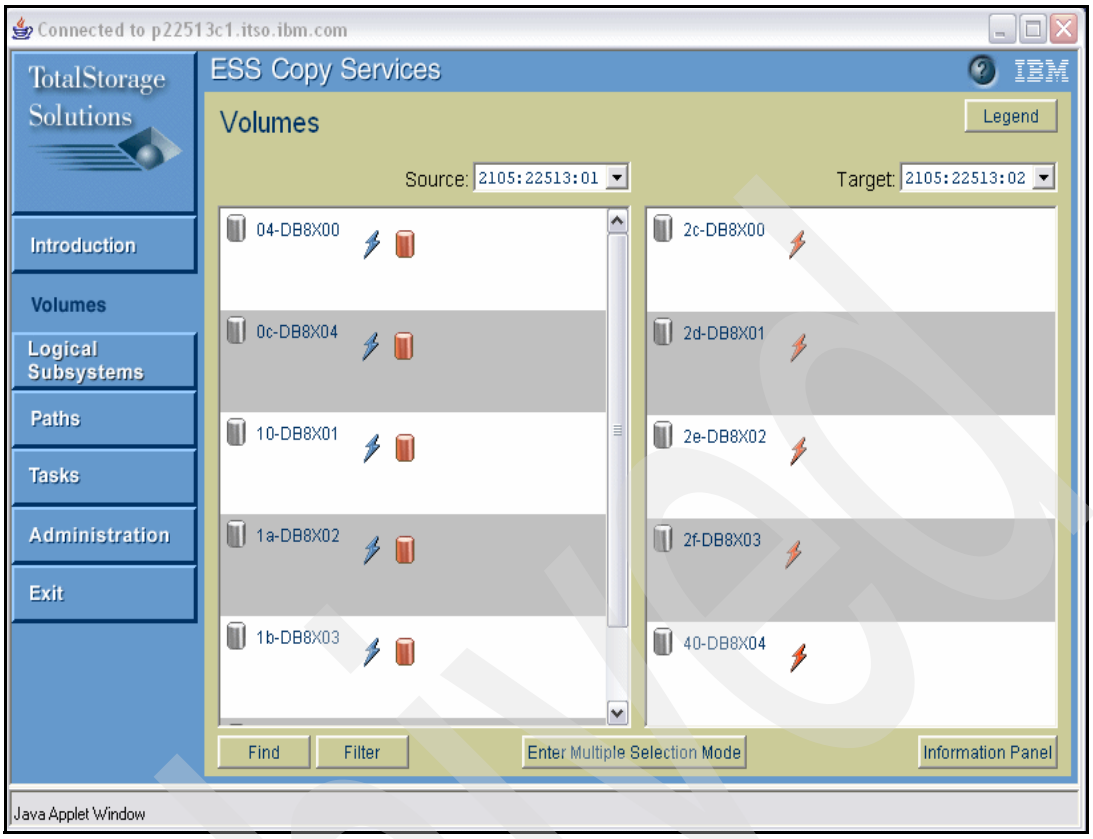


Figure 17-14 FlashCopy PPRC target to FlashCopy target

17.3.5 FlashCopy Consistency Group Created task

We now run Task No.15 'FC_CG_created'. This task will tell ESS that the FlashCopy task is finished, and will reset the Extended Long Busy state. After the task is finished, the DB2 workload will resume and I/O activity will start again, as shown in Example 17-3. Then the DB2 workload should finish successfully.

Example 17-3 SDSF DA OJOB after FlashCopy completes

```
Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF DA SC64  SC64     PAG    0 SIO    47 CPU   9/  9  LINE 1-2 (2)
COMMAND INPUT ===>                                          SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     PAOLOR51 PH01S19  DSNUPROC JOB13577 PAOLOR5  A NS  F9  17T   0.00   9.90
     PAOLOR53 PH01S13  DSNUPROC JOB13578 PAOLOR5  A NS  F9  948   0.28   6.33
```

17.3.6 Stop the DB2 subsystem

After the DB2 workload is finished, we stop the DB2 subsystem as shown in Example 17-4.

Example 17-4 Stop DB2

```
/-Q81X STO DB2
 15.38.30 STC21379  DSNL005I  -DB8X DDF IS STOPPING
 15.38.30 STC21379  DSNY002I  -DB8X SUBSYSTEM STOPPING
 15.38.30 STC21379  DSNL006I  -DB8X DDF STOP COMPLETE
 15.38.55 STC21379  DSN9022I  -DB8X DSNYASCP 'STOP DB2' NORMAL COMPLETION
```

17.3.7 Stop the PPRC pair

We now run Task No. 6 'PP_stop_pairs_p'. This task terminates a PPRC pair; the PPRC relationship between source and target volume ends.

17.3.8 Terminate PPRC path

We run Task No. 4 'PP_rmpath_00_01' to remove the PPRC path without the Force option. You specify the *Force* option, if you want to delete a path even though there are PPRC volumes pairs.

17.3.9 Vary offline the primary volumes

We vary offline the PPRC source volumes as shown in Example 17-5.

Example 17-5 Offline volumes

```
/V 8011,OFFLINE
RESPONSE=SC64      IEE302I 8011     OFFLINE
/V 802C-802F,OFFLINE
RESPONSE=SC64
 IEE457I 18.50.32 UNIT STATUS 994
 UNIT TYPE STATUS        VOLSER     VOLSTATE
 802C 3390 F-NRD                       /RSDNT
 802D 3390 F-NRD                       /RSDNT
 802E 3390 F-NRD                       /RSDNT
 802F 3390 F-NRD                       /RSDNT
```

Note: If the user catalog of DB2 is located in any of these volumes, you have to unallocate the ICF catalog before offline the volumes in advance.

17.3.10 Vary online the volumes at the recovery site

We now vary online the FlashCopy target volumes as shown in Example 17-6.

Example 17-6 Online volumes

```
/V 822C-822F,ONLINE
RESPONSE=SC64
 IEE457I 18.51.51 UNIT STATUS 453
 UNIT TYPE STATUS        VOLSER     VOLSTATE
 822C 3390 O             DB8X00     PRIV/RSDNT
 822D 3390 O             DB8X01     PRIV/RSDNT
 822E 3390 O             DB8X02     PRIV/RSDNT
 822F 3390 O             DB8X03     PRIV/RSDNT
/V 8240,ONLINE
RESPONSE=SC64      IEE302I 8240     ONLINE
```

17.3.11 Restart DB2 at recovery site

Restart the DB2 subsystem from the FlashCopy target volumes. We assume that the target volumes are physically at the remote site, and we restart the subsystem at the remote recovery site. When restarting the subsystem, there are one in-commit and one inflight transactions which DB2 resolves automatically by applying log forward and backward. Next, the DB2 restart log in our test is shown in Example 17-7.

Example 17-7 DB2 restart log

```
1                 J E S 2  J O B  L O G  --  S Y S T E M  S C 6 4  --  N O D E  W T S C P L X 2
0
 13.48.06 STC21340 ---- FRIDAY,    30 JUL 2004 ----
 13.48.06 STC21340  IEF695I START DB8XMSTR WITH JOBNAME DB8XMSTR IS ASSIGNED TO USER STC     , GROUP
SYS1
 13.48.06 STC21340  $HASP373 DB8XMSTR STARTED
 13.48.06 STC21340  IEF403I DB8XMSTR - STARTED - TIME=13.48.06 - ASID=04CF - SC64
 13.48.06 STC21340  DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE NAME IS
DSNZPARM
 13.48.07 STC21340  S DB8XIRLM
 13.48.08 STC21340  DSNY001I  -DB8X SUBSYSTEM STARTING
 13.48.08 STC21340  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  759
    759             IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
 13.48.08 STC21340  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  760
    760             IEC161I DB8XU.BSDS01.INDEX,UCAT.DB8XL
 13.48.08 STC21340  IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  761
    761             IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
 13.48.08 STC21340  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  762
    762             IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
 13.48.08 STC21340  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  763
    763             IEC161I DB8XU.BSDS02.INDEX,UCAT.DB8XL
 13.48.08 STC21340  IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  764
    764             IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
 13.48.09 STC21340  DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.212 13:32:13.52
 13.48.10 STC21340  DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  766
    766             DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS02,
    766             STARTRBA=00008913A000,ENDRBA=00008B2F9FFF
 13.48.10 STC21340  DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  767
    767             DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS02,
    767             STARTRBA=00008913A000,ENDRBA=00008B2F9FFF
 13.48.10 STC21340  DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  768
    768             STARTRBA=000089CA2000
 13.48.10 STC21340  S DB8XDBM1
 13.48.13 STC21340  S DB8XDIST
 13.48.15 STC21340  DSNR001I  -DB8X RESTART INITIATED
 13.48.15 STC21340  DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=000089146C3F
 13.48.15 STC21340  DSNR004I  -DB8X RESTART...UR STATUS COUNTS  781
    781             IN COMMIT=1, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
 13.48.15 STC21340  DSNR007I  -DB8X RESTART...STATUS TABLE  782
    782             T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY  TIME
    782             - -------- ------------ -------- -------- - ------------ --- --------
    782             B UTILITY  PAOLOR31     PAOLOR3  DSNUTIL  F 000089C84386 212 13:32:15
    782             B UTILITY  PAOLOR32     PAOLOR3  DSNUTIL  C 000089CA0399 212 13:32:16
 13.48.22 STC21340  DSNI001I  -DB8X RESTART HAS BEEN DEFERRED  784
    784                        REASON 00C90095
    784                        TYPE 00000200
    784                        NAME DRTESTDB.DSN8S81D
 13.48.22 STC21340  DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  785
    785             THE LOGICAL PAGE LIST
    785                        DATABASE NAME=DRTESTDB
    785                        SPACE NAME=DSN8S81D
    785                        DATA SET NUMBER=1
    785                        PAGE RANGE X'00000000' TO X'00000000'
    785                        START LRSN=X'000089C84416'
    785                        END LRSN=X'000089CA16C8'
    785                        START RBA=X'000089C84416'
```

```
   785                        LPL TRACE ID=00000003
   785                        LPL REASON TYPE=LOGAPPLY
   785             …
13.48.22 STC21340  DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  786
   786             THE LOGICAL PAGE LIST
   786                        DATABASE NAME=DRTESTDB
   786                        SPACE NAME=DSN8S81D
   786                        DATA SET NUMBER=1
   786                        PAGE RANGE X'00000001' TO X'00000001'
   786                        START LRSN=X'000089C85000'
   786                        END LRSN=X'000089CA16C8'
   786                        START RBA=X'000089C85000'
   786                        LPL TRACE ID=00000003
   786                        LPL REASON TYPE=LOGAPPLY
   786             …
13.48.22 STC21340  DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  787
   787             RECOVERY
   787             IN COMMIT=0, INDOUBT=0
13.48.22 STC21340  DSNR018I  -DB8X RESTART...BACKWARD RECOVERY PROCESSED  788
   788              FROM RBA 000089C86079 TO RBA 000089C84386
13.48.22 STC21340  DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  789
   789             RECOVERY
   789             INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
13.48.22 STC21340  DSNR002I  -DB8X RESTART COMPLETED
13.48.22 STC21340  -DB8XRECOVER POSTPONED
13.48.22 STC21340  DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
13.48.22 STC21340  DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
13.48.22 STC21340  DSNL003I  -DB8X DDF IS STARTING
13.48.24 STC21340  DSNL519I  -DB8X DSNLILNR TCP/IP SERVICES AVAILABLE  795
   795                        FOR DOMAIN wtsc64.itso.ibm.com AND PORT 38080
13.48.24 STC21340  DSNL004I  -DB8X DDF START COMPLETE  796
   796                        LOCATION  DB8X
   796                        LU        USIBMSC.SCPDB8X
   796                        GENERICLU -NONE
   796                        DOMAIN    wtsc64.itso.ibm.com
   796                        TCPPORT   38080
   796                        RESPORT   38081
13.48.24 STC21340  DSN9022I  -DB8X DSNYASCP 'STA DB2' NORMAL COMPLETION
```

17.3.12 Check objects in restricted status

After restarting DB2, you have to recover all objects in restricted status. If FlashCopy with Consistency Group was issued at the primary site, multiple utility tasks (especially Load, Reorg, utilities, etc.) were suspended, and it is natural that DB2 at the remote site has many restricted objects:

1. Check for Utility status with **`Display Utility`**. See Example 17-8. In this case you can see that one job was stopped during Load utility's reload phase, the other job was stopped during Reorg utility's reload phase.

> **Attention:** The active utilities and restricted status may vary depending on the active workload suspended while FlashCopy. The following output also can be different by the timing and environment.

Example 17-8 Display utility

```
-DIS UTIL(*)
DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = DSNTEX
                PROCESSING UTILITY STATEMENT 1
                UTILITY = REORG
                PHASE = RELOAD   COUNT = 0
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = STOPPED
 DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = DSNTEY
                PROCESSING UTILITY STATEMENT 3
                UTILITY = LOAD
                PHASE = RELOAD   COUNT = 0
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = STOPPED
 DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
 ***
```

2. Check for DB in restricted status with **Display Database**. See Example 17-9. It shows that there are two CHKP (check pending) table spaces, one RECP (recover pending) table space, and three RBDP (rebuild pending) indexes.

Example 17-9 Display database

```
-DIS DB(*) SP(*) RESTRICT LIMIT(*)
DSNT360I  -DB8X ***********************************
DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                *   RESTRICTED
DSNT360I  -DB8X ***********************************
DSNT362I  -DB8X     DATABASE = DRTESTDB  STATUS = RW
                   DBD LENGTH = 24218
DSNT397I  -DB8X
NAME     TYPE PART  STATUS           PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ---------------- -------- -------- -------- -----
DSN8S81D TS         RW,RECP,LPL,UTUT
XDEPT1   IX         RW,RBDP,UTUT
XDEPT2   IX         RW,RBDP,UTUT
XDEPT3   IX         RW,RBDP,UTUT
******* DISPLAY OF DATABASE DRTESTDB ENDED      **********************
DSNT360I  -DB8X ***********************************
DSNT362I  -DB8X     DATABASE = DRTESTD2  STATUS = RW
                   DBD LENGTH = 24218
DSNT397I  -DB8X
NAME     TYPE PART  STATUS           PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ---------------- -------- -------- -------- -----
DSN8S81D TS         RW,CHKP
DSN8S81P TS         RW,UTUT
XPROJ1   IX         RW,UTUT
 XPROJ2   IX         RW,UTUT
 ******* DISPLAY OF DATABASE DRTESTD2 ENDED      **********************
```

17.3.13 Procedures to recover objects from restricted status

The action required varies depending on how the DB2 workload was stopped and which utilities and phases were going on. First of all check the DB2 utility and object status as shown in Example 17-8 and Example 17-9. Then select the action corresponding to the situation.

Terminate active utilities

By terminating utilities, you clean up utilities and release all related resources. If the *stopped* status of the utilities of your executed job remains, do the step shown in Example 17-10.

Example 17-10 Terminate utility

```
/-DB8X TERM UTIL(DSNTE*)
RESPONSE=SC64
 DSNU166I  -DB8X DSNUGTER - LOAD UTILITY,
                 UTILID = DSNTEY NOT EXECUTING,
                 CLEANUP COMPLETE
 DSNU166I  -DB8X DSNUGTER - REORG UTILITY,
                 UTILID = DSNTEX NOT EXECUTING,
                 CLEANUP COMPLETE
 DSN9022I  -DB8X DSNUGCCC '-TERM UTIL' NORMAL COMPLETION
```

Reset RECP (recovery pending) status

To reset the recovery pending status, you need to run the Recover utility for each table space in pending status. Example 17-11 shows a sample job to recover table spaces.

Example 17-11 Recovery pending resolution

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEB'
//SYSIN    DD  *
RECOVER     TABLESPACE DRTESTDB.DSN8S81D
RECOVER     TABLESPACE DRTESTD2.DSN8S81P
```

Reset REBP (rebuild pending) status

To reset the rebuild pending status, you need to run the Rebuild Index utility for the indexes in pending status. Example 17-12 shows a sample job.

Example 17-12 Rebuild index

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEA'
//SYSIN    DD  *
REBUILD INDEX
    ( PAOLOR3.PXC#CKNKMS,
      PAOLOR3."PXO#OKODCKSPOP",
      PAOLOR3.PXP#PKPTPN,
      PAOLOR3.SXP@SZPTPKMF,
      PAOLOR3.SXRI#CUS,
      PAOLOR3.SXRI#ORD,
      PAOLOR3.SXRI#PRT,
      PAOLOR3.UXC@NKCK,
      PAOLOR3.UXO#CLOKOD  )
```

Reset COPY (copy pending) status

There are two ways to reset the copy pending status. The most correct action is to take an image copy. But it is likely that there is not enough data set space to take an image copy in the recovery site. In this case it is recommended to use the Repair utility. Example 17-13 shows a Repair sample job.

Example 17-13 Repair Set Nocopypend

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEB'
//SYSIN    DD  *
REPAIR SET  TABLESPACE PAOLODB.TSSUPLY   NOCOPYPEND
```

Reset CHKP (check pending) status

There are two ways to reset the check-pending status; you can use the ‘Check Data utility or the Repair utility with Nocopypend. From the point of view of data consistency, Check Data is preferred, but you can choose the Repair utility in case you want to save time. Example 17-14 shows an example of Check Data.

Example 17-14 Check Data

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEY'
//SYSUT1   DD  UNIT=VIO,SPACE=(4000,(20,20),,,ROUND)
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTOUT  DD  UNIT=VIO,SPACE=(4000,(20,20),,,ROUND)
//DSNTRACE DD  SYSOUT=*
//SYSERR   DD  UNIT=VIO,SPACE=(4000,(20,20),,,ROUND)
//SYSIN    DD  *

  CHECK DATA TABLESPACE DRTESTD2.DSN8S81D
             TABLESPACE DRTESTD2.DSN8S81E
             TABLESPACE DRTESTD2.DSN8S81P
       SCOPE ALL
       DELETE YES
       FOR EXCEPTION IN DRTEST2.DEPT       USE DRTEST2.EDEPT
                     IN DRTEST2.EMP        USE DRTEST2.EEMP
                     IN DRTEST2.PROJ       USE DRTEST2.EPROJ
                     IN DRTEST2.ACT        USE DRTEST2.EACT
                     IN DRTEST2.PROJACT    USE DRTEST2.EPROJACT
                     IN DRTEST2.EMPPROJACT USE DRTEST2.EEPA
     EXCEPTIONS 50
     SORTDEVT SYSDA SORTNUM 4
```

Example 17-15 shows the Repair job.

Example 17-15 Repair Set Nocheckpend

```
//PH01S01 EXEC DSNUPROC,PARM='DB8X,DSNTEB'
 //SYSIN    DD  *
 REPAIR SET TABLESPACE DRTESTD2.DSN8S81D NOCHECKPEND
        SET TABLESPACE DRTESTD2.DSN8S81P NOCHECKPEND
```

Make sure all the restricted status objects are recovered

Make sure that all restricted objects are recovered successfully and that there is no restriction for all databases, as shown in Example 17-16.

Example 17-16 Display Database

```
DSNT365I  -DB8X    NO DATABASES FOUND
DSN9022I  -DB8X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

17.4 Scenario 2: Using DB2 Set Log Suspend

In this scenario we tested the possibility of using **`DB2 Set Log Suspend`** to maintain consistency of the remote site copy. With PPRC source and target volume pairs synchronized run the DB2 workload and issue the **`DB2 Set Log Suspend`**. While DB2 suspends update and log activity, run FlashCopy without consistency group from the PPRC remote volumes to the third set of volumes and issue **`DB2 Set Log resume`** to restart the DB2 workload. And then try to restart from the third set of volumes (FlashCopy target volumes).

The procedure is as follows:

1. Establish PPRC path lss00 to lss01 without consistency group (Task No.1*).
2. Establish pairs and critical volume=no (Task No.3*).
3. Start the DB2 workload JCL.
4. DB2 ‘Set Log Suspend’.
5. FlashCopy inbound from PPRC source volume without Consistency Group (Task No.13*).
6. DB2 ‘Set Log Resume’.
7. Stop the DB2 subsystem.
8. Delete pairs in primary/source LSS (Task No.6*).
9. Remove path lss00 to lss01 without force option (Task No.4*).
10. Switch volumes from source PPRC to FlashCopy target:
    a. Volume offline source PPRC volumes.
    b. Volume online source PPRC FlashCopy target volumes.
11. Restart DB2 and see what happens.

*The tasks refer to Table 17-1 on page 302.

We now describe each task.

17.4.1 Establish PPRC path

The procedure refers to 17.3.1, “Establish PPRC path" on page 310

17.4.2 Establish PPRC volume pair

The procedure refers to 17.3.2, “Establish PPRC volume pair" on page 310.

17.4.3 Start the DB2 workload

Submit the DB2 workload JCL. We use two jobs, and each job updates different databases with several tasks that include Create schema, LOAD data, Copy, Reorg, etc. You should wait until this workload reaches the ‘Load Data’ step or ‘Reorg’. As soon as either of these steps begins, issue **`Set Log Suspend`** as described in 17.4.4, “DB2 Set Log Suspend".

17.4.4 DB2 Set Log Suspend

Issue the DB2 command **`Set Log Suspend`** in MVS console as described in Example 17-17.

Example 17-17 .DB2 Set Log Suspend

```
DSNJ372I  -DB8X DSNJC009 UPDATE ACTIVITY HAS BEEN SUSPENDED FOR DB8X
AT RBA 00005C65CE60, LRSN 00005C65CE60, PRIOR CHECKPOINT RBA
00005C657644
DSN9022I  -DB8X DSNJC001 '-SET LOG' NORMAL COMPLETION
```

After the command has executed, the DB2 workload will be suspended. You can see in Example 17-18 that there is no I/O activity in these jobs.

Example 17-18 SDSF ‘ DA OJOB during FlashCopy”

```
SDSF DA SC64  SC64     PAG    0 SIO     4 CPU  23/ 21  LINE 1-2 (2)
COMMAND INPUT ===> DA OJOB                                  SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     PAOLOR51 PH01S18  DSNUPROC JOB13577 PAOLOR5  A NS  F5  939   0.00   0.00
     PAOLOR53 PH01S13  DSNUPROC JOB13578 PAOLOR5  A NS  F5  16T   0.00   0.00
```

17.4.5 FlashCopy from the secondary to the third volumes

1. Run Task No.13 ’FC_01_02_ib_NC’. This task will do FlashCopy from PPRC target (FlashCopy source) to the FlashCopy target using inbound command from PPRC source LSS. In this test scenario we do not create Consistency Group with FlashCopy because we have already created the consistency group with the previous **`Set Log Suspend`** command.
2. Make sure that FlashCopy is logically completed. You can see this at the volume status of the Web Interface as shown in Example 17-14 on page 313.

Note: There are some cases where you need to consider dumping the secondary volumes to tape as a substitute for FlashCopy. When the PPRC primary and secondary volume pairs are established, the volume IDs of the primary volumes are clipped to secondary volumes, and the secondary volumes are set offline. You need to change volume ID first and then dump to tape. The details about how to do this are discussed in 16.6, “Dumping the FlashCopy" on page 292.

17.4.6 DB2 Set Log Resume

Issue the DB2 command **`Set Log Resume`** from the MVS console as shown in Example 17-19.

Example 17-19 Set Log Resume

```
DSNJ373I  -DB8X DSNJC009 UPDATE ACTIVITY HAS BEEN RESUMED FOR DB8X
DSN9022I  -DB8X DSNJC001 '-SET LOG' NORMAL COMPLETION
```

After the command is executed, DB2 workload will resume and the I/O activity will start again as shown in Example 17-20, then the DB2 workload will complete successfully.

Example 17-20 SDSF ‘ DA OJOB after FlashCopy completes

```
Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC64  SC64     PAG    0 SIO    47 CPU   9/  9  LINE 1-2 (2)
COMMAND INPUT ===>                                          SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     PAOLOR51 PH01S19  DSNUPROC JOB13577 PAOLOR5  A NS  F9  17T   0.00   9.90
     PAOLOR53 PH01S13  DSNUPROC JOB13578 PAOLOR5  A NS  F9  948   0.28   6.33
```

17.4.7 Stop the DB2 subsystem

After the B2 workload has finished, stop the DB2 subsystem as shown in Example 17-4 on page 313.

17.4.8 Stop the PPRC pair

Run Task No. 6 'PP_stop_pairs_p'. This task terminates a PPRC pair, and the PPRC relationship between source and target volume ends.

17.4.9 Terminate the PPRC path

Run Task No.4 'PP_rmpath_00_01' to remove the PPRC path without force option. Specify the Force option, if you want to delete the path even when there are PPRC pair volumes.

17.4.10 Vary offline the primary volumes

Vary offline the PPRC source volumes. See Example 17-5 on page 314.

> **Note:** If the user catalog of DB2 is located in anyone of these volumes, you have to unallocate the ICF catalog before varying offline the volumes.

17.4.11 Vary online the recovery site volumes

Vary online the FlashCopy target volumes. See Example 17-6 on page 314.

17.4.12 Restart DB2 at the recovery site

Restart the DB2 subsystem from the FlashCopy target volumes. We assume that the target volumes are physically at the remote site, and restart the subsystem at the remote recovery site. When restarting the subsystem, there are two inflight transactions which DB2 resolves automatically by applying log forward and backward. The DB2 restart log in our test is shown in Example 17-21.

Example 17-21 Restart Subsystem MVS LOG

```
1                 J E S 2  J O B  L O G  --  S Y S T E M  S C 6 4  --  N O D E  W T S C P L X 2
0
 14.32.32 STC21360 ---- FRIDAY,    30 JUL 2004 ----
 14.32.32 STC21360  IEF695I START DB8XMSTR WITH JOBNAME DB8XMSTR IS ASSIGNED TO USER STC     , GROUP
SYS1
 14.32.32 STC21360  $HASP373 DB8XMSTR STARTED
 14.32.32 STC21360  IEF403I DB8XMSTR - STARTED - TIME=14.32.32 - ASID=0087 - SC64
 14.32.32 STC21360  DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE NAME IS
DSNZPARM
 14.32.33 STC21360  S DB8XIRLM
 14.32.34 STC21360  DSNY001I  -DB8X SUBSYSTEM STARTING
 14.32.34 STC21360  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  817
    817             IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
 14.32.34 STC21360  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  818
    818             IEC161I DB8XU.BSDS01.INDEX,UCAT.DB8XL
 14.32.34 STC21360  IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  819
    819             IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
 14.32.34 STC21360  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  820
    820             IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
 14.32.34 STC21360  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  821
    821             IEC161I DB8XU.BSDS02.INDEX,UCAT.DB8XL
```

```
14.32.34 STC21360  IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  822
   822             IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
14.32.35 STC21360  DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.212 14:20:57.88
14.32.35 STC21360  DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  824
   824             DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS02,
   824             STARTRBA=00008913A000,ENDRBA=00008B2F9FFF
14.32.35 STC21360  DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  825
   825             DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS02,
   825             STARTRBA=00008913A000,ENDRBA=00008B2F9FFF
14.32.35 STC21360  DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  826
   826             STARTRBA=00008AAB5000
14.32.35 STC21360  S DB8XDBM1
14.32.37 STC21360  S DB8XDIST
14.32.38 STC21360  DSNR001I  -DB8X RESTART INITIATED
14.32.38 STC21360  DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=00008AAAA9C6
14.32.38 STC21360  DSNR004I  -DB8X RESTART...UR STATUS COUNTS  839
   839             IN COMMIT=0, INDOUBT=0, INFLIGHT=2, IN ABORT=0, POSTPONED ABORT=0
14.32.38 STC21360  DSNR007I  -DB8X RESTART...STATUS TABLE  840
   840             T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
   840             - -------- ------------ -------- -------- - ------------ --- --------
   840             B UTILITY  PAOLOR31     PAOLOR3  DSNUTIL  F 00008AA83EFE 212 14:20:59
   840             B UTILITY  PAOLOR32     PAOLOR3  DSNUTIL  F 00008AAB24B8 212 14:21:00
14.32.45 STC21360  DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  842
   842             RECOVERY
   842             IN COMMIT=0, INDOUBT=0
14.32.45 STC21360  DSNR018I  -DB8X RESTART...BACKWARD RECOVERY PROCESSED  843
   843              FROM RBA 00008AAB2548 TO RBA 00008AA83EFE
14.32.45 STC21360  DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  844
   844             RECOVERY
   844             INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
14.32.45 STC21360  DSNR002I  -DB8X RESTART COMPLETED
14.32.45 STC21360  -DB8XRECOVER POSTPONED
14.32.45 STC21360  DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
14.32.45 STC21360  DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
14.32.45 STC21360  DSNL003I  -DB8X DDF IS STARTING
14.32.47 STC21360  DSNL519I  -DB8X DSNLILNR TCP/IP SERVICES AVAILABLE  850
   850                       FOR DOMAIN wtsc64.itso.ibm.com AND PORT 38080
14.32.47 STC21360  DSNL004I  -DB8X DDF START COMPLETE  851
   851                       LOCATION  DB8X
   851                       LU        USIBMSC.SCPDB8X
   851                       GENERICLU -NONE
   851                       DOMAIN    wtsc64.itso.ibm.com
   851                       TCPPORT   38080
   851                       RESPORT   38081
14.32.47 STC21360  DSN9022I  -DB8X DSNYASCP 'STA DB2' NORMAL COMPLETION
```

17.4.13 Check the objects in restricted status

1. Check the Utility status with **DISPLAY UTILITY.** See Example 17-8. In our case you can see that one job was stopped during the Load utility build phase, and another one was stopped during the Reorg utility reload phase.

> **Attention:** The active utilities and restricted status may vary depending on the type of active workload that was suspended during the FlashCopy. The corresponding output will depend on timing and environment.

Example 17-22 Display utility

```
-DIS UTIL(*)
DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = DSNTEX
                PROCESSING UTILITY STATEMENT 1
                UTILITY = REORG
                PHASE = RELOAD   COUNT = 0
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = STOPPED
 DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR3
                MEMBER =
                UTILID = DSNTEY
                PROCESSING UTILITY STATEMENT 3
                UTILITY = LOAD
                PHASE = BUILD   COUNT = 0
                NUMBER OF OBJECTS IN LIST = 1
                LAST OBJECT STARTED = 1
                STATUS = STOPPED
 DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
***
```

2. Check for DB in restricted status with **DISPLAY DATABASE**. See Example 17-23. It shows that there is one copy pending table space, and several utility pending table spaces and indexes.

Example 17-23 Display database

```
-DIS DB(*) SP(*) RESTRICT LIMIT(*)
DSNT360I  -DB8X ************************************
DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                *    RESTRICTED
DSNT360I  -DB8X ************************************
DSNT362I  -DB8X     DATABASE = DRTESTD2  STATUS = RW
                   DBD LENGTH = 24218
DSNT397I  -DB8X
NAME     TYPE PART  STATUS           PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ---------------- -------- -------- -------- -----
DSN8S81D TS         RW,CHKP
DSN8S81P TS         RW,CHKP,UTUT
XPROJ1   IX         RW,UTUT
XPROJ2   IX         RW,RBDP,UTUT
******* DISPLAY OF DATABASE DRTESTD2 ENDED      **********************
DSNT360I  -DB8X ************************************
DSNT362I  -DB8X     DATABASE = DRTESTDB  STATUS = RW
                   DBD LENGTH = 24218
DSNT397I  -DB8X
NAME     TYPE PART  STATUS           PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ---------------- -------- -------- -------- -----
DSN8S81D TS         RW,RECP,UTUT
XDEPT1   IX         RW,RBDP,UTUT
XDEPT2   IX         RW,RBDP,UTUT
XDEPT3   IX         RW,RBDP,UTUT
******* DISPLAY OF DATABASE DRTESTDB ENDED      **********************
```

17.4.14 Procedures to recover the objects in restricted status

For how to reset restricted status, refer to 17.3.13, “Procedures to recover objects from restricted status" on page 317.

17.5 Scenario 3: Using PPRC Consistency Group

In this scenario we tested the possibility of using PPRC Consistency Group to maintain consistency of the remote site copy. While synchronous PPRC is established with the Consistency Group option, we run the DB2 workload and then run the **`Freeze`** task. While all the I/O activity on the volumes related to PPRC consistency group is suspended, we run FlashCopy without Consistency Group from the PPRC remote volumes to third volumes and run the **`PPRC consistency group created`** task to restart I/O activity. Then we restart from the third volumes (FlashCopy target volumes).

The procedure is as follows:

1. Establish PPRC path lss00 to lss01 with consistency group (Task No.2*).
2. Establish pairs and critical volume=no (Task no.3*).
3. Start the DB2 workload JCL.
4. PPRC Freeze task (Task no.10*).
5. FlashCopy from secondary to third volumes without Consistency Group (Task no.14*).
6. PPRC Consistency Group Created task (Task no.9).
7. Reestablish PPRC path (Task no.2*).
8. Resynchronize volume pairs (Task no.11).
9. Stop the DB2 subsystem.
10. Delete pairs in primary/source LSS (Task no.6*).
11. Remove path lss00 to lss01 without force option (Task no.4*).
12. Switch volumes from source PPRC to FlashCopy target.
    a. Volume Offline source PPRC volumes.
    b. Volume Online source PPRC FlashCopy target volumes.
13. Restart DB2 and see what happens.

* The task names refer to Table 17-1 on page 304.

17.5.1 Establish PPRC with Consistency Group

To establish and start PPRC path with PPRC Consistency Group, we run Task No.2 'PP_patcg_00_01'. Make sure that in the task panel, the task state is displayed as *successful*.

- How to define the task refers to 17.2.2, "Define task to Establish PPRC Path" on page 304.

Note on PPRC Consistency Group:

The PPRC Consistency group is created in volume pairs which have the same primary and secondary LSS. If an error occurs that affects any of those volumes, and if the volume in error is participating in the Consistency Group, the volume enters long busy state. The primary host temporarily queues all the updates to the primary volumes.

17.5.2 Establish PPRC volume pair

The procedure refers to 17.3.2, “Establish PPRC volume pair" on page 310.

17.5.3 Start the DB2 workload

Submit the DB2 workload JCL. We use two jobs, and each job update different database with several tasks that include Create Schema, Load Data, Copy, Reorg, etc. You should wait until this workload reaches the Load Data step or Reorg. As soon as either of these steps begins, run the PPRC freeze task as described in 17.5.4, “PPRC Freeze task with Consistency Group".

17.5.4 PPRC Freeze task with Consistency Group

Run Task No.10 ‘PP_freeze’ to freeze PPRC session. With the Freeze operation, all write activity on the active PPRC primary and secondary volumes will stop. In this scenario, the objective of running the Freeze operation is to maintain consistency across the secondary volumes that we will use for remote site recovery.

Note: What happens after the Freeze operation with PPRC Consistency Group?

- The paths that connect the pair of LSSs being frozen are terminated.
- The volumes under frozen LSSs are suspended.
- The write activity to the primary LSSs is temporarily queued.

- All the PPRC volume pairs will enter Long Busy State. It is the same situation as when running FlashCopy Consistency group in scenario 2. See Example 17-1 on page 312.
- As a result, the activity of the DB2 workload will be suspended, and no I/O activity happens after the Freeze operation, as shown in Example 17-24.

Example 17-24 SDSF DA OJOB during PPRC Freeze operation

```
SDSF DA SC64  SC64     PAG    0 SIO    25 CPU  15/ 14  LINE 1-2 (2)
COMMAND INPUT ===>                                          SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     PAOLOR51 PH01S18  DSNUPROC JOB13611 PAOLOR5  A NS  F9  929   0.00   0.00
     PAOLOR53 PH01S13  DSNUPROC JOB13612 PAOLOR5  A NS  F9  854   0.00   0.00
```

You can also view the PPRC volume status in the Web Interface volume panel while the PPRC source volumes are suspended. See Figure 17-15.

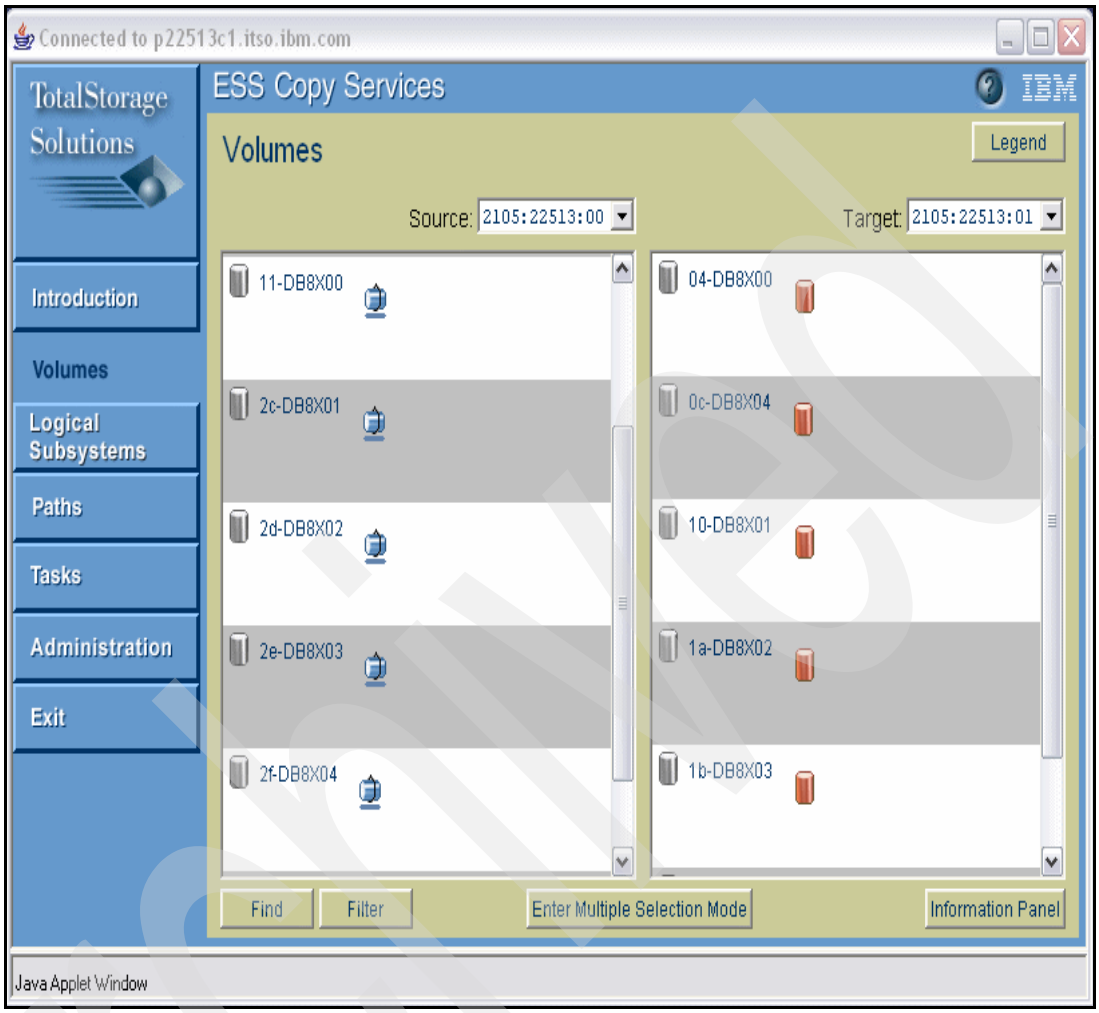


Figure 17-15 After Freeze task issued with PPRC Consistency Group

17.5.5 FlashCopy from secondary to third volumes

Run Task No.14'FC_01_02_NC'. This task will do FlashCopy from the PPRC target (FlashCopy source) to the FlashCopy target. In this test scenario, the FlashCopy is invoked from the FlashCopy target LSS instead of by inbound command from the PPRC source, because the PPRC path is terminated by the Freeze operation, and the communication between the PPRC source and the target LSS is disabled. Make sure that FlashCopy is logically completed. You can see the volume status via the Web Interface as shown in Figure 17-16.

Note: There are some cases where you need to consider dumping the secondary volumes to tape as a substitute of FlashCopy. When PPRC primary and secondary volume pairs are established, the volume IDs of primary volumes are clipped to secondary volumes, and the secondary volumes are set offline. You need to change the volume ID first and then dump to tape. The details are discussed in 16.6, “Dumping the FlashCopy" on page 292.

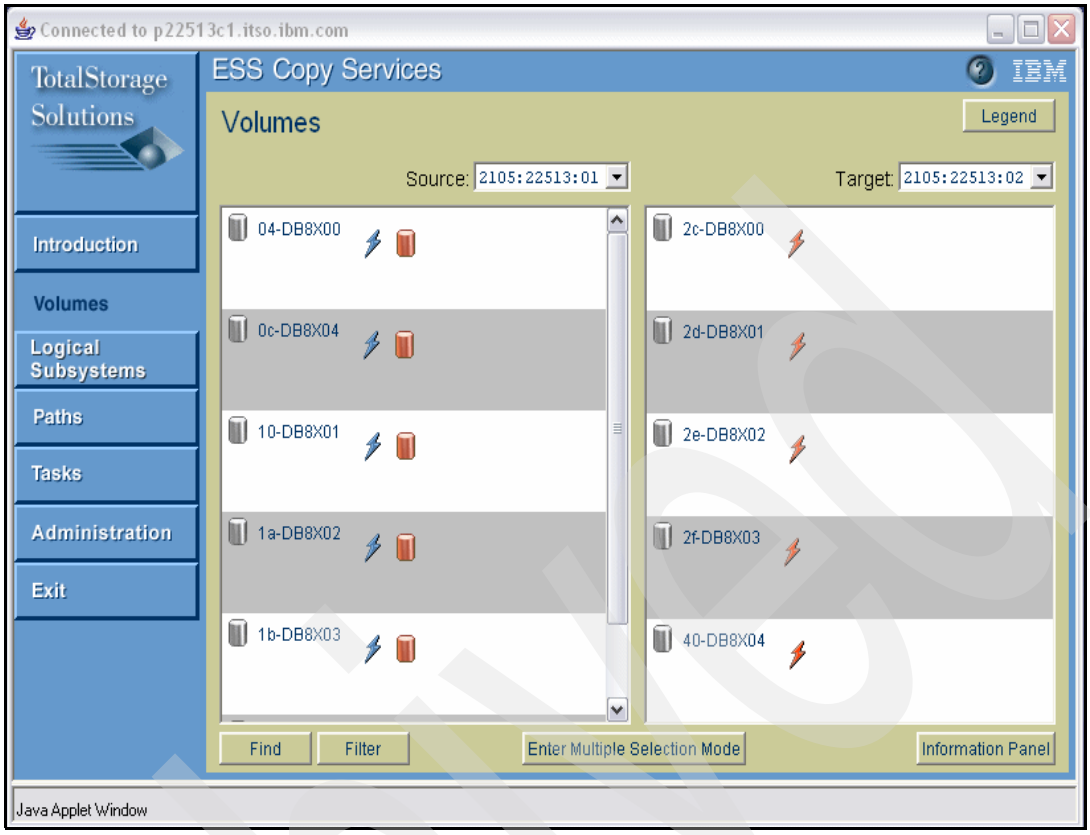


Figure 17-16 FlashCopy Secondary to Third volumes

17.5.6 PPRC Consistency Group Created task

Run Task No.9 'PP_CG_created' to stop the extended long busy state of the PPRC target volumes. This task will allow writes to the primary volumes to resume. Unless you invoke this task, you have to wait until the time-outs expire (the default is 2 minutes). After the task is completed, the DB2 workload will resume I/O activity as shown in Example 17-25, and the workload should finish successfully.

Example 17-25 'DA OJOB' after PPRC Consistency Group Created

```
Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC64  SC64     PAG    0 SIO    47 CPU   9/  9  LINE 1-2 (2)
COMMAND INPUT ===>                                          SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     PAOLOR51 PH01S19  DSNUPROC JOB13577 PAOLOR5  A NS  F9  17T   0.00   9.90
     PAOLOR53 PH01S13  DSNUPROC JOB13578 PAOLOR5  A NS  F9  948   0.28   6.33
```

17.5.7 Reestablish PPRC path

Since you ran the Freeze command, the path between PPRC primary and secondary LSSs have been suspended. To reestablish the PPRC path, run Task No.2 'PP_patcg_00_01'. Make sure that in the task panel, the task state is displayed as *successful*.

17.5.8 Resynchronize volume pairs

1. Run Task No. 11 'PP_pairs_resync'. This task resynchronizes the PPRC copy pairs. When you resynchronize copy pairs, all changed cylinders on the source volumes are copied to the target volumes. PPRC synchronous operations are then resumed for the pairs.
   – While the resynchronize task is going on, you can see the status of volume pairs as shown in Figure 17-17.

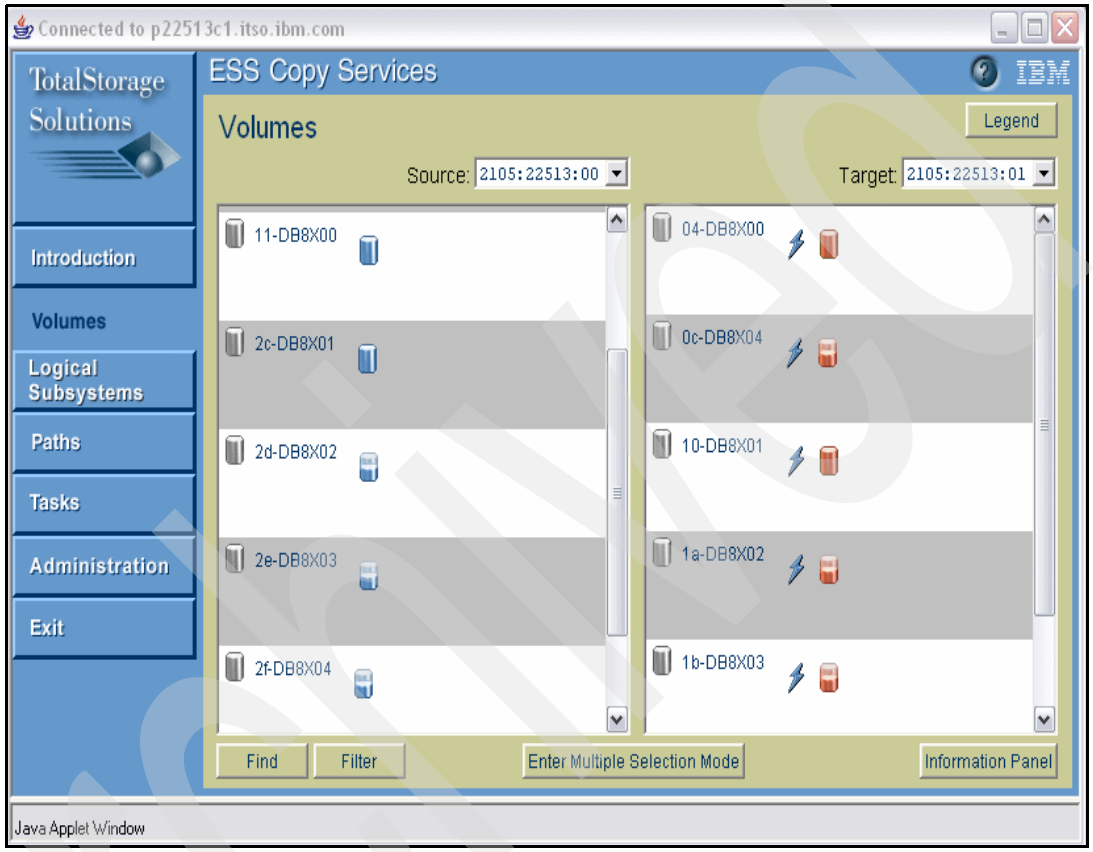


Figure 17-17 PPRC Resync volume pairs while the resynchronization is going on

   – While resynchronization is taking place, you can find messages about pending status of the PPRC source volumes in MVS console. See Example 17-26.

Example 17-26 During resynchronization of volumes in MVSLOG

```
IEA494I 8011,DB8X00,PPRC PAIR PENDING,SSID=8910,CCA=11
IEA494I 802F,DB8X04,PPRC PAIR PENDING,SSID=8910,CCA=2F
IEA494I 802C,DB8X01,PPRC PAIR PENDING,SSID=8910,CCA=2C
IEA494I 802D,DB8X02,PPRC PAIR PENDING,SSID=8910,CCA=2D
IEA494I 802E,DB8X03,PPRC PAIR PENDING,SSID=8910,CCA=2E
```

2. Make sure the Resync task is completed and all PPRC volume pairs are synchronized in the ESS Web Interface. See Figure 17-13 on page 311. Also, you can find messages in the MVS console log as shown in Example 17-27.

Example 17-27 After resynchronization completed in MVSLOG

```
IEA494I 8011,DB8X00,PPRC PAIR FULL DUPLEX,SSID=8910,CCA=11
IEA494I 802F,DB8X04,PPRC PAIR FULL DUPLEX,SSID=8910,CCA=2F
IEA494I 802C,DB8X01,PPRC PAIR FULL DUPLEX,SSID=8910,CCA=2C
IEA494I 802D,DB8X02,PPRC PAIR FULL DUPLEX,SSID=8910,CCA=2D
IEA494I 802E,DB8X03,PPRC PAIR FULL DUPLEX,SSID=8910,CCA=2E
```

17.5.9 Stop the DB2 subsystem

After the DB2 workload finishes, stop the DB2 Subsystem. See Example 17-4 on page 313.

17.5.10 Stop the PPRC pair

Run Task No. 6 'PP_stop_pairs_p'. This task terminates a PPRC pair and the PPRC relationship between source and target volume ends.

17.5.11 Terminate the PPRC path

Run Task No.4 'PP_rmpath_00_01' to remove PPRC path without force option. Specify the 'Force' option if you want to delete the path even when there are PPRC pair volumes.

17.5.12 Vary offline volumes of the primary site

Vary offline the PPRC source volumes. See Example 17-5 on page 314.

Note: If the user catalog of DB2 is located in any of these volumes, you have to unallocate the ICF catalog before offline the volumes in advance.

17.5.13 Vary online volumes of recovery site

Vary online the FlashCopy target volumes. See Example 17-6 on page 314.

17.5.14 Restart DB2 at the recovery site

Restart the DB2 subsystem from the FlashCopy target volumes. We assume that the target volumes are at the remote site, and restart the subsystem at the remote recovery site. When restarting the subsystem, there are one in-commit transaction and one inflight transaction which DB2 resolves automatically by applying log forward and backward. The DB2 restart log in our test is shown in Example 17-28.

Example 17-28 DB2 restart log

```
1                J E S 2  J O B  L O G  --  S Y S T E M  S C 6 4  --  N O D E  W T S C P
L X 2
0
 17.37.54 STC13614 ---- MONDAY,    12 JUL 2004 ----
 17.37.54 STC13614  IEF695I START DB8XMSTR WITH JOBNAME DB8XMSTR IS ASSIGNED TO USER STC
, GROUP SYS1
 17.37.54 STC13614  $HASP373 DB8XMSTR STARTED
 17.37.54 STC13614  IEF403I DB8XMSTR - STARTED - TIME=17.37.54 - ASID=0453 - SC64
 17.37.54 STC13614  DSNZ002I -DB8X DSNZINIT SUBSYSTEM DB8X SYSTEM PARAMETERS LOAD MODULE
NAME IS DSNZPARM
 17.37.54 STC13614  S DB8XIRLM
 17.37.55 STC13614  DSNY001I  -DB8X SUBSYSTEM STARTING
```

```
17.37.56 STC13614  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  130
   130             IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
17.37.56 STC13614  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  131
   131             IEC161I DB8XU.BSDS01.INDEX,UCAT.DB8XL
17.37.56 STC13614  IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS1,,,DB8XU.BSDS01,  132
   132             IEC161I DB8XU.BSDS01.DATA,UCAT.DB8XL
17.37.56 STC13614  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  133
   133             IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
17.37.56 STC13614  IEC161I 056-084,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  134
   134             IEC161I DB8XU.BSDS02.INDEX,UCAT.DB8XL
17.37.56 STC13614  IEC161I 062-086,DB8XMSTR,DB8XMSTR,BSDS2,,,DB8XU.BSDS02,  135
   135             IEC161I DB8XU.BSDS02.DATA,UCAT.DB8XL
17.37.57 STC13614  DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.194 17:23:48.06
17.37.57 STC13614  DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG  137
   137             DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS02,
   137             STARTRBA=00001B6C0000,ENDRBA=00001D87FFFF
17.37.57 STC13614  DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG  138
   138             DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS02,
   138             STARTRBA=00001B6C0000,ENDRBA=00001D87FFFF
17.37.57 STC13614  DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH  139
   139             STARTRBA=00001BAA5000
17.37.57 STC13614  S DB8XDBM1
17.38.00 STC13614  S DB8XDIST
17.38.02 STC13614  DSNR001I  -DB8X RESTART INITIATED
17.38.02 STC13614  DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=00001B6C34F1
17.38.02 STC13614  DSNR004I  -DB8X RESTART...UR STATUS COUNTS  152
   152             IN COMMIT=1, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
17.38.02 STC13614  DSNR007I  -DB8X RESTART...STATUS TABLE  153
   153             T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
   153             - -------- ------------ -------- -------- - ------------ --- --------
   153             B UTILITY  PAOLOR51     PAOLOR5  DSNUTIL  F 00001BA87000 194 17:23:50
   153             B UTILITY  PAOLOR53     PAOLOR5  DSNUTIL  C 00001BA950C4 194 17:23:51
17.38.07 STC13614  DSNI001I  -DB8X RESTART HAS BEEN DEFERRED  155
   155                        REASON 00C90095
   155                        TYPE 00000200
   155                        NAME DRTESTDB.DSN8S81D
17.38.07 STC13614  DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  156
   156             THE LOGICAL PAGE LIST
   156                        DATABASE NAME=DRTESTDB
   156                        SPACE NAME=DSN8S81D
   156                        DATA SET NUMBER=1
   156                        PAGE RANGE X'00000000' TO X'00000000'
   156                        START LRSN=X'00001BA87090'
   156                        END LRSN=X'00001BAA44F5'
   156                        START RBA=X'00001BA87090'
   156                        LPL TRACE ID=00000003
   156                        LPL REASON TYPE=LOGAPPLY
   156             …
17.38.07 STC13614  DSNB250E  -DB8X DSNIIMPD A PAGE RANGE WAS ADDED TO  157
   157             THE LOGICAL PAGE LIST
   157                        DATABASE NAME=DRTESTDB
   157                        SPACE NAME=DSN8S81D
   157                        DATA SET NUMBER=1
   157                        PAGE RANGE X'00000001' TO X'00000001'
   157                        START LRSN=X'00001BA87C21'
   157                        END LRSN=X'00001BAA44F5'
   157                        START RBA=X'00001BA87C21'
   157                        LPL TRACE ID=00000003
   157                        LPL REASON TYPE=LOGAPPLY
   157             …
```

```
17.38.07 STC13614  DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD  158
   158             RECOVERY
   158             IN COMMIT=0, INDOUBT=0
17.38.07 STC13614  DSNR018I  -DB8X RESTART...BACKWARD RECOVERY PROCESSED  159
   159              FROM RBA 00001BA88C9A TO RBA 00001BA87000
17.38.07 STC13614  DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD  160
   160             RECOVERY
   160             INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
17.38.08 STC13614  DSNR002I  -DB8X RESTART COMPLETED
17.38.08 STC13614  -DB8XRECOVER POSTPONED
17.38.08 STC13614  DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
17.38.08 STC13614  DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
17.38.08 STC13614  DSNL003I  -DB8X DDF IS STARTING
17.38.10 STC13614  DSNL519I  -DB8X DSNLILNR TCP/IP SERVICES AVAILABLE  166
   166                        FOR DOMAIN wtsc64.itso.ibm.com AND PORT 38080
17.38.10 STC13614  DSNL004I  -DB8X DDF START COMPLETE  167
   167                        LOCATION  DB8X
   167                        LU        USIBMSC.SCPDB8X
   167                        GENERICLU -NONE
   167                        DOMAIN    wtsc64.itso.ibm.com
   167                        TCPPORT   38080
   167                        RESPORT   38081
17.38.10 STC13614  DSN9022I  -DB8X DSNYASCP 'STA DB2' NORMAL COMPLETION
```

17.5.15 Check the DB2 objects in restricted status

1. Check for the Utility status with ‘Display Utility”. See Example 17-29. In this case you can see that one job was stopped during Reorg utility’s reload phase, the other one was stopped during the Load utility reload phase.

Attention: The active utilities and restricted status may vary depending on the active workload suspended during FlashCopy. The following output also can be different dependent on timing and environment.

Example 17-29 Display utility

```
-DIS UTIL(*)
 DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR5
                 MEMBER =
                 UTILID = DSNTEX
                 PROCESSING UTILITY STATEMENT 1
                 UTILITY = REORG
                 PHASE = RELOAD   COUNT = 0
                 NUMBER OF OBJECTS IN LIST = 1
                 LAST OBJECT STARTED = 1
                 STATUS = STOPPED
 DSNU100I  -DB8X DSNUGDIS - USERID = PAOLOR5
                 MEMBER =
                 UTILID = DSNTEY
                 PROCESSING UTILITY STATEMENT 5
                 UTILITY = LOAD
                 PHASE = RELOAD   COUNT = 77
                 NUMBER OF OBJECTS IN LIST = 1
                 LAST OBJECT STARTED = 1
                 STATUS = STOPPED
 DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
 ***
```

2. Check for DB objects in restricted status with **DISPLAY DATABASE,** as shown in Example 17-30. It shows that there are one recover pending table space and three rebuild pending indexes in one database. There are one check-pending table space, one recover-and-check pending table space, and one rebuild pending index in the other database.

Example 17-30 Display database

```
DSNT360I  -DB8X ***********************************
 DSNT361I  -DB8X *  DISPLAY DATABASE SUMMARY
                 *    RESTRICTED
 DSNT360I  -DB8X ***********************************
 DSNT362I  -DB8X     DATABASE = DRTESTDB  STATUS = RW
                    DBD LENGTH = 24218
 DSNT397I  -DB8X
 NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
 -------- ---- ----- ----------------- -------- -------- -------- -----
 DSN8S81D TS         RW,RECP,LPL,UTUT
 XDEPT1   IX         RW,RBDP,UTUT
 XDEPT2   IX         RW,RBDP,UTUT
 XDEPT3   IX         RW,RBDP,UTUT
 ******* DISPLAY OF DATABASE DRTESTDB ENDED      *********************
 DSNT360I  -DB8X ***********************************
 DSNT362I  -DB8X     DATABASE = DRTESTD2  STATUS = RW
                    DBD LENGTH = 24218
 DSNT397I  -DB8X
 NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
 -------- ---- ----- ----------------- -------- -------- -------- -----
 DSN8S81D TS         RW,CHKP
 DSN8S81P TS         RW,RECP,CHKP,UTUT
 XPROJAC1 IX         RW,RBDP,UTUT
 ***
```

17.5.16 Procedures to recover objects in restricted status

For details on how to reset the restricted status, refer to 17.3.13, “Procedures to recover objects from restricted status" on page 317.

17.6 Considerations

We have tested the possibilities for obtaining backup images of DB2 subsystems using PPRC and FlashCopy and shown how to restart from the backup at the remote site. The main difference among the three scenarios is on how to create consistency of backup data which will be used for recovery. Test Scenario 2 only uses the DB2 command **SET LOG SUSPEND** to create data consistency, and the other two test cases apply the techniques of the storage system provided by FlashCopy V2 and PPRC V2.

- Test Scenario 1 uses FlashCopy Consistency Group.

  We establish PPRC without the consistency group option and use FlashCopy with Consistency Group from secondary to third volumes and then restart from the FlashCopy target. In this scenario, additional volume pairs for FlashCopy are needed, so this may be the most costly alternative. From the DB2 point of view there is no problem about data consistency, but it takes longer to restart than using Test Scenario 2. This is because DB2 will search the most recent system checkpoint record and will have to apply all the logs from the point where restarted.

- Test Scenario 2 uses **`Set Log Suspend`** before the FlashCopy to the third set of volumes.

  Depending on the your environment, you can substitute FlashCopy with DFSMSdfs dump to tape. When DB2 Set Log Suspend is issued, DB2 suspends logging and update activity and creates a system checkpoint. It means that DB2 can use the checkpoint created just before the backup when restarting at the remote site, so recovery time should be quick.

- Test Scenario 3 applies PPRC Consistency Group and issues the Freeze command to suspend all I/O of the PPRC pairs before making copy for recovery.

  PPRC with Consistency Group option maintains consistency between PPRC source and target pairs. Considering the possibility of rolling disaster, it is strongly recommended to apply at least this option. The **`Freeze`** command stops the write activity on all active PPRC primary and secondary volumes under the frozen pairs. With the Consistency Groups created and the extended long busy window active, **`the Freeze`** command can be issued by an automation routine when disaster happens. It is suitable for creating point-in-time copy of the data, but the **`Freeze`** command is issued individually on each LSSs and this will not prevent a rolling disaster because of the slight delay between each command. If PPRC is used in conjunction with GDPS or eRCMF, it can ensure global consistency of data across all LSSs of the secondary volumes.

18

XRC and restart

This chapter describes the tests that we conducted at the IBM Silicon Valley Laboratory utilizing the XRC function for mirroring a large sysplex DB2 environment.

The following publications should be referenced to complement the information presented in this chapter:

- *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413
- *DB2 UDB for z/OS Version 8 Data Sharing: Planning and Administration,* SC18-7417
- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *DFSMS Extended Remote Copy Installation Planning Guide*, GC35-0481
- *DFSMS Extended Remote Copy Reference Information for Advanced Users*, GC35-0482

This chapter covers the following topics:

- Description of the test environment
- Preparing for using XRC
- Scenario 1: Simple DB2 restart
- Scenario 2: Disaster recovery
- Scenario 3: Using XRC to get PIT copies

18.1 Description of the test environment

The tests were conducted in a *four-way data sharing environment*, using the following hardware and software:

- Hardware:
  - zSeries 990 processor Model D32:
    - 1 LPAR with 16 shared CPs and 32 GB real memory
    - 1 LPAR with 16 shared CPs and 32 GB real memory
    - 3 Internal Coupling Facilities – CFCC release 13
  - zSeries 900 processor Model 116:
    - 1 LPAR with 12 shared CPs and 14 GB real memory, where the SDM will run
    - 1 LPAR with 12 shared CPs and 13 GB real memory
- Disks:
  - ESS Model 800 with 64 GB cache and FICON channels – XRC:
    - 192 volumes (3390-9) defined as primary volumes
    - 192 volumes (3390-9) defined as secondary volumes
- Software:
  - z/OS V1R4
  - DB2 for z/OS V8 New Function Mode

Figure 18-1 illustrates the XRC test environment. For the purpose of our tests, we are mirroring only the volumes containing DB2-related data, including DB2 catalog and directory, logs and BSDS for all members, and user databases. All these volumes are SMS managed, and all the data sets that they contain are cataloged in one ICF catalog, which is also defined in the same pool of volumes.

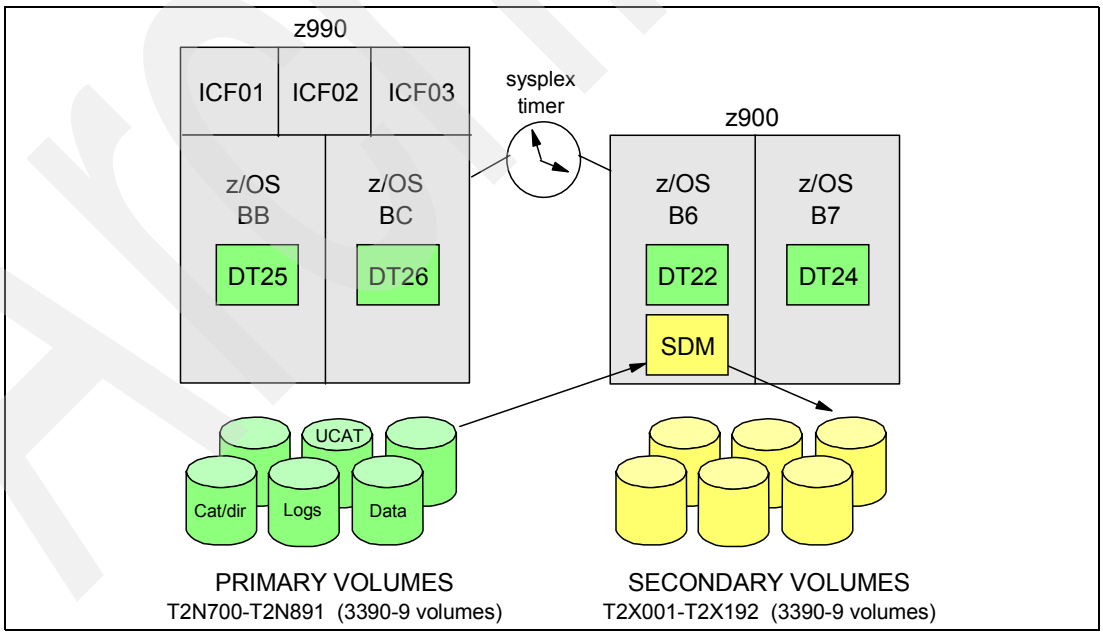


Figure 18-1 XRC test environment

Note: In our configuration, the primary and secondary volumes are both located on the same ESS. This may not reflect a “real-life” situation, but it is sufficient for testing the solution.

18.2 Preparing for using XRC

First of all, we must choose a *session_id* for the XRC session that we are going to define: We decide to use *SVL*. Based on this name, we can prepare the XRC environment as described in 13.5, “Implement the solution” on page 221.

18.2.1 Allocating the XRC data sets

We need to allocate the journal, control, and state data sets for the session SVL. We decide to use a specific high-level qualifier for these objects: XRC, instead of the default HLQ (SYS1).

Journal data sets

Since our configuration is relatively small, we define only four journal data sets named XRC.XCOPY.SVL.JRNL*nn*, with *nn* in (01,02,03,04). Each journal data set is a fixed blocked sequential data set with the attributes shown in Figure 18-2.

```
                              Allocate New Data Set
Command ===>
                                                                  More:     +
Data Set Name  . . . : XRC.XCOPY.SVL.JRNL01

Management class . . .                (Blank for default management class)
Storage class  . . . .                (Blank for default storage class)
 Volume serial . . . . JRNL01         (Blank for system default volume) **
 Device type . . . . .                (Generic unit or device address) **
Data class . . . . . .                (Blank for default data class)
 Space units . . . . . CYLINDER       (BLKS, TRKS, CYLS, KB, MB, BYTES
                                       or RECORDS)
 Average record unit                  (M, K, or U)
 Primary quantity  . . 350            (In above units)
 Secondary quantity   0               (In above units)
 Directory blocks  . . 0              (Zero for sequential data set) *
 Record format . . . . FB
 Record length . . . . 7680
 Block size  . . . . . 7680
 Data set name type  :                (LIBRARY, HFS, PDS, or blank)  *
                                      (YY/MM/DD, YYYY/MM/DD
```

Figure 18-2 XRC journal data set definitions

The size of the journal data sets depends on the number of XRC storage control sessions. In our configuration, we have 16 LSS in the ESS, hence 16 XRC storage control sessions. The SDM will allocate 16 x 576 buffers = 9216 full track buffers (equivalent to 615 cylinders). Therefore, the aggregate, minimum size for the journal data sets should be 1230 cylinders (twice the size of the SDM buffer). As we have four journals, the minimum size for each data set should be 307 cylinders.

Tip: For maximum performance, it is recommended to define the journal data sets as striped data sets, although we have chosen not to do so in our configuration.

Control data set

We define a control data set named XRC.XCOPY.SVL.CONTROL as a fixed blocked sequential data set with the attributes shown in Figure 18-3.

```
                              Allocate New Data Set
Command ===>
                                                                        More:     +
Data Set Name  . . . : XRC.XCOPY.SVL.CONTROL

Management class . . .                 (Blank for default management class)
Storage class  . . . .                 (Blank for default storage class)
 Volume serial . . . . GEN089          (Blank for system default volume) **
 Device type . . . . .                 (Generic unit or device address) **
Data class . . . . . .                 (Blank for default data class)
 Space units . . . . . TRACK           (BLKS, TRKS, CYLS, KB, MB, BYTES
                                        or RECORDS)
 Average record unit                   (M, K, or U)
 Primary quantity  . . 2               (In above units)
 Secondary quantity    0               (In above units)
 Directory blocks  . . 0               (Zero for sequential data set) *
 Record format . . . . FB
 Record length . . . . 15360
 Block size  . . . . . 15360
 Data set name type  :                 (LIBRARY, HFS, PDS, or blank)  *
                                       (YY/MM/DD, YYYY/MM/DD
```

Figure 18-3 XRC control data set definitions

State data set

Finally, we define a state data set named XRC.XCOPY.SVL.STATE as a PDSE with the attributes shown in Figure 18-4.

```
                              Allocate New Data Set
Command ===>
                                                                        More:     +
Data Set Name  . . . : XRC.XCOPY.SVL.STATE

Management class . . .                 (Blank for default management class)
Storage class  . . . .                 (Blank for default storage class)
 Volume serial . . . . GEN034          (Blank for system default volume) **
 Device type . . . . .                 (Generic unit or device address) **
Data class . . . . . .                 (Blank for default data class)
 Space units . . . . . TRACK           (BLKS, TRKS, CYLS, KB, MB, BYTES
                                        or RECORDS)
 Average record unit                   (M, K, or U)
 Primary quantity  . . 1000            (In above units)
 Secondary quantity    100             (In above units)
 Directory blocks  . . 0               (Zero for sequential data set) *
 Record format . . . . FB
 Record length . . . . 4096
 Block size  . . . . . 4096
 Data set name type  : LIBRARY         (LIBRARY, HFS, PDS, or blank)  *
                                       (YY/MM/DD, YYYY/MM/DD
```

Figure 18-4 XRC state data set definitions

The size of the state data sets depends on the number of XRC storage control sessions, the number of volume pairs, and the type of volumes. In our case, the recommended minimum allocation is:

```
16 (storage control sessions) x 10 tracks + 192 (pairs) x 3 tracks (3390-9) = 736 tracks
```

18.2.2 Establishing the XRC session environment

Now that the XRC data sets have been defined, we can start the XRC session SVL, using the XSTART command as shown in Example 18-1.

Example 18-1 XSTART command

```
XSTART SVL HLQ(XRC) SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
```

We use the following parameters:

- SVL HLQ(XRC) – Session_id and high-level qualifier for the XRC data sets.
- SESSIONTYPE(XRC) – XRC operates in recovery mode. In this mode XRC makes use of the journal and control data sets to guarantee secondary volume consistency. You must use this mode for disaster recovery solutions.
- ERRORLEVEL(SESSION) – If a volume encounters an error, all volumes in the XRC session will be suspended. You must correct the error and resynchronize all the suspended pairs.

Example 18-2 shows some of the messages that are sent to the system log. The XSTART command creates the address space ANTAS001 (1), which will manage the movement of data from the primary to the secondary volumes for the session SVL. ANTAS001 allocates the XRC data sets. It knows the name of the files (based on the naming convention), but does not know in advance the number of journal data sets. This is why it tries to allocate them one after the other, using the number sequence 01,02, etc., until it receives a message (IKJ56228I) for the first data set that does not exist (2).

A completion message (ANTS8100I) indicates that the session is active (3).

Example 18-2 XSTART command – Messages in syslog

```
ANTL8800I XSTART SVL HLQ(XRC) SESSIONTYPE(XRC) ERRORLEVEL(SESSION)
IEF196I         1 //IEESYSAS JOB TIME=1440,
IEF196I           // MSGLEVEL=1                                        (1)
IEF196I         2 //ANTAS001 EXEC IEESYSAS,PROG=ANTXAINI
...
IEF196I IKJ56228I DATA SET XRC.XCOPY.SVL.JRNL05 NOT IN CATALOG OR
IEF196I CATALOG CAN NOT BE ACCESSED                                   (2)
IKJ56228I DATA SET XRC.XCOPY.SVL.JRNL05 NOT IN CATALOG OR CATALOG CAN
NOT BE ACCESSED
...
ANTS8100I XSTART COMPLETE FOR XRC SESSION(SVL) WITH SESSIONTYPE(XRC),  (3)
ERRORLEVEL(SESSION), HLQ(XRC).  SESSION NOW ACTIVE
```

Once the XRC session is active, we can define the volume pairs for that session, using the XADDPAIR command as shown in Example 18-3.

Example 18-3 XADDPAIR commands

```
XADDPAIR SVL VOL(T2N700,T2X001) DONOTBLOCK
XADDPAIR SVL VOL(T2N701,T2X002) DONOTBLOCK
...
XADDPAIR SVL VOL(T2N890,T2X191) DONOTBLOCK
XADDPAIR SVL VOL(T2N891,T2X192) DONOTBLOCK
```

We use the following parameter:

- DONOTBLOCK – If the SDM is unable to drain updates from the ESS cache rapidly enough, then the ESS microcode will slow down write activity from the primary systems to the primary volumes with high write I/O activity.

Example 18-4 shows some of the messages that are sent to the system log. When an XADDPAIR command is issued, XRC makes an initial copy of the primary volume to the secondary volume. First, XRC varies the secondary volume offline to prevent any access while the copy is in progress. A completion message (ANTI8024I) indicates that the duplexing has been established, and XRC varies the secondary volume online.

Example 18-4 XSTART command – Messages in syslog

```
ANTL8800I XADDPAIR SVL VOL(T2N700,T2X001) DONOTBLOCK
...
ANTA8101I XADDPAIR COMPLETE FOR VOLUME PAIR(T2N700,T2X001) FOR
SESSION(SVL) WITH ERRORLEVEL(SESSION)
...
VARY 600E,OFFLINE
...
ANTI8023I FULL INITIALIZATION STARTED FOR XRC VOLUME PAIR(T2N700,T2X001)
...
ANTI8024I XRC VOLUME PAIR(T2N700,T2X001) IN DUPLEX STATUS AT TIME OF
2004.208 15:21:38.476530
...
VARY 600E,ONLINE
```

While the XRC environment is being established, we use the XQUERY command shown in Example 18-5 to request a status at the session level. We send the output of the command to a pre-defined sequential data set.

Example 18-5 XQUERY command – Session level

```
XQUERY SVL DATASET('XRC.XQUERY.OUTPUT') DISP(MOD)
```

Example 18-6 shows the output of the XQUERY command, at the very beginning of the copy process. In the XQUERY summary report, we can see that XRC has registered 79 pairs so far, 8 primary volumes are being copied (status CPY), and 71 are waiting to be copied (status PND). The number of volume initialization tasks is controlled by XRC tuning parameters.

Example 18-6 XQUERY command – Session level – Summary report

```
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY SUMMARY REPORT - 003
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=79  DUP=0  CPY=8  PND=71  SUS=0  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.208 15:21:40.055150)   IDLE(00:03:53.4)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SVL)
```

We can also use the XQUERY command as shown in Example 18-7 to request a status at the volume level. We send the output of the command to our pre-defined sequential data set.

Example 18-7 XQUERY command – Volume level

```
XQUERY SVL VOLUME(ALL) DATASET('XRC.XQUERY.OUTPUT') DISP(MOD)
```

Example 18-8 shows the output of the XQUERY command. In the XQUERY volume report, we can see that XRC has now registered the 192 pairs. 63 primary volumes have been copied to the secondary volumes (status DUP), 16 are being copied (status CPY), and 113 are in pending status. For each volume that is being processed, the report indicates the percentage that has been copied so far.

Example 18-8 XQUERY command – Volume level – Volume report

```
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY VOLUME REPORT - 003
ANTQ8211I PRIM   SEC     ERROR  SYNCH
ANTQ8212I VOL    VOL     LEVEL    % STA CMD OP -------TIMESTAMP--------
ANTQ8203I ------------------------------------------------------------
ANTQ8213I T2N700 T2X001 SESSION     DUP
...
ANTQ8213I T2N709 T2X010 SESSION  00 PND
ANTQ8213I T2N710 T2X011 SESSION  00 PND
ANTQ8213I T2N711 T2X012 SESSION     DUP
ANTQ8213I T2N712 T2X013 SESSION  43 CPY
ANTQ8213I T2N713 T2X014 SESSION  13 CPY
...
ANTQ8238I TOTAL=192  DUP=63  CPY=16  PND=113  SUS=0  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.208 15:24:12.433469)  DELAY(00:00:15.68)
ANTQ8240I DATA EXPOSURE(00:00:05.25)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY VOLUME REPORT COMPLETE FOR SESSION(SVL)
```

The establishment of the XRC environment may generate a significant load on the system where the SDM is running. Figure 18-5 is a sample snapshot of the activity on system B6, printed at a time when an I/O intensive workload and the XRC copy were running concurrently. Notice the number of I/O per second on the system and the number of I/O per second generated by the SDM.

```
SDSF DA B6    STLABB6  PAG    0 SIO 73433 CPU 100/ 61  LINE 1-3 (3)
COMMAND INPUT ===>                                          SCROLL ==
NP   JOBNAME  REAL PAGING    SIO   CPU% ASID ASIDX  EXCP-CNT   CPU-TIME
     ANTAS001 158T   0.00 34238    3.08  148  0094  51M        14956.10
     ANTMAIN  1826   0.00   0.00   0.00   11  000B       448        9.28
     ANTAS000 1043   0.00   0.00   0.00   12  000C       195        0.19
```

Figure 18-5 Establishment of the XRC environment – System load

Example 18-9 shows the message sent to the system log when the session is established. In our basic configuration, without any major activity on the primary volumes, it took 5-6 hours to copy the 192 primary volumes (3390-9 disks) to the secondary volumes (around 1.5 TB). With a workload running on the primary system and primary disks, it took up to 10 hours.

Note: Keep in mind that we implemented a basic solution (such as not using striping for the journal data sets) and we did not spend any time tuning the XRC environment.

Example 18-9 Establishment of the XRC environment – Completion message

```
ANTX8120I 192 VOLUMES IN SESSION(SVL) ARE NOW DUPLEX
```

Example 18-10 shows the XQUERY summary report taken after the XRC environment has been established. All the volumes have been copied (DUP=192).

Example 18-10 XQUERY command – Session level – Summary report

```
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY SUMMARY REPORT - 003
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=192  DUP=192  CPY=0  PND=0  SUS=0  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.208 21:04:18.838465)  DELAY(00:00:03.02)
ANTQ8240I DATA EXPOSURE(00:00:03.02)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SVL)
```

The *data consistent field* gives the time of the last update that was secured on the journal data sets. The timestamp variable specifies the UTC time from the primary system (generated based on the updates received from the primary system). The data mover can bring data on the secondary volumes consistent up to that timestamp.

XQUERY also reports a *delay time*. This is the current delay between the last application I/O to a primary storage control and the update made by the data mover to a secondary target volume. This field provides an approximation of the time difference between the primary and secondary volumes.

The *data exposure field* gives an approximation of the time difference (potentially non-recoverable data) between data written to the primary volumes and data secured on the journal data set.

18.3 Scenario 1: Simple DB2 restart

In this scenario, we quiesce all DB2 members on the primary system. We use XRC commands to switch from the primary volumes to the secondary volumes. And we restart DB2 from the secondary volumes.

Note: In our test environment, the difference between UTC (used by XRC commands) and Pacific time (printed in the system log) is seven hours.

18.3.1 Preparation

The initial situation is as follows:

- Only two DB2 members are active: DT25 on system BB and DT26 on system BC.
- The XRC session is established – all 192 volumes are mirrored, as shown on the XQUERY summary report in Example 18-11.

Example 18-11 Scenario 1 – Initial situation

```
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY SUMMARY REPORT - 003
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=192  DUP=192  CPY=0  PND=0  SUS=0  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.208 22:31:07.742634)   IDLE(00:01:05.2)
ANTQ8240I DATA EXPOSURE(00:00:00.00)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SVL)
```

We quiesce both DB2 members. Example 18-12 is an extract of the system log, which shows the time when the -STO DB2 commands completed.

Example 18-12 Scenario 1 – Stop the DB2 members

```
-DT26STO DB2
-DT25STO DB2
...
15:34:21.46    DSN3104I -DT26 DSN3EC0X - TERMINATION COMPLETE
15:42:07.96    DSN3104I -DT25 DSN3EC0X - TERMINATION COMPLETE
```

18.3.2 Recovery on the secondary volumes

Now we want to switch to the secondary volumes.

End the XRC session

We use the XEND command as shown in Example 18-13 to stop all XRC activity on active volumes and terminate the XRC session. Issuing the XEND command stops all updates to the secondary volumes.

Example 18-13 Scenario 1 – XEND command

```
XEND SVL
```

Example 18-14 shows the messages that are sent to the system log. The completion message (ANTE8106I) returns a timestamp in the *consistency_group time* field: it indicates the time up to which the data on the secondary volumes are consistent with the corresponding data primary volumes.

During our test, it took 15 seconds for the XEND command to complete.

Example 18-14 Scenario 1 – XEND command – Messages in syslog

```
15:57:09.07    ANTL8800I XEND SVL
...
15:57:24.36    ANTE8106I XEND(IMMEDIATE) COMPLETE FOR SESSION(SVL) AT CONSISTENCY_GROUP
               TIME(2004.208 22:57:03.812661).  SESSION NOW INACTIVE
```

Vary all primary volumes offline to the SDM system

Now, we need to take all primary volumes offline to the SDM system. But before we can issue the VARY OFFLINE commands, we must unallocate the ICF catalog that resides on those volumes, as shown in Example 18-15.

In our specific configuration, since the SDM is part of a parallel sysplex, we need to route the commands to all the systems in the sysplex.

Example 18-15 Scenario 1 – Vary all primary volumes offline

```
RO *ALL,F CATALOG,UNALLOCATE(SYS1.ICFCAT.DSNT2)
RO *ALL,V 6000-600B,OFFLINE
```

Recover the secondary volumes

We use the XRECOVER command as shown in Example 18-16 to switch to the secondary volumes. We have to specify the name of the session that was previously specified on the XSTART command. This XRECOVER command will bring data on the secondary volumes to the last consistent, recoverable state where they are equivalent to the primary volumes.

Example 18-16 Scenario 1 – XRECOVER command

```
XRECOVER SVL HLQ(XRC)
```

Example 18-17 shows some of the messages that are sent to the system log.

The XRECOVER command brings up the ANTAS001 address space, which will manage the recovery (1) and reports the timestamp up to which all data on the secondary volumes can be consistent (2). The XRECOVER command processing applies the journaled data to update the secondary volumes up to that time, and reformats the labels of the secondary volumes to match the labels of their associated primary volumes.

When all updates have been applied to the secondary volumes, the XRECOVER command automatically generates an XQUERY recover report. The report includes the last applied timestamp that defines the recovered, consistent state for all volumes (3) – 15:57:03 in our case. By this time, all the DB2 members were quiesced (see Example 18-12).

The XRECOVER command finally brings the secondary volumes online (4), issues a completion message (ANTR8102I), and closes the ANTAS001 address space (5). During our test, it took 23 seconds for the XRECOVER command to complete.

Example 18-17 Scenario 1 – XRECOVER command – Messages in syslog

```
ANTL8800I XRECOVER SVL HLQ(XRC)
IEF196I         1 //IEESYSAS JOB TIME=1440,
IEF196I           // MSGLEVEL=1                                          (1)
IEF196I         2 //ANTAS001 EXEC IEESYSAS,PROG=ANTXAINI
...
ANTR8108I XRECOVER STARTED FOR SESSION(SVL) AT STARTING CONSISTENCY_GROU
P TIME(2004.208 22:57:03.812661), HLQ(XRC)                              (2)
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY RECOVER REPORT - 002
ANTQ8271I ------ORIGINAL-------
ANTQ8274I PRIMARY  SECONDARY STA CON CLP  ------TIMESTAMP---------
ANTQ8203I ------------------------------------------------------------
ANTQ8275I T2N865   T2X166    DUP YES YES
ANTQ8275I T2N866   T2X167    DUP YES YES
ANTQ8275I T2N867   T2X168    DUP YES YES
...
ANTQ8275I T2N710   T2X011    DUP YES YES
ANTQ8275I T2N876   T2X177    DUP YES YES
ANTQ8237I TOTAL=192   DUP=192   PND=0   SUS=0
ANTQ8231I DATA CONSISTENT(2004.208 22:57:03.812661)                     (3)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8201I XQUERY RECOVER REPORT COMPLETE FOR SESSION(SVL)
IEE302I 6031     ONLINE
IEE302I 6032     ONLINE                                                 (4)
IEE302I 6033     ONLINE
IEE302I 6034     ONLINE
...
ANTR8102I XRECOVER COMPLETE FOR SESSION(SVL) AT RECOVERED CONSISTENCY_GR
OUP TIME(2004.208 22:57:03.812661)
...                                                                     (5)
IEF196I IEF375I  JOB/ANTAS001/START 2004208.1557
IEF196I IEF376I  JOB/ANTAS001/STOP  2004208.1558 CPU    0MIN 23.41SEC
```

The XRECOVER command brings the secondary volumes online to the SDM system only. In our specific configuration, since the SDM is part of a parallel sysplex, we need to make them available to the other systems in the sysplex, as shown in Example 18-18.

Example 18-18 Scenario 1 – Vary all secondary volumes online

```
RO *ALL,V 6031-6034,ONLINE
```

18.3.3 Restart DB2

Finally, we restart DB2 from the secondary volumes.

Clean up the DB2 structures

Before we start DB2, we delete the structures belonging to the data sharing group DSNT2.

First, we enter the commands shown in Example 18-19 to force the connections off the group buffer pools and LOCK structures (connections for the SCA are not held at termination, so there are no SCA connections to force off).

Example 18-19 Scenario 1 – Clean up the DB2 structures (1/2)

```
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_GBP0
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_GBP1
...
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_LOCK1
```

Then, we enter the commands shown in Example 18-20 to delete all the structures.

Example 18-20 Scenario 1 – Clean up the DB2 structures (2/2)

```
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_GBP0
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_GBP1
...
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_LOCK1
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_SCA
```

Finally, we verify that the structures have been successfully deleted. Example 18-21 shows the status of the structures – all the structures belonging to the data sharing group DSNT2 must be in status ‘NOT ALLOCATED’.

Example 18-21 Scenario 1 – Clean up the structures – Verification

```
D XCF,STR
IXC359I  16.00.26  DISPLAY XCF 596
STRNAME            ALLOCATION TIME   STATUS
...
DSNT2_GBP0            --       --    NOT ALLOCATED
DSNT2_GBP1            --       --    NOT ALLOCATED
...
DSNT2_LOCK1           --       --    NOT ALLOCATED
DSNT2_SCA             --       --    NOT ALLOCATED
```

Start DB2

We can now start DB2. Since the structures have been deleted, DB2 must perform a *group restart*. Group restart is distinguished from normal restart by the activity of rebuilding the information that was lost from the SCA and lock structures. Group restart does not necessarily mean that all DB2 subsystems in the group start up again, but information from all non-starting DB2 subsystems must be used to rebuild the lock structure or SCA.

Tip: Although one DB2 can perform restart on behalf of the group, you should restart all of the non-quiesced members together, perhaps by using an automated procedure. This shortens the total restart time. Also, because retained locks are held for non-starting DB2 subsystems, it is best to start all members of the group for maximum data availability.

However, in order to show the messages issued during group restart, we choose to start only one DB2 member: DT22 on system B6. Note that, since all the DB2 members were quiesced at the time of the consistency group established by XRC, there is no risk of retained locks.

DT22 initiates the group restart, which requires scanning the logs of each member to rebuild the SCA and retained lock information, shown in Example 18-22.

After the DB2 initialization and the current status rebuild (CSR) phases (1), DT22 goes through a *peer CSR phase* for the other three members, and rebuilds the SCA (2). Then, after the forward recovery phase (3), DT22 goes through a *peer forward recovery phase*, which will rebuild the retained lock information (4). Finally, DT22 completes restart recovery processing with the backward recovery phase (5).

Since all members had been normally quiesced, the restart (even though it is a group restart) is performed quickly. During our test, DB2 came up in 75 seconds.

Example 18-22 Scenario 1 – Restart DB2

```
16:06:38  -DT22STA DB2
...
DSNY001I  -DT22 SUBSYSTEM STARTING
DSNJ127I  -DT22 SYSTEM TIMESTAMP FOR BSDS= 04.208 10:55:34.09
...
DSNR001I  -DT22 RESTART INITIATED
DSNR003I  -DT22 RESTART...PRIOR CHECKPOINT RBA=04015A13FBF8
DSNR004I  -DT22 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0          (1)
DSNR021I  -DT22 DSNRRGRC DB2 SUBSYSTEM MUST PERFORM
GROUP RESTART FOR PEER MEMBERS
DSNR023I  -DT22 DSNRRGRC GROUP RESTART INITIATED TO                        (2)
RECOVER THE SCA FOR GROUP MEMBER DT24
DSNR003I  -DT22 RESTART...PRIOR CHECKPOINT RBA=024599B51CBE
DSNR004I  -DT22 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR024I  -DT22 DSNRRGRC GROUP RESTART COMPLETED TO
RECOVER THE SCA FOR GROUP MEMBER DT24
...
DSNR022I  -DT22 DSNRRGRC DB2 SUBSYSTEM HAS COMPLETED
GROUP RESTART FOR PEER MEMBERS
DSNR005I  -DT22 RESTART...COUNTS AFTER FORWARD                             (3)
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR021I  -DT22 DSNRRGRH DB2 SUBSYSTEM MUST PERFORM
GROUP RESTART FOR PEER MEMBERS
DSNR025I  -DT22 DSNRRGRH GROUP RESTART INITIATED TO                        (4)
RECOVER RETAINED LOCKS FOR GROUP MEMBER DT24
DSNR003I  -DT22 RESTART...PRIOR CHECKPOINT RBA=024599B51CBE
DSNR004I  -DT22 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR005I  -DT22 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR026I  -DT22 DSNRRGRH GROUP RESTART COMPLETED TO
```

```
RECOVER RETAINED LOCKS FOR GROUP MEMBER DT24
...
DSNR022I  -DT22 DSNRRGRH DB2 SUBSYSTEM HAS COMPLETED
GROUP RESTART FOR PEER MEMBERS
DSNR006I  -DT22 RESTART...COUNTS AFTER BACKWARD                    (5)
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
...
16:07:53    DSN9022I  -DT22 DSNYASCP 'STA DB2' NORMAL COMPLETION
```

18.3.4 Summary

Here we summarize what we experienced during our test:

- XEND command: This took 15 seconds to complete.
- XRECOVER command: This took 23 seconds to complete.
- Delay between primary and secondary volumes: This took 5 seconds.
- DB2 came up in 75 seconds.

Note: Keep in mind that we implemented a basic solution (for example, we did not use striping for the journal data sets) and we did not spend any time tuning the XRC environment.

18.4 Scenario 2: Disaster recovery

In this scenario, we simulate a disaster by cancelling the SDM address space. We use XRC commands to switch from the primary volumes to the secondary volumes. And we restart DB2 from the secondary volumes.

Note: In our test environment, the difference between UTC (used by XRC commands) and Pacific time (printed in the system log) is seven hours.

18.4.1 Preparation

The initial situation is as follows:

- Only two DB2 members are active: DT22 on system B6 and DT24 on system B7.
- The XRC session is established – all 192 volumes are mirrored, as shown on the XQUERY summary report in Example 18-23.

Example 18-23 Scenario 2 – Initial situation

```
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY SUMMARY REPORT - 003
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=192  DUP=192  CPY=0  PND=0  SUS=0  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.209 22:03:04.753888)  DELAY(00:00:11.51)
ANTQ8240I DATA EXPOSURE(00:00:11.51)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SVL)
```

18.4.2 Disaster

While a heavy update workload is running on the two members (generating around 30,000 SIO), we simulate a disaster by cancelling the SDM address space (C ANTAS001), as shown in Example 18-24. The primary volumes get suspended at 15:13:43.73.

Example 18-24 Scenario 2 – Disaster

```
15:13:43.73 CHUN3    00000280  C ANTAS001
```

We let the workload terminate on B6 and B7 and we quiesce the DB2 members.

18.4.3 Recovery on the secondary volumes

Now we want to switch to the secondary volumes.

Vary all primary volumes offline to the SDM system

First, we unallocate the ICF catalog that resides on the primary volumes, and we take all primary volumes offline to the SDM system, as shown in Example 18-25.

In our specific configuration, since the SDM is part of a parallel sysplex, we need to route the commands to all the systems in the sysplex.

Example 18-25 Scenario 2 – Vary all primary volumes offline

```
RO *ALL,F CATALOG,UNALLOCATE(SYS1.ICFCAT.DSNT2)
RO *ALL,V 6000-600B,OFFLINE
```

Recover the secondary volumes

We use the XRECOVER command as shown in Example 18-26 to add all valid, non-applied journaled data to the secondary volumes. This XRECOVER command will bring data on the secondary volumes to the last consistent, recoverable state prior to the disaster.

Example 18-26 Scenario 2 – XRECOVER command

```
XRECOVER SVL HLQ(XRC)
```

Example 18-27 shows some of the messages that are sent to the system log. When all updates have been applied to the secondary volumes, the XRECOVER command automatically generates an XQUERY recover report. The report includes the last applied timestamp that defines the recovered, consistent state for all volumes – 15:13:28.84 in this case, that is less than 15 seconds before the disaster.

> **Note:** Keep in mind that we implemented a basic solution (for example, we did not use striping for the journal data sets) and we did not spend any time tuning the XRC environment.

During our test, it took 25 seconds for the XRECOVER command to complete.

Example 18-27 Scenario 2 – XRECOVER command – Messages in syslog

```
ANTL8800I XRECOVER SVL HLQ(XRC)
...
ANTR8108I XRECOVER STARTED FOR SESSION(SVL) AT STARTING CONSISTENCY_GROU
P TIME(2004.209 22:13:28.842861), HLQ(XRC)
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY RECOVER REPORT - 002
```

```
ANTQ8271I ------ORIGINAL-------
ANTQ8274I PRIMARY  SECONDARY STA CON CLP  ------TIMESTAMP---------
ANTQ8203I -------------------------------------------------------------
ANTQ8275I T2N865   T2X166    DUP YES YES
ANTQ8275I T2N866   T2X167    DUP YES YES
ANTQ8275I T2N867   T2X168    DUP YES YES
...
ANTQ8275I T2N876   T2X177    DUP YES YES
ANTQ8237I TOTAL=192   DUP=192   PND=0   SUS=0
ANTQ8231I DATA CONSISTENT(2004.209 22:13:28.842861)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8201I XQUERY RECOVER REPORT COMPLETE FOR SESSION(SVL)
IEE302I 6031     ONLINE
IEE302I 6032     ONLINE
IEE302I 6033     ONLINE
...
ANTR8102I XRECOVER COMPLETE FOR SESSION(SVL) AT RECOVERED CONSISTENCY_GR
OUP TIME(2004.209 22:13:28.842861)
...
IEF196I IEF373I STEP/IEFPROC /START 2004209.1533
IEF196I IEF374I STEP/IEFPROC /STOP  2004209.1534 CPU    0MIN 24.50SEC
```

The XRECOVER command brings the secondary volumes online to the SDM system only. In our specific configuration, since the SDM is part of a parallel sysplex, we need to make them available to the other systems in the sysplex, as shown in Example 18-28.

Example 18-28 Scenario 2 – Vary all secondary volumes online

```
RO *ALL,V 6031-6034,ONLINE
```

18.4.4 Restart DB2

Finally, we restart DB2 from the secondary volumes.

Clean up the DB2 structures

Before we start DB2, we delete the structures belonging to the data sharing group DSNT2.

First, we enter the commands shown in Example 18-29 to force the connections off the group buffer pools and LOCK structures (connections for the SCA are not held at termination, so there are no SCA connections to force off).

Example 18-29 Scenario 2 – Clean up the DB2 structures (1/2)

```
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_GBP0
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_GBP1
...
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_LOCK1
```

Then, we enter the commands shown in Example 18-30 to delete all the structures.

Example 18-30 Scenario 2 – Clean up the DB2 structures (2/2)

```
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_GBP0
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_GBP1
...
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_LOCK1
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_SCA
```

Finally, we verify that the structures have been successfully deleted. Example 18-31 shows the status of the structures – all the structures belonging to the data sharing group DSNT2 must be in status “NOT ALLOCATED”.

Example 18-31 Scenario 2 – Clean up the structures – Verification

```
D XCF,STR
IXC359I  15.37.48  DISPLAY XCF 596
STRNAME            ALLOCATION TIME   STATUS
...
DSNT2_GBP0            --       --    NOT ALLOCATED
DSNT2_GBP1            --       --    NOT ALLOCATED
...
DSNT2_LOCK1           --       --    NOT ALLOCATED
DSNT2_SCA             --       --    NOT ALLOCATED
```

Start DB2

We can now start DB2. Since the DB2 members DT22 and DT24 were active at the time of the consistency group established by XRC, we should bring up at least these members (in order to release the retained locks that they might hold).

Example 18-32 and Example 18-33 show the first messages printed by DT22 and DT24 at restart. The two members restart in the same way they would do after a disaster. Some inflight and inabort transactions must be backed out (6 on DT22, 11 on DT24).

Notice the *system timestamp for BSDS* value: 15:13:28.57 for DT22 and 15:13:28.72 for DT24. The time of XRC consistency group was 15:13:28.84.

Example 18-32 Scenario 2 – Restart DT22 (1/2)

```
-DT22STA DB2
15.38.42 STC37529  $HASP373 DT22MSTR STARTED
...
DSNJ127I  -DT22 SYSTEM TIMESTAMP FOR BSDS= 04.209 15:13:28.57
...
DSNR001I  -DT22 RESTART INITIATED
DSNR003I  -DT22 RESTART...PRIOR CHECKPOINT RBA=040253A27506
DSNR004I  -DT22 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=6, IN ABORT=0, POSTPONED ABORT=0
DSNR007I  -DT22 RESTART...STATUS TABLE
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
B BATCH    T2B01006     SOFPROD  CRWWB01  F 04025A99FA3E 209 15:13:26
B BATCH    T2B01010     SOFPROD  CRWWB01  F 04025AC140E4 209 15:13:26
B BATCH    T2B01002     SOFPROD  CRWWB01  F 04025AF71720 209 15:13:27
B BATCH    T2B01004     SOFPROD  CRWWB01  F 04025B076EB2 209 15:13:28
B BATCH    T2B01008     SOFPROD  CRWWB01  F 04025B291F24 209 15:13:28
I SERVER   t3cli        LGB0032  DISTSERV F 04020375D02F 209 14:27:22
...
```

Example 18-33 shows the first messages printed by DT24 at restart.

Example 18-33 Scenario 2 – Restart DT24 (1/2)

```
-DT24STA DB2
15.39.11 STC37560  $HASP373 DT24MSTR STARTED
...
DSNJ127I  -DT24 SYSTEM TIMESTAMP FOR BSDS= 04.209 15:13:28.72
...
```

```
DSNR001I  -DT24 RESTART INITIATED
DSNR003I  -DT24 RESTART...PRIOR CHECKPOINT RBA=0246CC361F0D
DSNR004I  -DT24 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=7, IN ABORT=4, POSTPONED ABORT=0
DSNR007I  -DT24 RESTART...STATUS TABLE  971
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
B BATCH    T2DDLA5      SUELI    DSNTEP3  A 0246CA58A190 209 15:00:25
B BATCH    T2DDLE5      SUELI    DSNTEP3  A 0246CA88D178 209 15:00:39
B BATCH    T2DDLE4      SUELI    DSNTEP3  A 0246CBB319AB 209 15:07:07
B BATCH    T2DDLA4      SUELI    DSNTEP3  A 0246CBC15BB5 209 15:07:29
B BATCH    T2DDLU3      SUELI    DSNTEP3  F 0246CBCBE000 209 15:07:31
B BATCH    T2DDLU5      SUELI    DSNTEP3  F 0246CC1E428B 209 15:11:11
B BATCH    T2B01005     SOFPROD  CRWWB01  F 0246D3FBC558 209 15:13:25
B BATCH    T2B01009     SOFPROD  CRWWB01  F 0246D458F8E8 209 15:13:27
B BATCH    T2B01003     SOFPROD  CRWWB01  F 0246D477D804 209 15:13:27
B BATCH    T2B01001     SOFPROD  CRWWB01  F 0246D4826474 209 15:13:27
B BATCH    T2B01007     SOFPROD  CRWWB01  F 0246D4AF2E40 209 15:13:28
...
```

During the group restart, all restarting DB2 subsystems update the lock structure with information contained on their logs. Example 18-34 shows the output of the MODIFY irlmproc,STATUS command issued shortly after the beginning of the group restart. The modify locks that DT22 and DT24 were holding at the time of the consistency group established by XRC are marked retained. Retained locks are necessary to protect data in the process of being updated from being accessed by another active DB2 member of the group. They will be released during restart.

Example 18-34 Scenario 2 – Retained locks

```
F DT22IRLM,STATUS,ALLD
DXR102I IT22002 STATUS 017
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS    RET_LKS   IRLMID  IRLM_NAME   IRLM_LEVL
        DT22     UP            235    002      IT22         2.023
        DT24     UP            483    004      IT24         2.023
    DXR102I End of display
```

The table spaces, indexes, or partitions that had changed pages in the group buffer pools at the time of the consistency group established by XRC are marked as group buffer pool recovery pending (GRECP), as shown in Example 18-35. The entire page set is unavailable.

DB2 can usually automatically recover GRECP page sets when the group buffer pool is defined with AUTOREC(YES) – but DB2 never initiates automatic recovery during restart.

Example 18-35 Scenario 2 – Group buffer pool recovery pending (GRECP)

```
DSNB323I  -DT24 DSNB1DA2 TABLE SPACE OR INDEX SPACE
WITH
          DBID X'0001' PSID X'00CF'
          IN GROUP BUFFERPOOL GBP0
          IS IN GROUP BUFFERPOOL RECOVERY PENDING STATE
          AND CANNOT BE AUTOMATICALLY RECOVERED
```

Specific pages (or page ranges) are added to the logical page list (LPL), as shown in Example 18-36. An LPL status condition does not make the entire page set or partition unavailable. Only those pages of the page set recorded in the LPL are unavailable.

As of V8, DB2 can attempt to automatically recover LPL page sets – but not in the case of a DB2 restart (message DSNB357I).

Example 18-36 Scenario 2 – Logical page list (LPL)

```
DSNI001I  -DT24 RESTART HAS BEEN DEFERRED
           REASON 00C20210
           TYPE 00000220
           NAME DSNT2SYS.DSNDBC.DSNDB06.SYSDBASE.I0001.A001
DSNB250E  -DT24 DSNIIMPD A PAGE RANGE WAS ADDED TO
THE LOGICAL PAGE LIST
           DATABASE NAME=DSNDB06
           SPACE NAME=SYSDBASE
           DATA SET NUMBER=1
           PAGE RANGE X'000054B0' TO X'000054B0'
           START LRSN=X'BB94A835196F'
           END LRSN=X'BB94B0F8CFF1'
           START RBA=X'0246CA5904BC'
           LPL TRACE ID=00000003
           LPL REASON TYPE=LOGAPPLY
DSNB357I  -DT24 DSNB1LPL AUTOMATIC LPL RECOVERY
SUPPRESSED FOR
           OBJECT TYPE TABLE SPACE
           OBJECT NAME DSNDB06.SYSDBASE
           REASON RESTART
```

DT22 and DT24 complete their restart as shown in Example 18-37 and Example 18-38.

Since we are using the LIMIT BACKOUT parameter (together with the BACKOUT DURATION parameter) to delay the backout of long running URs, one of the inflight transactions on DT22 is converted to postponed-abort status (1). Page sets or partitions with postponed backout work are put into restart-pending (RESTP). This state blocks all access to the object other than access by the RECOVER POSTPONED command. As the option AUTO is selected, DB2 automatically issue the RECOVER POSTPONED command (2).

However, no matter what you specify as BACKOUT DURATION, DB2 guarantees that all catalog and directory changes are completely backed out during restart. If you have inflight or inabort URs with activity against the catalog or directory, the backward recovery proceeds until the oldest UR is backed out (3).

Also note that DDF could not be started, due to the restricted state of several objects (4). DDF will have to be manually started after GRECP and LPL recovery.

Example 18-37 Scenario 2 – Restart DT22 (2/2)

```
DSNR005I  -DT22 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
...
DSNR018I  -DT22 RESTART...BACKWARD RECOVERY PROCESSED
 FROM RBA 04025B3DFF62 TO RBA 040251A7ACB3
DSNR006I  -DT22 RESTART...COUNTS AFTER BACKWARD
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=1                          (1)
DSNR007I  -DT22 RESTART...STATUS TABLE
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
I SERVER   t3cli        LGB0032  DISTSERV P 04020375D02F 209 14:27:22
...
-DT22RECOVER POSTPONED                                             (2)
```

```
DSNV435I  -DT22 RESOLUTION OF POSTPONED ABORT URS HAS BEEN SCHEDULED
DSN9022I  -DT22 DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
...
15.51.37 STC37529  DSN9022I  -DT22 DSNYASCP 'STA DB2' NORMAL COMPLETION
15.51.47 STC37529  DSNL008I  -DT22 DDF ABNORMAL TERMINATION COMPLETE
```

DT24 completes the restart as shown in Example 18-38.

Example 18-38 Scenario 2 – Restart DT24 (2/2)

```
DSNR031I  -DT24 DSNRTIMR PROCESSING LOG RECORD AT RBA
0246D494C95A TO RBA 0246D4D44F24
DSNR005I  -DT24 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
...
DSNI030I  -DT24 DSNIARPL BACKOUT PROCESSING IS                    (3)
CONTINUING
        BEYOND THE REQUESTED LIMIT DUE TO CATALOG OR DIRECTORY
        ACTIVITY OF UR 0246CBC15BB5.
...
DSNR031I  -DT24 DSNRTIMR PROCESSING LOG RECORD AT RBA
0246CBFEAA55 TO RBA 0246CA58A190
DSNR031I  -DT24 DSNRTIMR PROCESSING LOG RECORD AT RBA
0246CBD0DBA3 TO RBA 0246CA58A190
DSNR018I  -DT24 RESTART...BACKWARD RECOVERY PROCESSED
 FROM RBA 0246D4D44F24 TO RBA 0246CA58A190
DSNR006I  -DT24 RESTART...COUNTS AFTER BACKWARD
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
...
15.51.50 STC37560  DSNL007I  -DT24 DDF IS ABNORMALLY TERMINATING   (4)
15.51.50 STC37560  DSN9022I  -DT24 DSNYASCP 'STA DB2' NORMAL COMPLETION
```

Execute GRECP and LPL recovery

During our test, it took 14 minutes for both DB2 members to come up. However, many objects, in particular page sets of the DB2 catalog and directory, are still in a restricted state and need to be manually recovered. We issue the DISPLAY commands shown in Example 18-39 to check the status of the DB2 directory (DSNDB01) and catalog (DSNDB06).

Example 18-39 Scenario 2 – Restricted state of DB2 catalog and directory

```
-DT22 DISPLAY DB(DSNDB01) SPACE(*) RESTRICT
-DT22 DISPLAY DB(DSNDB06) SPACE(*) RESTRICT
```

Example 18-40 and Example 18-41 show the output of these commands.

Example 18-40 Scenario 2 – DSNDB01 restricted states

```
DSNT360I  -DT22 ***********************************
DSNT361I  -DT22 *  DISPLAY DATABASE SUMMARY
                *    RESTRICTED
DSNT360I  -DT22 ***********************************
DSNT362I  -DT22     DATABASE = DSNDB01  STATUS = RW
                   DBD LENGTH = 14200
DSNT397I  -DT22
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
DBD01    TS         RW,GRECP
```

```
SYSLGRNX TS         RW,GRECP
DSNLLX01 IX         RW,GRECP,ICOPY
DSNLLX02 IX         RW,GRECP,ICOPY
******* DISPLAY OF DATABASE DSNDB01  ENDED      **********************
```

Example 18-41 shows the output for DSNDB06.

Example 18-41 Scenario 2 – DSNDB06 restricted states

```
DSNT360I  -DT22 ************************************
DSNT361I  -DT22 *  DISPLAY DATABASE SUMMARY
                *    RESTRICTED
DSNT360I  -DT22 ************************************
DSNT362I  -DT22     DATABASE = DSNDB06  STATUS = RW
                   DBD LENGTH = 246308
DSNT397I  -DT22
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
SYSDBASE TS         RW,LPL,GRECP
DSNDSX01 IX         RW,GRECP
DSNDTX01 IX         RW,GRECP
DSNATX02 IX         RW,LPL,GRECP
DSNDXX02 IX         RW,GRECP
DSNDLX01 IX         RW,GRECP
DSNDTX02 IX         RW,GRECP
******* DISPLAY OF DATABASE DSNDB06  ENDED      **********************
```

As we have LPL and GRECP status conditions for page sets of the DB2 catalog and directory, we must issue the appropriate START DATABASE commands for these page sets before we issue the commands for the other page sets or partitions. You must have INSTALL SYSADM authority to issue the START DATABASE command for catalog or directory page sets.

Example 18-42 Scenario 2 – GRECP and LPL recovery for DB2 catalog and directory

```
-DT22 START DB(DSNDB01) SPACE(*)
-DT22 START DB(DSNDB06) SPACE(*)
```

GRECP and LPL recovery is an asynchronous process which might complete after the START DATABASE command has completed. As shown in Example 18-43, message DSNI006I is issued to indicate that GRECP or LPL recovery is in progress; additionally, message DSNI022I might be issued periodically to indicate the recovery status; and message DSNI021I indicates the successful completion of the recovery.

Example 18-43 Scenario 2 – GRECP and LPL recovery – Messages

```
DSNI006I  -DT22 DSNISREC ASYNCHRONOUS GRECP OR LPL
RECOVERY IN PROGRESS FOR
          START DATABASE COMMAND-ID = 1
          TYPE 00000200
          NAME DSNDB01 .SYSUTILX
...
DSN9022I  -DT22 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
...
DSNI021I  -DT22 DSNICLST GRECP OR LPL RECOVERY FOR
          START DATABASE COMMAND-ID = 1
          TYPE 00000200
          NAME DSNDB01 .SYSUTILX
          HAS COMPLETED SUCCESSFULLY.
```

When a page set is placed in the GRECP state, DB2 sets the starting point for the merge log scan to the LRSN of the last complete group buffer pool checkpoint.

As soon as the LPL and GRECP recovery has been successful for the DB2 catalog and directory, we can issue the START commands for the other page sets that are in GRECP or LPL status.

Deal with active utilities

We use the DISPLAY UTIL command to see if any utilities are running. During our test, no utility was running at the time of the consistency group established by XRC (see Example 18-44). Otherwise, we would have had to terminate the active utilities.

Example 18-44 Scenario 2 – DISPLAY UTIL

```
-DT22 DIS UTIL(*)
DSNU112I  -DT22 DSNUGDIS - NO AUTHORIZED UTILITY FOUND FOR UTILID = *
DSN9022I  -DT22 DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

18.4.5 Summary

Here we summarize what we experienced during our test:

- XRECOVER command: This took 24 seconds to complete.
- Delay between primary and secondary volumes: This took < 15 seconds.
- The DB2 members came up in 14 minutes.
- GRECP/LPL recovery for the DB2 catalog and directory: This took 6 minutes.

Note: Keep in mind that we implemented a basic solution (for example, we did not use striping for the journal data sets) and we did not spend any time tuning the XRC environment.

18.5 Scenario 3: Using XRC to get PIT copies

In this scenario, we use XRC to create system level point-in-time (PIT) backup copies of our DB2 data sharing group.

Figure 18-6 illustrates this first part of the scenario:

1. We start the test in a duplex situation. Primary volumes are at level P_1 – secondary at level S_1 (very close to P_1).
2. We execute some work on the system. Primary volumes are now at level P_2 – secondary at level S_2 (very close to P_2). We suspend the duplexing activity at this point.
3. We execute some more work on the system. Primary volumes are now at level P_3 – secondary are still at level S_2. As we are satisfied with the updates, we resume the XRC activity to re-established the duplexing.
4. At the intermediary state, the duplexing is re-established: we are back to normal mode. Primary volumes are at level P_4 – secondary at level S_4 (very close to P_4).

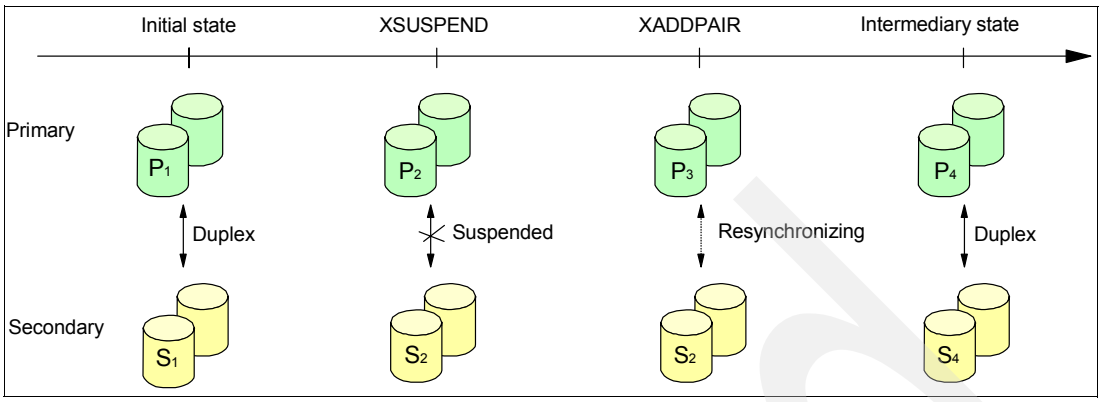


Figure 18-6 Scenario 3 – Part 1

Figure 18-7 illustrates the second part of the scenario:

5. We start from the intermediary state. Primary volumes are at level P_4 – secondary at level S_4 (very close to P_4).
6. We execute some work on the system. Primary volumes are now at level P_5 – secondary at level S_5 (very close to P_5). We suspend the duplexing activity at this point.
7. We execute some more work on the system. Primary volumes are now at level P_6 – secondary volumes are still at level S_5. This time, we want to back out the changes.
8. We use XRC commands to switch from the primary volumes to the secondary volumes. The SDM brings the secondary volumes to the last consistent state (S_5) and reformats their labels to match the labels of their associated primary volumes (P_5).
9. We can restart DB2 from the secondary volumes.

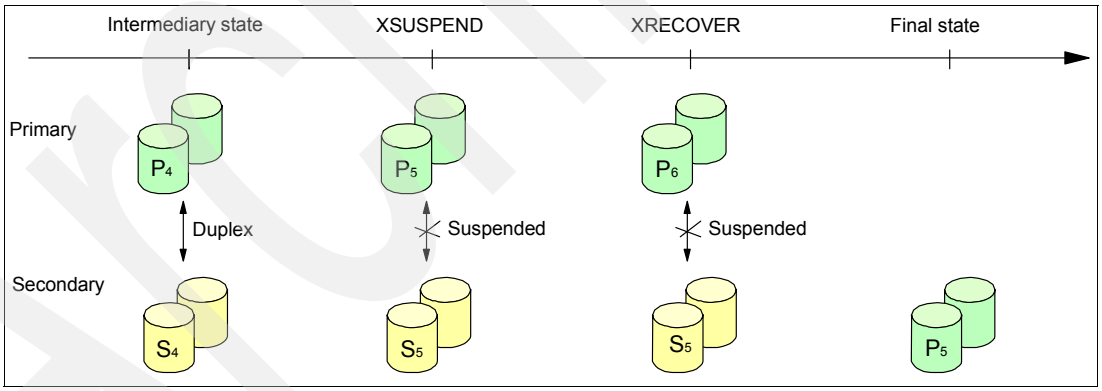


Figure 18-7 Scenario 3 – Part 2

Note: In our test environment, the difference between UTC (used by XRC commands) and Pacific time (printed in the system log) is seven hours.

18.5.1 Preparation

The initial situation is as follows:

- Only two DB2 members are active: DT22 on system B6 and DT24 on system B7.
- The XRC session is established – all 192 volumes are mirrored, as shown on the XQUERY summary report in Example 18-45.

Example 18-45 Scenario 3 – Initial situation

```
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY SUMMARY REPORT - 003
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=192  DUP=192  CPY=0  PND=0  SUS=0  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.210 17:10:34.819343)  DELAY(00:00:05.51)
ANTQ8240I DATA EXPOSURE(00:00:05.51)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SVL)
```

18.5.2 Suspend the XRC volumes pairs

We use the XSUSPEND command as shown in Example 18-46 to suspend *all the volume pairs* in the XRC session SVL. Note that we do not suspend the session itself, as it would terminate the SDM and we would have to restart it before we can resynchronize the pairs.

Example 18-46 Scenario 3 – XSUSPEND command

```
XSUSPEND SVL VOLUME(ALL)
```

Example 18-47 shows some of the messages that are sent to the system log. A message ANTV8109I is issued for every volume pair, indicating the time of the consistency group established by XRC. When all the volume pairs are suspended, a completion message (ANTX8120I) is issued. During our test, it took 20 microseconds for the XSUSPEND command to complete.

Example 18-47 Scenario 3 – XSUSPEND command – Messages in syslog

```
ANTL8800I XSUSPEND SVL VOLUME(ALL)
ANTV8109I XSUSPEND(IMMEDIATE) COMPLETE FOR VOLUME PAIR(T2N865,T2X166)
FOR SESSION(SVL) AT CONSISTENCY_GROUP TIME(2004.210
ANTV8109I (CONT) 17:20:34.819343) AND STATUS DUPLEX
ANTV8109I XSUSPEND(IMMEDIATE) COMPLETE FOR VOLUME PAIR(T2N866,T2X167)
FOR SESSION(SVL) AT CONSISTENCY_GROUP TIME(2004.210
ANTV8109I (CONT) 17:20:34.819343) AND STATUS DUPLEX
...
ANTX8120I ALL VOLUMES IN SESSION(SVL) ARE NOW SUSPENDED
```

Example 18-48 shows an XQUERY volume report taken after all the pairs have been suspended (SUS=192).

Example 18-48 Scenario 3 – XQUERY volume report

```
ANTL8800I XQUERY SVL VOL(ALL)
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY VOLUME REPORT - 003
ANTQ8211I PRIM   SEC     ERROR  SYNCH
ANTQ8212I VOL    VOL     LEVEL    % STA CMD OP -------TIMESTAMP--------
ANTQ8203I -------------------------------------------------------------
ANTQ8213I T2N700 T2X001 SESSION  99 SUS        2004.210 17:20:34.819343
ANTQ8213I T2N701 T2X002 SESSION  99 SUS        2004.210 17:20:34.819343
...
ANTQ8238I TOTAL=192  DUP=0  CPY=0  PND=0  SUS=192  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.210 17:20:34.819343)  DELAY(00:00:07.02)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY VOLUME REPORT COMPLETE FOR SESSION(SVL)
```

18.5.3 Resynchonize the XRC volume pairs

We execute some work on the system. The outcome is positive, so we decide to keep the changes and resynchronize the XRC volume pairs.

While the pairs are suspended, the ESS continues to record changed tracks in a hardware bit map with no impact to the primary system. XRC uses this bit map to resynchronize the volumes when the volumes are added back to the session, using the XADDPAIR command with the parameter SUSPENDED, as shown in Example 18-49.

Example 18-49 Scenario 3 – XADDPAIR command

```
XADDPAIR SVL SUSPENDED
```

Example 18-50 shows some of the messages that are sent to the system log. When the SUSPENDED parameter is specified, XRC automatically resynchronizes all the suspended pairs (1) and brings the secondary volumes in duplex status (2). When all the volume pairs are in duplex status, the XADDPAIR command issues a completion message (ANTA8124I). During our test, it took two minutes for the resynchronization to complete.

Example 18-50 Scenario 3 – XADDPAIR command – Messages in syslog

```
10:33:54.09   ANTL8800I XADDPAIR SVL SUSPENDED
ANTA8101I XADDPAIR COMPLETE FOR VOLUME PAIR(T2N865,T2X166) FOR
SESSION(SVL) WITH ERRORLEVEL(SESSION)
....
ANTI8023I RESYNC INITIALIZATION STARTED FOR XRC VOLUME PAIR(T2N865,          (1)
T2X166)
....
ANTA8101I XADDPAIR COMPLETE FOR VOLUME PAIR(T2N865,T2X166) FOR
SESSION(SVL) WITH ERRORLEVEL(SESSION)
...
ANTI8024I XRC VOLUME PAIR(T2N865,T2X166) IN DUPLEX STATUS AT TIME OF         (2)
2004.210 17:34:26.087431
...
ANTA8124I XADDPAIR SUSPENDED COMPLETED FOR SESSION(SVL)
10:37:55.74   ANTX8120I 192 VOLUMES IN SESSION(SVL) ARE NOW DUPLEX
```

Example 18-51 shows an XQUERY summary report taken while the pairs are being resynchronized. We can see that 54 pairs are in duplex status (DUP=54), 7 are being synchronized (CPY=7), and 131 are still to be processed (PND=4 and SUS=127).

Example 18-51 Scenario 3 – XQUERY summary report

```
ANTL8800I XQUERY SVL
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001) 901
ANTQ8202I XQUERY SUMMARY REPORT - 003
ANTQ8203I -------------------------------------------------------------
ANTQ8238I TOTAL=192  DUP=54  CPY=7  PND=4  SUS=127  SEQ=0  UTL=0
ANTQ8231I DATA CONSISTENT(2004.210 17:34:50.373880)   IDLE(00:00:01.4)
ANTQ8240I DATA EXPOSURE(00:00:00.00)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8233I DEFAULT TIMEOUT(STORAGE_CONTROL_DEFAULT)
ANTQ8201I XQUERY SUMMARY REPORT COMPLETE FOR SESSION(SVL)
```

18.5.4 Suspend the XRC volume pairs

Once duplexing is established, we create a new system level point-in-time (PIT) backup copy of our DB2 data sharing group, by suspending the volume pairs (see Example 18-52) at 10:38:44.81. The XSUSPEND command issues a completion message (ANTX8120I) indicating that all volumes in session SVL are suspended. The consistency group time is 10:38:43.68, that is about one second before the suspension.

Example 18-52 Scenario 3 – XSUSPEND command – Messages

```
10:38:44.81   ANTL8800I XSUSPEND SVL VOL(ALL)
ANTV8109I XSUSPEND(IMMEDIATE) COMPLETE FOR VOLUME PAIR(T2N865,T2X166)
FOR SESSION(SVL) AT CONSISTENCY_GROUP TIME(2004.210
ANTV8109I (CONT) 17:38:43.684090) AND STATUS DUPLEX
...
10:38:45.03   ANTX8120I ALL VOLUMES IN SESSION(SVL) ARE NOW SUSPENDED
```

We run some more work on the system, but this time, we need to come back to the PIT backup in order to back out the changes.

18.5.5 Recovery to the secondary volumes

We quiesce both DB2 members and switch to the secondary volumes.

Vary all primary volumes offline to the SDM system

First, we unallocate the ICF catalog that resides on the primary volumes, and we take all primary volumes offline to the SDM system, as shown in Example 18-53.

In our specific configuration, since the SDM is part of a parallel sysplex, we need to route the commands to all the systems in the sysplex.

Example 18-53 Scenario 3 – Vary all primary volumes offline

```
RO *ALL,F CATALOG,UNALLOCATE(SYS1.ICFCAT.DSNT2)
RO *ALL,V 6000-600B,OFFLINE
```

Recover to the secondary volumes

We use the XRECOVER command as shown in Example 18-54 to switch to the secondary volumes. This XRECOVER command will bring data on the secondary volumes to the last consistent, recoverable state prior to the suspension.

Example 18-54 Scenario 3 – XRECOVER command

```
XRECOVER SVL HLQ(XRC)
```

Example 18-55 shows some of the messages that are sent to the system log. When all updates have been applied to the secondary volumes, the XRECOVER command automatically generates an XQUERY recover report. The report includes the last applied timestamp that defines the recovered, consistent state for all volumes – 10:38:43.68 in this case, that is about one second before the suspension.

During our test, it took 24 seconds for the XRECOVER command to complete.

Example 18-55 Scenario 3 – XRECOVER command – Messages in syslog

```
ANTL8800I XRECOVER SVL HLQ(XRC)
...
ANTR8108I XRECOVER STARTED FOR SESSION(SVL) AT STARTING CONSISTENCY_GRO
P TIME(2004.210 17:38:43.684090), HLQ(XRC)
ANTQ8200I XQUERY STARTED FOR SESSION(SVL) ASNAME(ANTAS001)
ANTQ8202I XQUERY RECOVER REPORT - 002
ANTQ8271I ------ORIGINAL-------
ANTQ8274I PRIMARY  SECONDARY STA CON CLP  ------TIMESTAMP---------
ANTQ8203I ------------------------------------------------------------
ANTQ8275I T2N865   T2X166    SUS YES YES  2004.210 17:38:43.684090
ANTQ8275I T2N866   T2X167    SUS YES YES  2004.210 17:38:43.684090
ANTQ8275I T2N867   T2X168    SUS YES YES  2004.210 17:38:43.684090
...
ANTQ8237I TOTAL=192   DUP=0   PND=0   SUS=192
ANTQ8231I DATA CONSISTENT(2004.210 17:38:43.684090)
ANTQ8232I SESSIONTYPE(XRC)  ERRORLEVEL(SESSION)  HLQ(XRC)
ANTQ8201I XQUERY RECOVER REPORT COMPLETE FOR SESSION(SVL)
IEE302I 6031     ONLINE
IEE302I 6032     ONLINE
IEE302I 6033     ONLINE
...
ANTR8102I XRECOVER COMPLETE FOR SESSION(SVL) AT RECOVERED CONSISTENCY_GR
OUP TIME(2004.210 17:38:43.684090)
...
IEF196I IEF375I  JOB/ANTAS001/START 2004210.1050
IEF196I IEF376I  JOB/ANTAS001/STOP  2004210.1051 CPU    0MIN 23.63SEC
```

The XRECOVER command brings the secondary volumes online to the SDM system only. In our specific configuration, since the SDM is part of a parallel sysplex, we need to make them available to the other systems in the sysplex, as shown in Example 18-56.

Example 18-56 Scenario 2 – Vary all secondary volumes online

```
RO *ALL,V 6031-6034,ONLINE
```

18.5.6 Restart DB2

Finally, we restart DB2 from the secondary volumes.

Clean up the DB2 structures

Before we start DB2, we delete the structures belonging to the data sharing group DSNT2.

First, we enter the commands shown in Example 18-57 to force the connections off the group buffer pools and LOCK structures (connections for the SCA are not held at termination, so there are no SCA connections to force off).

Example 18-57 Scenario 3 – Clean up the DB2 structures (1/2)

```
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_GBP0
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_GBP1
...
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNT2_LOCK1
```

Then, we enter the commands shown in Example 18-58 to delete all the structures.

Example 18-58 Scenario 3 – Clean up the DB2 structures (2/2)

```
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_GBP0
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_GBP1
...
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_LOCK1
SETXCF FORCE,STRUCTURE,STRNAME=DSNT2_SCA
```

Finally, we verify that the structures have been successfully deleted. Example 18-59 shows the status of the structures – all the structures belonging to the data sharing group DSNT2 must be in status 'NOT ALLOCATED'.

Example 18-59 Scenario 3 – Clean up the structures – Verification

```
D XCF,STR
IXC359I  15.37.48  DISPLAY XCF 596
STRNAME            ALLOCATION TIME   STATUS
DSNT2_GBP0            --       --    NOT ALLOCATED
DSNT2_GBP1            --       --    NOT ALLOCATED
...
DSNT2_LOCK1           --       --    NOT ALLOCATED
DSNT2_SCA             --       --    NOT ALLOCATED
```

Start DB2

We can now start DB2. Since the DB2 members DT22 and DT24 were active at the time of the consistency group established by XRC, we should bring up at least these members (in order to release the retained locks that they might hold). The members will restart in the same way they would do after a disaster. GRECP and LPL recovery will have to be executed like in the previous scenario. For more details, see 18.4.4, "Restart DB2" on page 349.

18.5.7 Summary

The following summarizes what we experienced during our test:

- XRECOVER command: 24 seconds to complete
- Delay between primary and secondary volumes: about 1 second
- The DB2 members came up in 9 minutes
- GRECP/LPL recovery for the DB2 catalog and directory: 1 minute

Note: Keep in mind that we implemented a basic solution (e.g. we did not use striping for the journal data sets) and we did not spend any time tuning the XRC environment.

19

Local recovery: System PITR

DB2 for z/OS V8 provides enhanced backup and recover capabilities at the DB2 subsystem or data sharing group level. The purpose is to provide an easier and less disruptive way to make fast volume level backups of an entire DB2 subsystem or data sharing group with minimal disruption, and recover the subsystem or data sharing group to any point-in-time.

This chapter describes system level point-in-time recovery (PITR) procedures and scenarios based on the new DB2 utilities BACKUP SYSTEM and RESTORE SYSTEM, and their interface to the new z/OS V1R5 DFSMShsm services and SMS constructs that are described in Chapter 6, “SMS copy pools and DB2 point in time recovery” on page 79.

The following publications should be referenced to complement the information presented in this chapter:

- *DFSMShsm Fast Replication Technical Guide*, SG24-7069
- *DB2 UDB for z/OS Version 8 Utility Guide and Reference,* SC18-7427
- *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413
- *DB2 UDB for z/OS Version 8 Technical Preview*, SG24-6871
- *DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There*, SG24-6097

This chapter covers the following topics:

- The need for system PITR
- Description of the scenarios
- Description of the test environment
- SMS definitions
- Restoring to an arbitrary point-in-time
- Considerations

19.1 The need for system PITR

Many Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) applications consider the entire DB2 subsystem and all objects in it, including user databases and the DB2 catalog and directory, as a single entity. This means that the entire DB2 system needs to be backed up and recovered as a single entity. If you need to recover your system to a prior point-in-time, all database objects must simultaneously be recovered to the exact point-in-time arbitrarily chosen in the past to guarantee consistency. However, when a single DB2 has more that 50,000 tables/indexes, as it is often the case with ERP/CRM applications, it is too slow and too difficult to manage the recovery at the table space and index levels.

This is why ERP/CRM customers expressed the need for:

- A fast non-disruptive way to take a consistent backup of an entire DB2 subsystem or data sharing group
- A system level recovery utility that would support point-in-time recovery of the entire DB2 subsystem or data sharing group at a user given time

System PITR – Prior to DB2 V8

Prior to DB2 V8, volume level backup strategies can be implemented using fast copy services provided by disk storage vendors (similar to IBM's ESS FlashCopy and RVA SnapShot). However, to ensure that the data and logs in the backup are consistent for restart recovery, you need to use the SET LOG SUSPEND command. This command stops all write activity while the copy relationship is being established, causing a possible disruption in service.

After the copy is complete, the data have to be dumped to tape so that they can be available at some point in the future in case recovery is required. You can also manage several backup versions on disk, but this has to be done manually.

Recovering the system to the time of the backup means restoring all the volume backups (data and logs) and restart DB2 from there. But this could result in significant data loss.

Recovering the system to any point-in-time after the time of the backup is also possible. But it means restoring the volume backups containing the DB2 catalog and directory and the user databases from tape to the source data volumes (the logs are not restored), identifying which objects have been changed with some kind of log scanning process, creating recovery JCL for all objects identified as requiring recovery, and recovering each object to the same point-in-time. At best this process is time consuming, labor intensive, and very prone to error.

System PITR – With DB2 V8

DB2 V8 delivers two new system utilities to simplify and improve the performance and reliability of the backup and recovery process. The BACKUP SYSTEM and RESTORE SYSTEM utilities take advantage of the new z/OS V1R5 DFSMSdss Fast Replication function and encapsulate all tasks previously required to perform each function into one utility statement.

The BACKUP SYSTEM utility provides fast non-disruptive volume level copies of DB2 databases and logs.

The RESTORE SYSTEM utility recovers a DB2 system or a data sharing group to an arbitrary point-in-time. RESTORE SYSTEM automatically handles any creates and drops that might have occurred between the backup and the recovery point-in-time. If LOG NO activities had been done on some objects between the CopyPool backup and the PITR log truncation, these objects are flagged for recovery with RECP, RBDP, PSRBD.

For detailed information about those utilities, refer to the DB2 manuals and the redbook *DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There*, SG24-7088.

19.2 Description of the scenarios

This section summarizes the steps necessary to recover an entire DB2 subsystem or data sharing group to a previous point-in-time. The procedures vary depending on the point-in-time to which you want to recover:

- Recover to an *arbitrary point-in-time*, either between two backup times or between the last backup time and the current time.

  In this case, after the appropriate volume copies have been restored, the outstanding logs are applied to the databases to recover the data to the designated point-in-time.
- Recover to the *point-in-time of a backup*.

  In this case, the recovery involves restoring the appropriate volume copies of the data and logs. The logs are used only to back out inflight transactions on restart.

We differentiate the procedure for non-data sharing and data sharing environments.

19.2.1 Non-data sharing

If you are *not* using DB2 data sharing, use the following procedures.

Restoring to an arbitrary point-in-time

To restore the system to an arbitrary point-in-time, use the following procedure:

Backup

1. Use BACKUP SYSTEM DATA ONLY to take the backup. You can also take a FULL backup although it is not needed.

Recovery

1. Stop the DB2 subsystem.
2. Unallocate the ICF catalogs associated with the *data* copy pool.
3. Clean up the volumes that may have been added to the *data* copy pool since the backup[1].
4. Find the RBA (log truncation point) corresponding to the point-in-time to which the system is to be recovered[2].
5. Run the Change Log Inventory utility (DSNJU003) with the CRESTART SYSPITR option specifying the log truncation point identified in the previous step.
6. Start the DB2 subsystem. DB2 will start in system recover-pending mode.
7. Submit the RESTORE SYSTEM utility job.
8. After the utility ends successfully, stop DB2. This will reset system recover-pending status.
9. Start the DB2 subsystem.
10. Use the DISPLAY UTIL command to see if any utilities are in stopped status. If utilities are displayed, use the TERM UTIL command to end them.
11. Execute the following display commands to check for restricted objects:

```
-DIS DB(*) SPACE(*) LIMIT(*) RESTRICT
```

[1] See 19.6.1, “What if you added volumes between backup time and restore time?” on page 379
[2] See 2.5.2, “Choosing an RBA value for SYSPITR” on page 36

12. Use the RECOVER utility to recover all objects in RECOVER-pending (RECP) or REBUILD-pending (RBDP) mode, or use the REBUILD INDEX utility to rebuild indexes. If a CREATE TABLESPACE, CREATE INDEX, or data set extension has failed, you may also have to recover or rebuild objects in the logical page list (LPL). If you have restricted status conditions for page sets of the DB2 catalog and directory, you must recover or rebuild these page sets before you recover the other page sets or partitions.

Restoring to a previous backup

To restore the system to a previous backup, use the following procedure:

Backup

1. Use the BACKUP SYSTEM FULL utility to backup both data *and* log copy pools.

Recovery

1. Stop the DB2 subsystem.
2. Unallocate the ICF catalogs associated with both data *and* log copy pools.
3. Use DFSMSdss to restore both data *and* log copy pools from the desired backup (identified by the TOKEN parameter):

```
FRRECOV COPYPOOL(DSN$locn$DB) VERIFY(YES) TOKEN(token)
FRRECOV COPYPOOL(DSN$locn$LG) VERIFY(YES) TOKEN(token)
```

4. Clean up the volumes that may have been added to the *data* copy pool since the backup[3].
5. Start the DB2 subsystem (normal restart).
6. Use the DISPLAY UTIL command to see if any utilities are in stopped status. If utilities are displayed, use the TERM UTIL command to end them.
7. Execute the following display commands to verify that no object is in a restricted state:

```
-DIS DB(*) SPACE(*) LIMIT(*) RESTRICT
```

19.2.2 Data sharing

If you are using DB2 data sharing, use the following procedures.

Restoring to an arbitrary point-in-time

To restore the system to an arbitrary point-in-time, use the following procedure:

Backup

1. Use BACKUP SYSTEM DATA ONLY to take the backup. You can also take a FULL backup although it is not needed.

Recovery

1. Stop all the members of the group.
2. Delete all CF structures owned by this group.
3. Unallocate the ICF catalogs associated with the *data* copy pool.
4. Clean up the volumes that may have been added to the *data* copy pool since the backup[3].
5. Find the LRSN (log truncation point) corresponding to the point-in-time to which the system is to be recovered[4], normally the lowest common LRSN of all members.

[3] See 19.6.1, “What if you added volumes between backup time and restore time?” on page 379
[4] See 2.5.2, “Choosing an RBA value for SYSPITR” on page 36

6. Run the Change Log Inventory utility (DSNJU003) with the CRESTART SYSPITR option specifying the log truncation point identified in the previous step. Ensure that all members that were active at the SYSPITR log truncation point (or restarted after this point) are restarted with the same SYSPITR LRSN.
7. Start *all active members*. DB2 will start in system recover-pending mode.
8. Submit the RESTORE SYSTEM utility job. The utility can only be executed on one member (any member). If the utility terminates and must be restarted, it can only be restarted on the member on which it was originally executed.
9. After the utility ends successfully, stop all active members. This will reset system recover-pending status.
10. Restart all active members of the group.
11. Use the DISPLAY UTIL command to see if any utilities are in stopped status. If utilities are displayed, use the TERM UTIL command to end them.
12. Execute the following display commands to check for restricted objects:

    ```
    -DIS DB(*) SPACE(*) LIMIT(*) RESTRICT
    ```

13. Use the RECOVER utility to recover all objects in RECOVER-pending (RECP) or REBUILD-pending (RBDP) mode, or use the REBUILD INDEX utility to rebuild indexes. If a CREATE TABLESPACE, CREATE INDEX, or data set extension has failed, you may also have to recover or rebuild objects in the logical page list (LPL). If you have restricted status conditions for page sets of the DB2 catalog and directory, you must recover or rebuild these page sets before you recover the other page sets or partitions.

Restoring to a previous backup

To restore the system to a previous backup, use the following procedure:

Backup

1. Use the BACKUP SYSTEM FULL utility to backup both data *and* log copy pools.

Recovery

1. Stop all members of the group.
2. Delete all CF structures owned by this group.
3. Unallocate the ICF catalogs associated with both data *and* log copy pools.
4. Use DFSMSdss to restore both data *and* log copy pools from the desired backup (identified by the TOKEN value issued by the BACKUP or listed in the BSDS):

    ```
    FRRECOV COPYPOOL(DSN$locn$DB) VERIFY(YES) TOKEN(token)
    FRRECOV COPYPOOL(DSN$locn$LG) VERIFY(YES) TOKEN(token)
    ```

5. Clean up the volumes that may have been added to the *data* copy pool since the backup[5].
6. Start all active members.
7. Execute the following display commands to check for restricted objects:

    ```
    -DIS DB(*) SPACE(*) LIMIT(*) RESTRICT
    ```

8. Execute the GRECP and LPL recovery, which recovers the changed data that were stored in the coupling facility at the time of the backup. If you have LPL and GRECP status conditions for page sets of the DB2 catalog and directory, you must issue the appropriate START DATABASE commands for these page sets before you issue the commands for the other page sets or partitions[6].

[5] See 19.6.1, “What if you added volumes between backup time and restore time?” on page 379
[6] See 21.6.5, “GRECP/LPL recovery recommendations” on page 428

9. Use the DISPLAY UTIL command to see if any utilities are in stopped status. If utilities are displayed, use the TERM UTIL command to end them.

19.3 Description of the test environment

The tests were conducted at the IBM SAP Integration & Support Center, IBM Silicon Valley Lab, in a *two-way data sharing environment*, using the following hardware and software:

- Hardware:
  - zSeries 900 processor Model 116
  - 1 Logical Partition (LPAR) with 8 dedicated CPUs and 3 GB real memory
  - 2 Internal Coupling Facilities (ICF) – CFCC release 12
- Disks:
  - ESS Model 800 with 8 FICON channels – FlashCopy V2
  - RVA Model X82 with 8 FICON channels – SnapShot
- Software:
  - z/OS V1R5
  - DB2 for z/OS V8 New Function Mode, service level PUT0406 (COR0407)
  - SAP Basis 6.20, patch number 1474

The two members of the data sharing group are called P870 and P872. Both run on the same LPAR, as shown in Figure 19-1.

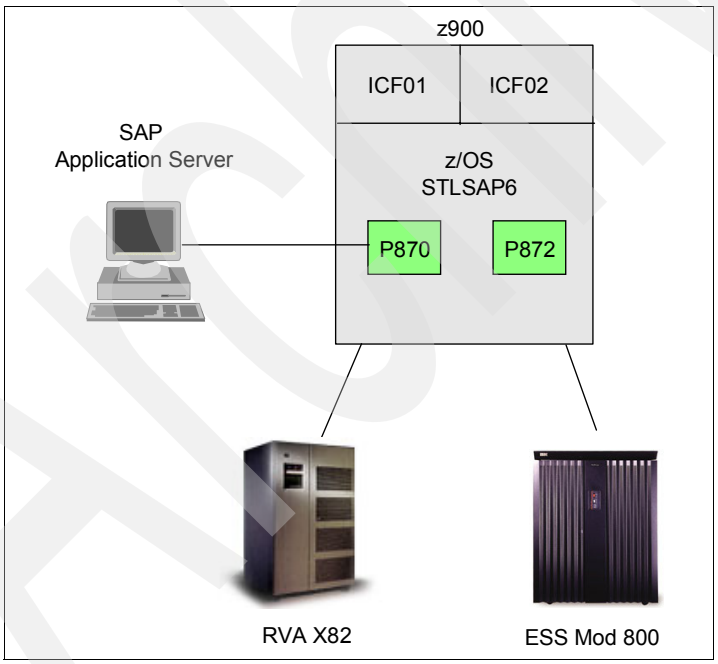


Figure 19-1 System PITR test environment

19.4 SMS definitions

As required for the BACKUP SYSTEM and RESTORE SYSTEM utilities, all volumes containing DB2-related data, including the DB2 catalog, directory and logs, are SMS managed.

Additionally, two copy pools and a copy pool backup storage group have been defined:

- One copy pool called DSN$P870$DB for DB2 catalog/directory, user databases, and associated ICF catalogs:
  - P87VCAT1.USER.CATALOG
  - P88VCAT1.USER.CATALOG
  - P89VCAT1.USER.CATALOG
- One copy pool called DSN$P870$LG for DB2 logs and BSDSs of all members, and associated ICF catalog:
  - DSNP870.USER.CATALOG
- One copy pool backup storage group called P87VCATP defined to hold one backup version of the two copy pools (for a total of 4 copy pools)

Note: It is recommended to keep at least two backup versions in the copy pool backup storage group, although we chose not to do so in our test environment.

For more information on how to set up an SMS environment that supports the BACKUP SYSTEM and RESTORE SYSTEM utilities, refer to 20.4, “Setting up the SMS environment” on page 392.

Figure 19-2 illustrates the SMS definitions in the system PITR test environment.

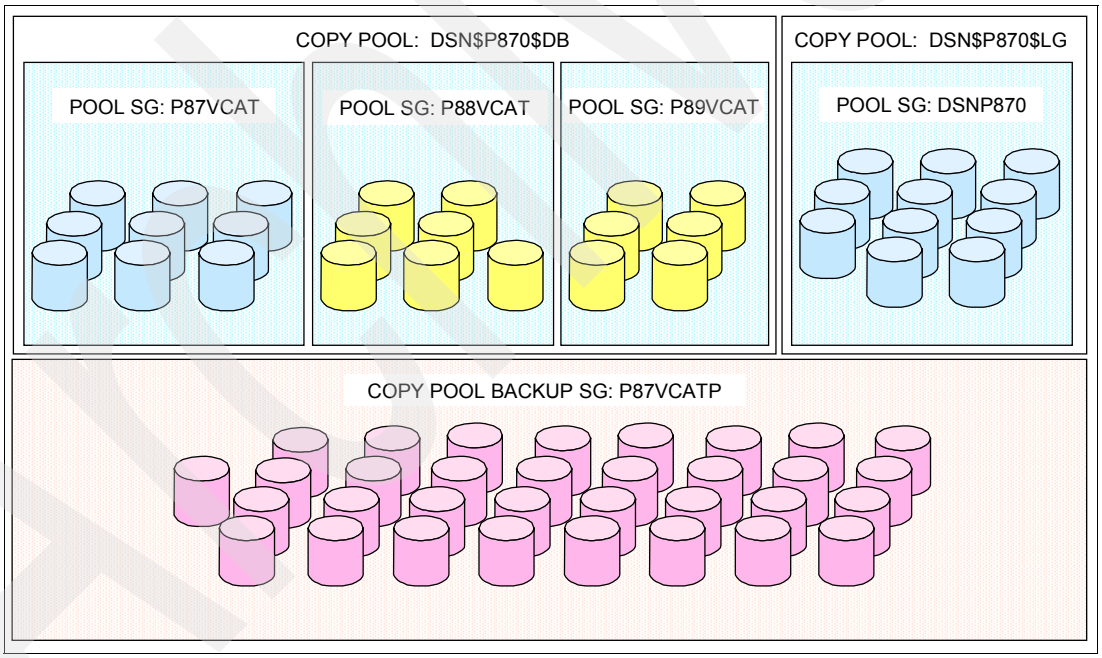


Figure 19-2 SMS definitions for system PITR test

19.5 Restoring to an arbitrary point-in-time

In this scenario, we use the RESTORE SYSTEM utility to recover to a point-in-time (point B) chosen between the last backup time (point A, backup created using the BACKUP SYSTEM utility) and the current time (point C), as illustrated in Figure 19-3.

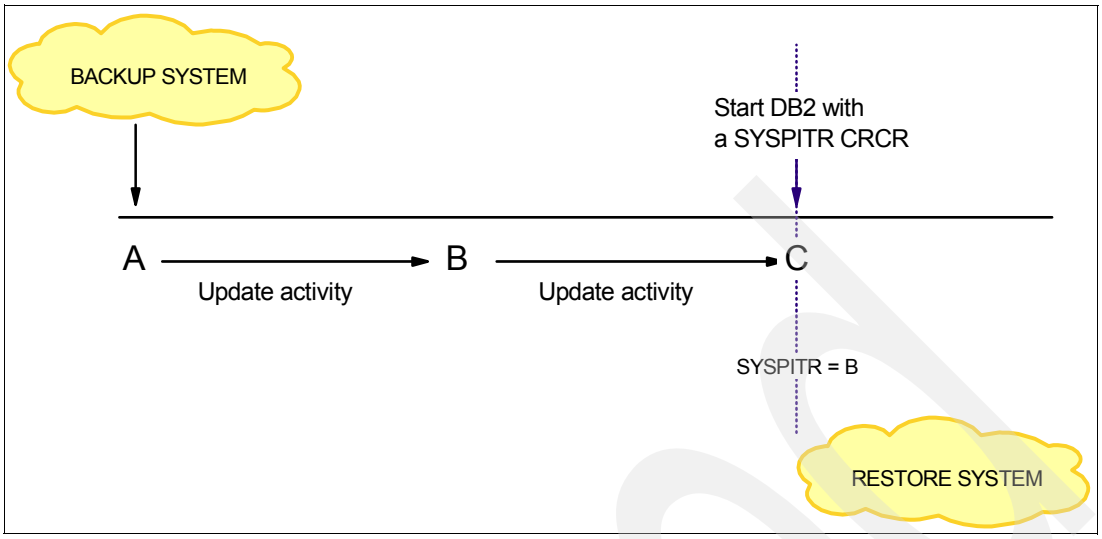


Figure 19-3 Scenario – Restoring to an arbitrary point-in-time

19.5.1 Creating the backup

For the purpose of our test, we use the BACKUP SYSTEM utility with the option DATA ONLY to take a backup of the data copy pool (see Example 19-1). We execute the utility on P870.

Example 19-1 BACKUP SYSTEM utility

```
//BACKUP   EXEC DSNUPROC,PARM='P870,BACKUP'
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  BACKUP SYSTEM DATA ONLY
//
```

The output of the BACKUP SYSTEM utility is shown in Example 19-2. Notice that only the *data* copy pool DSN$P870$DB is copied.

Example 19-2 BACKUP SYSTEM utility – Output

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = BACKUP
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  BACKUP SYSTEM DATA ONLY
DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA STARTING,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BB8E5142855EF907BB8E502C0AC4'.
DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BB8E5142855EF907BB8E502C0AC4'
                       ELAPSED TIME = 00:00:16.
DSNU1602I   DSNUVBBD - BACKUP SYSTEM UTILITY COMPLETED, ELAPSED TIME = 00:00:19.
```

A successful BACKUP SYSTEM results in a *token* being established with DFSMShsm to identify the backup. The token is stored in the BSDS of the submitting member (P870), in the new “BACKUP SYSTEM utility history” section. The BSDS of the other member is not updated.

Example 19-3 is an extract of the Print Log Map utility (DSNJU004) showing the “BACKUP SYSTEM utility history” of member P870. Two system level backups are recorded:

- BACKUP SYSTEM FULL backup taken on 2003/10/02
- BACKUP SYSTEM DATA ONLY backup taken on 2004/07/22 (the LOG column is empty)

The information contained in the BSDS will be used by the RESTORE SYSTEM utility to determine the backup version that must be restored. RESTORE SYSTEM will use the latest backup version before the point-in-time of the recovery.

Example 19-3 BSDS content after BACKUP SYSTEM – DSNJU004 output

```
                    BACKUP SYSTEM UTILITY HISTORY
                         SUBSYSTEM ID P870
                      20:59:59 JULY 22, 2004
            START STCK                          DATA COMPLETE   DATA/LOG COMPLETE
     DATA               LOG             RBLP          LRSN         DATE      LTIME    LOCATION NAME
----------------  ----------------  ------------  ------------  --------------------  ----------------
BB8E5142855EF907  0000000000000000  BB8E502C0AC4  BB8E502C0AC4  2004/07/22  13:59:45  P870
                 TOKEN = D7F8F7F0BB8E5142855EF907BB8E502C0AC4
BA1CCAA9A3820B8A  BA1CCAB1FF964845  00120CDAC090  00120CDAC090  2003/10/02  16:12:43  P870
                 TOKEN = D7F8F7F0BA1CCAA9A3820B8A00120CDAC090
```

The ending bytes of the token contain the recover base log point (RBLP) RBA, which will be used by the RESTORE SYSTEM utility as the log scan starting point.

The RBLP is stored in the header page of DBD01, as shown in Example 19-4, and it is updated every time you run either the BACKUP SYSTEM utility or the SET LOG SUSPEND command.

Example 19-4 DBD01 header page – DSN1PRINT output

```
DSN1999I START OF DSN1PRNT FOR JOB PITBSD72 PRINT
DSN1989I DSN1PRNT IS PROCESSED WITH THE FOLLOWING OPTIONS:
          4K/NO IMAGECOPY/NUMPARTS =   0/   FORMAT/NO EXPAND/NO SWONLY/   PRINT/NO VALUE/     /   /
          DSSIZE=   /PIECESIZ=    /       /
DSN1998I INPUT  DSNAME = P87VCAT.DSNDBC.DSNDB01.DBD01.I0001.A001     , VSAM

PAGE: # 00000000 ---------------------------------------------------------------------------------
HEADER PAGE:  PGCOMB='10'X  PGLOGRBA='00000000BEE2'X  PGNUM='00000000'X  PGFLAGS='18'X
              HPGOBID='0001001F'X  HPGHPREF='000055C8'X  HPGCATRL='D3'X  HPGREL='D3'X  HPGZLD='00'X
              HPGCATV='50'X  HPGTORBA='000000000000'X  HPGTSTMP='0000000000000000000'X
              HPGSSNM='....'  HPGFOID='0010'X  HPGPGSZ='1000'X  HPGSGSZ='0000'X  HPGPARTN='0000'X
              HPGZ3PNO='000000'X  HPGZNUMP='00'X  HPGTBLC='0002'X  HPGROID='0000'X
              HPGZ4PNO='00000000'X  HPGMAXL='0000'X  HPGNUMCO='0000'X  HPGFLAGS='2810'X
              HPGCONTM='00000000000000000000'X  HPGSGNAM='........'  HPGVCATN='........'
              HPGRBRBA='BAC4312A5C56'X  HPGLEVEL='BAC4312A5C56'X  HPGPLEVL='001239FC0478'X
              HPGCLRSN='000000000060'X  HPGSCCSI='0025'X  HPGDCCSI='0000'X  HPGMCCSI='0000'X
              HPGFLAG2='00'X  HPGEPOCH='0000'X  HPGRBLP='BB8E502C0AC4'X  HPGDNUMB='1F'X
              HPGDNUMC='0007'X  HPGDFSG='00000000'X  HPGDLSG='00000000'X  FOEND='N'

DSN1994I DSN1PRNT COMPLETED SUCCESSFULLY,  00000001 PAGES PROCESSED
```

Note that the token is also stored by DFSMShsm in its BCDS records. For more information on how DFSMShsm records the backup versions, see “Chapter 20, “Restart using tape dump of copy pools” on page 389”.

19.5.2 Generating log activity

We generate some workload on both members:

1. Utilities:
   a. P872 – We prepare for the online REORG utility by creating a mapping table TOLRUMAP with a unique index XOLRUMAP.
   b. P870 – Online Reorg of A230X999.LCABNT

      ```
      REORG TABLESPACE A230X999.LCABNT
      SORTDEVT SYSDA SORTNUM 32 COPYDDN(COPYDDN)
      SHRLEVEL CHANGE LOG NO MAPPINGTABLE TOLRUMAP
      KEEPDICTIONARY
      STATISTICS TABLE(ALL) INDEX(ALL) UPDATE ALL KEYCARD REPORT YES
      ```

   c. P870 – Reorg of A230X999.LCABN

      ```
      REORG TABLESPACE A230X999.LCABN (default: SHRLEVEL NONE LOG YES)
      ```

   d. P870 – Runstats of 155 table spaces

      ```
      RUNSTATS TABLESPACE SX60XAAA.LREPODVF SHRLEVEL CHANGE
      ...
      RUNSTATS TABLESPACE A201XAAA.XSAP SHRLEVEL CHANGE
      ```

2. SQL activity:
   a. P872 – Insert 425,000 rows into table SAPR3.SWD_TEXT
   b. P870 – Insert 330,000 rows into table SAPR3.SREPOVARI
   c. P870 – Delete 325,000 rows from table SAPR3.SWD_TEXT (100,000 rows left)
   d. P870 – Delete 300,000 rows from table SAPR3.SREPOVARI (30,000 rows left)
3. SAP activity:
   a. Create and generate a few ABAP programs that manipulate LOB objects. Note that ABAP programs are stored in various DB2 tables.
   b. Run transports – This will generate some activity on the SAP database.

Just for the purpose of our test, we issue a SET LOG SUSPEND command on every data sharing member, immediately followed by a SET LOG RESUME command. The SET LOG SUSPEND command returns a LRSN value for each member, as shown in Example 19-5. We will use the lowest LRSN among active members to establish log truncation point *logpoint1*. This is the point to which we want to recover.

This is done only to simplify the LRSN value determination. In real production this is not necessary and it would have the negative effect of suspending I/O to the log.

Example 19-5 SET LOG SUSPEND command

```
-P870 SET LOG SUSPEND
DSN9022I  -P870 DSNJC001 '-SET LOG' NORMAL COMPLETION
DSNJ372I  -P870 DSNJC09A UPDATE ACTIVITY HAS BEEN
SUSPENDED FOR P870 AT RBA 001253C84D81, LRSN BB8E6368132F, PRIOR
CHECKPOINT RBA 001250B61C08
-P872 SET LOG SUSPEND
DSN9022I  -P872 DSNJC001 '-SET LOG' NORMAL COMPLETION
DSNJ372I  -P872 DSNJC09A UPDATE ACTIVITY HAS BEEN
SUSPENDED FOR P872 AT RBA 00000C1C0D82, LRSN BB8E6373A5DF, PRIOR
CHECKPOINT RBA 00000003D0A5
-P870 SET LOG RESUME
DSNJ373I  -P870 DSNJC09A UPDATE ACTIVITY HAS BEEN RESUMED FOR P870
-P872 SET LOG RESUME
DSNJ373I  -P872 DSNJC09A UPDATE ACTIVITY HAS BEEN RESUMED FOR P872
```

Note: In this scenario we use the SET LOG SUSPEND command to easily retrieve a valid LRSN that we will use later as the log truncation point. You may decide to use a similar technique if you know *a priori* the point to which you may have to recover (for example, the beginning of a batch window). However, in most cases, you will have to determine the RBA/LRSN *a posteriori*, which is a more complex operation. For more information on how to find the RBA/LRSN for SYSPITR, refer to 2.5.2, “Choosing an RBA value for SYSPITR” on page 36.

We run some more workload on the system:

1. SQL activity:
   a. P872 – Insert 80,000 rows into table SAPR3.SWD_TEXT (total 180,000 rows)
2. SAP activity:
   a. Create and generate another ABAP program

19.5.3 Cleaning up before recovery

Now, we want to restore the entire data sharing group to the prior point-in-time that we have prepared in the previous step. But before we can do so, we need to clean up the environment.

Stop DB2

All the DB2 members must be stopped.

Delete the DB2 structures

We must delete the structures belonging to the data sharing group DSNP870.

First, we enter the commands shown in Example 19-6 to force the connections off the group buffer pools and LOCK structures (connections for the SCA are not held at termination, so there are no SCA connections to force off).

Example 19-6 Clean up the DB2 structures (1/2)

```
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNP870_GBP0
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNP870_GBP1
...
SETXCF FORCE,CONNECTION,CONNAME=ALL,STRNAME=DSNP870_LOCK1
```

Then, we enter the commands shown in Example 19-7 to delete all the structures.

Example 19-7 Clean up the DB2 structures (2/2)

```
SETXCF FORCE,STRUCTURE,STRNAME=DSNP870_GBP0
SETXCF FORCE,STRUCTURE,STRNAME=DSNP870_GBP1
...
SETXCF FORCE,STRUCTURE,STRNAME=DSNP870_LOCK1
SETXCF FORCE,STRUCTURE,STRNAME=DSNP870_SCA
```

Finally, we verify that the structures have been successfully deleted. Example 19-8 shows the status of the structures – all the structures belonging to the data sharing group DSNP870 must be in status “NOT ALLOCATED”.

Example 19-8 Clean up the structures – Verification

```
D XCF,STR
IXC359I  12.44.10  DISPLAY XCF
```

```
STRNAME            ALLOCATION TIME   STATUS
...
DSNP870_GBP0          --       --    NOT ALLOCATED
DSNP870_GBP1          --       --    NOT ALLOCATED
...
DSNP870_LOCK1         --       --    NOT ALLOCATED
DSNP870_SCA           --       --    NOT ALLOCATED
```

Unallocate the ICF catalogs

We also need to unallocate the ICF catalogs associated with the *data* copy pool, using the MODIFY CATALOG command, as shown in Example 19-9. If an ICF catalog located on one of the volumes of the data copy pool is open, the RESTORE SYSTEM utility will fail.

Example 19-9 MODIFY CATALOG commands

```
F CATALOG,UNALLOCATE(P87VCAT1.USER.CATALOG)
F CATALOG,UNALLOCATE(P88VCAT1.USER.CATALOG)
F CATALOG,UNALLOCATE(P89VCAT1.USER.CATALOG)
```

Clean up any volumes that may have been added since the backup

If volumes have been added to any of the SMS storage groups associated with the data copy pool since the time of the backup, you need to manually clean up these volumes before you execute the RESTORE SYSTEM utility. For more information, refer to 19.6.1, “What if you added volumes between backup time and restore time?” on page 379.

19.5.4 Recovering the data to an arbitrary prior point-in-time

Now that the environment has been cleaned up, we can restore the entire data sharing group to the prior point-in-time.

Create the SYSPITR conditional restart control record (CRCR)

Since both members were active at the moment of failure, we must create a SYSPITR CRCR record for each member. We use the Change Log Inventory utility (DSNJU003) as shown in Example 19-10. For each member, we specify the log truncation point established previously.

Example 19-10 Change Log Inventory utility (DSNJU003) – SYSPITR CRCR record

```
//STEP1    EXEC PGM=DSNJU003
//SYSUT1   DD   DISP=SHR,DSN=DSNP870.P870.BSDS01
//SYSUT2   DD   DISP=SHR,DSN=DSNP870.P870.BSDS02
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  CRESTART CREATE,SYSPITR=BB8E6368132F
//*
//STEP2    EXEC PGM=DSNJU003
//SYSUT1   DD   DISP=SHR,DSN=DSNP870.P872.BSDS01
//SYSUT2   DD   DISP=SHR,DSN=DSNP870.P872.BSDS02
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  CRESTART CREATE,SYSPITR=BB8E6368132F
```

Example 19-11 is an extract of the Print Log Map utility (DSNJU004) showing the “Conditional Restart Control Record” section of one BSDS. The ENDLRSN field contains the log truncation point. In data sharing the RBLP LRSN is determined by taking minimum of all member level RBLP values.

Example 19-11 Print Log Map utility (DSNJU004) – BSDS content after SYSPITR CRCR

```
                 CONDITIONAL RESTART CONTROL RECORD
                      22:29:05 JULY 22, 2004
 **** ACTIVE CRCR RECORD ****
 CRCR IDENTIFIER  0001
      USE COUNT   0
      RECORD STATUS
          CRCR ACTIVE
          CRCR NOT USED
      PROCESSING STATUS
          FORWARD = YES
          BACKOUT = YES
      SYSPITR SYSTEM LEVEL RECOVERY MODE RESTART
      STARTRBA                        NOT SPECIFIED
      ENDRBA                          NOT SPECIFIED
      ENDLRSN                         BB8E6368132F
      EARLIEST REQUESTED RBA          000000000000
      FIRST LOG RECORD RBA            000000000000
      ORIGINAL CHECKPOINT RBA         000000000000
      NEW CHECKPOINT RBA (CHKPTRBA)   001250B61C08
      END CHECKPOINT RBA              001250B79066
      CRCR CREATED            22:28:20 JULY 22, 2004
      TIME OF CHECKPOINT      22:02:54 JULY 22, 2004
      RESTART PROGRESS                STARTED     ENDED
                                      =======     =====
          CURRENT STATUS REBUILD        NO         NO
          FORWARD RECOVERY PHASE        NO         NO
          BACKOUT RECOVERY PHASE        NO         NO
 ****
```

Restart DB2

We restart the two DB2 members using the -START DB2 command, as shown in Example 19-12. The numbers in the figure correspond to the numbers listed here. DB2 is implicitly started with ACCESS(MAINT) (5) and DEFER ALL. Each member issues a message (DSNJ245I) to indicate that the log will be truncated (1). Once we reply Y to continue, the restart is initiated (2). DB2 does not go through the forward log recovery phase (except for acquiring lock for indoubt URs) – all committed changes and database write operations will be processed during the log apply phase of the RESTORE SYSTEM utility. DB2 backs out the inflight URs, including the long running URs. Postponed abort URs are treated as inflight URs and are completely backed out.

After restart, DB2 is in system recover pending mode (4). At this point, we can only run the RESTORE SYSTEM utility.

Example 19-12 DB2 restart with active SYSPITR CRCR

```
-P870 STA DB2
-P872 STA DB2
...
*38 DSNJ245I  -P870 CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT      (1)
LRSN BB8E6368132F.  REPLY Y TO CONTINUE, N TO CANCEL
*39 DSNJ245I  -P872 CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT
LRSN BB8E6368132F.  REPLY Y TO CONTINUE, N TO CANCEL
...
DSNR001I  -P870 RESTART INITIATED                                           (2)
DSNR003I  -P870 RESTART...PRIOR CHECKPOINT RBA=001250B61C08
DSNR004I  -P870 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
DSNR007I  -P870 RESTART...STATUS TABLE
```

```
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
I RRSAF    032095130038 ICLIRUN  FP8762O  F 00124EFE86DA 204 15:02:45
...
DSNR001I  -P872 RESTART INITIATED
DSNR003I  -P872 RESTART...PRIOR CHECKPOINT RBA=00000003D0A5
DSNR004I  -P872 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
...
DSNR005I  -P870 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
...
DSNR006I  -P870 RESTART...COUNTS AFTER BACKWARD
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR050I  -P870 DSNRRPRC DB2 STARTED IN SYSTEM RECOVER PENDING MODE      (4)
DSNR002I  -P870 RESTART COMPLETED
DSNY014I  -P870 DSNYSTRT DB2 WAS STARTED WITH ACCESS(MAINT)              (5)
...
DSN9022I  -P870 DSNYASCP 'STA DB2' NORMAL COMPLETION
```

Run the RESTORE SYSTEM utility

We execute the RESTORE SYSTEM utility on member P872, as shown in Example 19-13. Note that the utility can be started on any member of the data sharing group, but only on one member. If the utility terminates and must be restarted, it can only be restarted on the member on which it was originally executed.

Restriction: You must have INSTALL SYSADM authority to run RESTORE SYSTEM.

Example 19-13 RESTORE SYSTEM utility

```
//RESTORE EXEC DSNUPROC,PARM='P872,RESTORE'
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  RESTORE SYSTEM
//
```

Example 19-14 shows some of the messages that are sent to the system log during the processing of the RESTORE SYSTEM utility. The utility uses the information contained in the BSDS to identify the backup version that needs to be recovered, and calls the DFSMShsm Fast Replication function to restore the correct *data* copy pool, that is, the last valid version before the truncation point (1). Once the data volumes are restored, the RESTORE SYSTEM utility retrieves the RBLP from DBD01. It starts scanning the log from that point, and applies log records until the log truncation point. Messages (DSNI040I) are issued periodically to indicate the recovery status (2).

Example 19-14 Messages sent to the console during RESTORE SYSTEM

```
$HASP373 PITRES2  STARTED - INIT 1    - CLASS A - SYS SAP6
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$P870$DB, AT 15:37:43 ON 2004/07/22
$HASP100 IEESYSAS ON STCINRDR
$HASP373 IEESYSAS STARTED
ARC1805I THE FOLLOWING 00022 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST REPLICATION RECOVERY
ARC1805I (CONT.) OF COPY POOL DSN$P870$DB
ARC1805I (CONT.) P870SC
```

```
ARC1805I (CONT.) P87001
ARC1805I (CONT.) P87002
...
ARC1805I (CONT.) P89004
ARC1805I (CONT.) P89005
ARC1805I (CONT.) P89006
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR                    (1)
ARC1802I (CONT.) COPY POOL DSN$P870$DB, AT 15:38:00 ON 2004/07/22,
ARC1802I (CONT.) FUNCTION RC=0000, MAXIMUM VOLUME RC=0000
IEF196I IEF237I 6E20 ALLOCATED TO SYS00018
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY                         (2)
PROCESSING LOG RANGE FROM LRSN BB8E502C0AC4 TO BB8E67483EAE
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY
PROCESSING LOG RANGE FROM LRSN BB8E58E4276E TO BB8E67483EAE
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY
PROCESSING LOG RANGE FROM LRSN BB8E59E118A7 TO BB8E67483EAE
...
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY
PROCESSING LOG RANGE FROM LRSN BB8E5F96C93E TO BB8E67483EAE
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY
PROCESSING LOG RANGE FROM LRSN BB8E5F9B778E TO BB8E67483EAE
-                                       --TIMINGS (MINS.)--
 ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
   PAGE   SWAP    VIO SWAPS
-PITRES2           RESTORE   0004    507   1.21    .00    6.4  7179K   0
      0      0      0     0
ESL02 PITRES2             RESTORE   DSNUTILB  SYSADM   0004
-PITRES2  ENDED.  NAME-                    TOTAL CPU TIME=  1.21  TOTAL
 ELAPSED TIME=   6.4
$HASP395 PITRES2  ENDED
```

Example 19-15 shows the output of the RESTORE SYSTEM utility. At the end of the log apply phase, an informational message (DSNU1629I) is issued to indicate that objects have be put in RECP, RBLP, PSRBD or LPL. It can be the result of one of the following situations:

- All objects involved in LOG NO events are marked RECP or RBDP.
- All indexes involved in LOAD LOG YES operations are marked RBDP/RECP/PSRBD (even for LOAD LOG YES operation, DB2 does not log index updates).
- If an index was created on a non-empty table between the time of the backup and the PIT recovery point, it is marked RBDP/RECP/PSRBD.
- If a CREATE TABLESPACE, CREATE INDEX, or data set extension has failed, DB2 may put pages of the objects involved in the logical page list (LPL).

Example 19-15 RESTORE SYSTEM utility – Output

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RESTORE
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RESTORE SYSTEM
DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BB8E5142855EF907BB8E502C0AC4'.
DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BB8E5142855EF907BB8E502C0AC4'
                       ELAPSED TIME = 00:00:16.
DSNU1604I -P872 DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT = X'BB8E502C0AC4'.
DSNU1629I -P872 DSNUVARL - DB2 PUT ONE OR MORE OBJECTS INTO THE RECOVER-PENDING STATE,
                       THE REBUILD-PENDING STATE, OR THE LOGICAL PAGE LIST DURING THE LOG APPLY PHASE.
```

```
DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME = 00:06:11
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

Stop DB2

After the utility ends successfully, we stop all the active DB2 members. This will reset the system recover-pending status.

Complete the recovery

We restart all the members and we need to manually recover the last objects.

We use the DISPLAY UTIL command, as shown in Example 19-16, to see if any utilities are running. During our test, no utility was running at the time chosen for the PITR. Otherwise, we would have had to terminate the active utilities.

Example 19-16 DISPLAY UTIL command

```
-P872 DIS UTIL(*)
DSNU112I  -P872 DSNUGDIS - NO AUTHORIZED UTILITY FOUND FOR UTILID = *
DSN9022I  -P872 DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

We execute the display command shown in Example 19-17 to check for restricted objects. The table space LCABNT is in recover-pending (RECP) state and the index LCABNTH0 is in rebuild-pending (RBDP) state — a consequence of the online REORG.

Example 19-17 DISPLAY DATABASE

```
-P872 DIS DB(*) SP(*) LIMIT(*) RESTRICT
DSNT360I  -P872 ***********************************
DSNT361I  -P872 *  DISPLAY DATABASE SUMMARY
                *    RESTRICTED
DSNT360I  -P872 ***********************************
DSNT362I  -P872     DATABASE = A230X999  STATUS = RW
                   DBD LENGTH = 125168
DSNT397I  -P872
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
LCABNT   TS         RW,RECP
LCABNTH0 IX         RW,RBDP
******* DISPLAY OF DATABASE A230X999 ENDED      **********************
DSNT360I  -P872 ***********************************
DSNT362I  -P872     DATABASE = DB2OLR7  STATUS = RW
                   DBD LENGTH = 4028
DSNT397I  -P872
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
XOLRUMAP IX         RW,RBDP
******* DISPLAY OF DATABASE DB2OLR7  ENDED      **********************
DSN9022I  -P872 DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

We use the RECOVER utility to recover the LCABNT table space, and the REBUILD INDEX utility to rebuild the LCABNTH0 and XOLRUMAP indexes.

Validate that recovery was successful

Finally, we validate that the recovery was successful by running some queries against the tables SAPR3.SWD_TEXT and SAPR3.SREPOVARI to verify that we have the expected number of rows (respectively 100,000 and 30,000), and we check that the last ABAP program that we created is not in the system. During our test, all the validation checks were successful.

19.6 Considerations

In this section, we discuss special considerations that must be taken into account when using system point-in-time recovery.

19.6.1 What if you added volumes between backup time and restore time?

When one or more volumes are added to a storage group within the copy pool, DFSMShsm automatically includes these new volumes to any subsequent backup versions. However, if a recovery must be performed from a version that does not contain the new volumes, and data sets have been allocated on these volumes, then, before recovery, these volumes contain unusable data (because not reflected in the ICF catalog at backup time) that should be deleted to speed up the restart process.

For new user-defined objects created after the CopyPool backup or, more generally, if the RESTORE SYSTEM is unable to locate disk space for create/extend, then those objects will be placed in LPL. In this case, DSNPxxxx messages will be issued on the system console.

Now, let us run some tests.

Suppose that the storage group containing the DB2 catalog and directory, and the user databases has three volumes – DB8X00, DB8X01 and DB8X04. We run the BACKUP SYSTEM utility – the backup version includes the three volumes. Then we create new DB2 objects, and run out of space. So we add a fourth volume (MHL2A0) to the storage group. The new objects (table space and indexes) are allocated on volume MHL2A0. Later on, we find out that an error occurred and we decide to recover the entire system.

We examine two cases: a recovery to the last backup version, and a recovery to an arbitrary point-in-time.

First case – Recovery to the last backup version

The recovery is performed using FRRECOV commands. During the recovery, the volumes DB8X00, DB8X01 and DB8X04 are recovered to the point of the Fast Replication. Since volume MHL2A0 was not a part of the Fast Replication backup, it is not processed during the recovery. DFDSMShsm sends a warning message (ARC1810I) for each volume that fits this scenario, as shown in Example 19-18.

Example 19-18 Added volumes – Messages sent by DFSMShsm

```
ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION RECOVERY OF COPY POOL
ARC1805I (CONT.) DSN$DB8X$DB
ARC1805I (CONT.) DB8X04
ARC1805I (CONT.) DB8X00
ARC1805I (CONT.) DB8X01
ARC1810I THE FOLLOWING VOLUMES WERE NOT DEFINED TO
ARC1810I (CONT.) COPY POOL DSN$DB8X$DB WHEN THIS
ARC1810I (CONT.) BACKUP VERSION WAS CREATED
ARC1810I (CONT.) MHL2A0
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR 877
ARC1802I (CONT.) COPY POOL DSN$DB8X$DB, AT 14:57:07
ARC1802I (CONT.) ON 2004/07/22, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

Because the ICF catalog has been recovered along with the data, volume MHL2A0 contains data sets that are no longer cataloged. These objects are no longer valid and need to be manually deleted by the user.

> **Important:** *After* running the FRRECOV command, you should check for ARC1810I messages indicating that volumes have been added between the time of the backup and the time of the restore. It is your responsibility to clean up these volumes (for example, using the ICKDSF INIT command).

Second case – Recovery to an arbitrary prior point-in-time

The recovery is performed using the RESTORE SYSTEM utility. The DB2 utility calls the Fast Replication services to perform the recovery of the volumes (DB8X00, DB8X01 and DB8X04) and, when it receives a successful return code from DFSMShsm, automatically starts the log apply phase. During this process, DB2 tries to create the objects again, but is likely to eventually run out of space, because of the space occupied by orphan data sets, or getting an error for duplicate data set name.

For each object, SMS looks in the storage group for a volume where it can allocate the data set. It will reject the volume MHL2A0 (even if there is enough space), because a data set with a duplicate name exists on this volume.

- If there is enough space on the other volumes (DB8X00, DB8X01 or DB8X04), SMS will allocate the data set there. But there will still be an orphan data set on MHL2A0.
- If there is not enough space on the other volumes, the RESTORE utility will provide the output shown in Example 19-19. The Return Code is 4 and the objects will show up in RECP or RBDP mode.

Example 19-19 Output from the Restore utility

```
1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RESTSYSF
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  RESTORE SYSTEM
 DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                        COPYPOOL = DSN$DB8X$DB
                        TOKEN = X'C4C2F8E7BBAD92D94634794800008C06D090'.
 DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                        COPYPOOL = DSN$DB8X$DB
                        TOKEN = X'C4C2F8E7BBAD92D94634794800008C06D090'
                        ELAPSED TIME = 00:00:01.
 DSNU1604I -DB8X DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT
 DSNU1629I -DB8X DSNUVARL - DB2 PUT ONE OR MORE OBJECTS INTO THE RECOVER-PENDING
                        OR THE LOGICAL PAGE LIST DURING THE LOG APPLY PHASE.
 DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME =
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=4
```

- In order to understand the reason for it, it is a good idea to also look at the System Log where SMS will return the error messages shown in Example 19-20.

Example 19-20 Added volumes – System log from RESTORE SYSTEM utility

```
$HASP373 RESTSYSF STARTED - INIT 2    - CLASS A - SYS SC64
IEF403I RESTSYSF - STARTED - ASID=001C - SC64
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR 802
ARC1801I (CONT.) COPY POOL DSN$DB8X$DB, AT 14:12:27
ARC1801I (CONT.) ON 2004/08/16
$HASP100 IEESYSAS ON STCINRDR
$HASP373 IEESYSAS STARTED
IEF403I IEESYSAS - STARTED - ASID=042B - SC64
```

```
ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE 806
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION RECOVERY OF COPY POOL
ARC1805I (CONT.) DSN$DB8X$DB
ARC1805I (CONT.) DB8X04
ARC1805I (CONT.) DB8X00
ARC1805I (CONT.) DB8X01
ARC1810I THE FOLLOWING VOLUMES WERE NOT DEFINED TO 810
ARC1810I (CONT.) COPY POOL DSN$DB8X$DB WHEN THIS
ARC1810I (CONT.) BACKUP VERSION WAS CREATED
ARC1810I (CONT.) MHL2A0
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR 812
ARC1802I (CONT.) COPY POOL DSN$DB8X$DB, AT 14:12:28
ARC1802I (CONT.) ON 2004/08/16, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
IEF196I IEF237I 8011 ALLOCATED TO SYS00022

DSNI040I  -DB8X DSNIRSTR RESTORE SYSTEM UTILITY 815
PROCESSING LOG RANGE FROM RBA 00008C06D090 TO 00008C08742C

IGD17272I VOLUME SELECTION HAS FAILED FOR INSUFFICIENT SPACE FOR 818
DATA SET DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
JOBNAME (DB8XDBM1) STEPNAME (DB8XDBM1)
PROGNAME (DSNYASCP) DDNAME (N/A     )
REQUESTED SPACE QUANTITY = 144000 KB
STORCLAS (DB8XD) MGMTCLAS (MCDB22) DATACLAS (        )
STORGRPS (DB8XD  )
DSNP009I  -DB8X THE FOLLOWING ERROR MESSAGES WERE 819
RECEIVED FOR DEFINE CLUSTER ON DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
IGD01009I MC ACS GETS CONTROL &ACSENVIR=ALLOC
IGD01010I SG ACS GETS CONTROL &ACSENVIR=ALLOC
IGD17273I ALLOCATION HAS FAILED FOR ALL VOLUMES SELECTED FOR DATA SET
DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:DB8XD
IGD17279I 3 VOLUMES WERE REJECTED BECAUSE OF A DADSM FAILURE
IGD17279I 2 VOLUMES WERE REJECTED BECAUSE THEY DID NOT HAVE
SUFFICIENT SPACE
IGD17219I UNABLE TO CONTINUE DEFINE OF DATA SET
DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
IDC3020I UNABLE TO ALLOCATE SPACE ON USER VOLUME
IDC3007I ** VSAM CATALOG RETURN CODE IS 68
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12

DSNP010I  -DB8X END OF MESSAGES. 820
   CONNECTION-ID=UTILITY
   CORRELATION-ID=RESTSYSF
   CORRELATION-ID=RESTSYSF
   LUW-ID=*
DSNP016I  -DB8X DSNPCILR - CREATE FAILED FOR 821
           DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001.
           RC=00D70025
           CONNECTION-ID=UTILITY, CORRELATION-ID=RESTSYSF,
           LUW-ID=*
DSNP012I  -DB8X DSNPCNP0 - ERROR IN VSAM CATALOG 822
LOCATE FUNCTION FOR DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
           CTLGRC=AAAAAA08
           CTLGRSN=AAAAAA08
```

```
           CONNECTION-ID=UTILITY, CORRELATION-ID=RESTSYSF,
           LUW-ID=*
IKJ56228I DATA SET DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001 NOT IN
CATALOG OR CATALOG CAN NOT BE ACCESSED
DSNB207I  -DB8X DYNAMIC ALLOCATION OF DATA SET 824
FAILED. REASON=17080002.
DSNAME=DB8XD.DSNDBC.PAOLODB.TS1000.I0001.A001
-                                        --TIMINGS (MINS.)--
   ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV
PG   PAGE   SWAP    VIO SWAPS STEPNO
-RESTSYSF RESTSYSF DSNUPROC    04    502    .00    .00    .10   1548
 0      0      0      0     0     1
IEF404I RESTSYSF - ENDED - ASID=001C - SC64
-RESTSYSF ENDED.  NAME-RESTSYSF             TOTAL CPU TIME=   .00
TOTAL ELAPSED TIME=   .10
$HASP395 RESTSYSF ENDED
$HASP309 INIT 2    INACTIVE ******** C=ABCDE
SE '14.12.33 JOB04718 $HASP165 RESTSYSF ENDED AT WTSCPLX2  MAXCC=4',
LOGON,USER=(PAOLOR1)
```

Important: We conducted these tests using the following service level: RSU0402 plus APARs PQ91099, PQ92187, and PQ89297.

Attention: *Before* running the RESTORE SYSTEM command, you should determine if volumes have been added between the time of the backup and the time of the restore. It is your responsibility to clean up these volumes (for example, using the ICKDSF INIT command). One way could be to use the DFSMShsm LIST COPYPOOL command and compare the output with the list of the volumes contained in the storage groups associated with the data copy pool.

A way to delete VSAM uncataloged data sets is by using the JCL in Example 19-21.

Example 19-21 Deleting a specific DB2 orphan data set

```
//DDLVVR  JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//        NOTIFY=&SYSUID,REGION=0M,TIME=30
//DEL EXEC  PGM=IDCAMS
//*THIS JOB DELETES VSAM DSN AND OTHER UNCATLOG DSN
//*IF IT'S UNCAT. FILE   NVR IS NEEDED.
//*IF IT'S UNCAT VSAM DATA COMPONENT VVR IS NEEDED
//*******************************************************
//SYSPRINT DD  SYSOUT=*
//SOURCE   DD  VOL=SER=MHL2A0,UNIT=3390,DISP=SHR,
//             DSN=SYS1.VVDS.VMHL2A0
//SYSIN  DD  *
 DELETE DB8XD.DSNDBD.PAOLODB.TS1000.I0001.A001 FILE(SOURCE) VVR
```

19.6.2 Restart checkpoint not found

DB2 records 100 system checkpoints in the BSDS. For example, if you take a system checkpoint every 2 minutes, this means that the first system checkpoint recorded in the BSDS was taken 3 hours and 20 minutes ago. If the log truncation point used to create the SYSPITR CRCR record is anterior to this time, the Change Log Inventory utility (DSNJU003) will fail, with the error messages shown in Example 19-22.

Example 19-22 Restart checkpoint not found – Error messages

```
DSNJ407E  DSNRJFCK NO VALID CHECKPOINT RBA FOUND
DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0005, DDNAME = SYSUT1
DSNJ221I  PREVIOUS ERROR CAUSED CRESTART OPERATION TO BE BYPASSED
DSNJ201I  DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING WAS UNSUCCESSFUL
```

In this case, you must manually get a valid system checkpoint RBA using the DSN1LOGP utility for each member of the data sharing group. Once you have found such an RBA, you can create a SYSPITR CRCR record as follows (see Example 19-23).

Example 19-23 Restart checkpoint not found – Creating a SYSPITR CRCR record

```
CRESTART CREATE,SYSPITR=<log_truncation_point>,CHKPTRBA=<system_checkpoint>
```

The <log_truncation_point> is the RBA (if non-data sharing) or LSRN (data sharing) to which you want to recover. The <system_checkpoint> is the begin checkpoint RBA that you have found using DSN1LOGP. In a data sharing environment, the <log_truncation_point> must be the same for each member; however, the <system_checkpoint> RBA will be different for each member. This can be a time consuming task, especially for a multi-member data sharing group, and can greatly prolong the PIT or disaster recovery time.

Tip: The very recent APAR PQ93548 helps in this area by the following changes:

1. The DSNJU003 execution will not fail, it will return RC=4 and the message:

   ```
   DSNJ407I  DSNRJFCK WARNING - NO VALID CHECKPOINT RBA FOUND
             LOG WILL BE SCANNED AT RESTART
   ```

2. At restart, DB2 will confirm that it is going to read the log backwards from the log truncation point until a complete checkpoint is found, with the message:

   ```
   DSNR054I RESTART...BEGIN SCAN FOR CHECKPOINT
   ```

Refer to the APAR text for details.

19.6.3 Fast Log Apply during RESTORE

The final tests for Restore with FLA were performed at the SAP Integration Lab.

In addition to the DB2 for z/OS V8 PUT 0406 level, the following fixes have been installed:

UQ90843 UQ89848 UQ89861 UQ90454 UQ90150 UQ91252 UQ91525 UQ91259 UQ91554 UQ91022 UQ90863 UQ90794 UQ91801 AQ92616 AQ90118 AQ92529 AQ92187 **AQ93156** AQ92958 AQ93042 AQ91795 AQ88776 CQ90217 BQ90218 BQ87377 AQ91040 **AQ95164**

With this maintenance (specifically, the PTFs in bold), you will activate Fast Log Apply during the RESTORE utility.

FLA buffer

If you specify LOGAPSTG as 0, FLA is disabled. Any non zero value will activate FLA and that value will be used as specified during *Recovery utility* executions. For *Restore utility* executions, if LOGAPSTG is not 0, since no other DB2 activity is taking place, DB2 will allocate 500 MB.

As mentioned, the message issued to the console will show the value in HEX:

```
DSNI029I  -P870 DSNIDEFR THE FAST LOG APPLY PROCESS 074
IS ACTIVE AND THE AVAILABLE P870DBM1 STORAGE IS 1F400000 BYTES
```

Only if otherwise storage constrained, DB2 will allocate a smaller buffer by decreasing the default value from 500 to 250, then 125, and 75 MB.

Multi-tasking

PTF UQ92067 for PQ89297 is also recommended in order to set the priority of the subtasks to the same value assigned to utility. DB2 will also try to allocate up to 99 tasks for concurrent executions on multiple DB2 objects. The priority assigned to these tasks is derived directly from the priority assigned by you to the batch utility job.

Sample RESTORE FLA executions

The RESTORE job output does not mention that FLA was active, only that Log Apply took place, see Example 19-24.

Example 19-24 RESTORE job output

```
1                       J E S 2  J O B  L O G  --  S Y S T E M  S A P 6  --  N
0
 16.43.15 JOB07315 ---- TUESDAY,   24 AUG 2004 ----
 16.43.15 JOB07315  ICH70001I SYSADM   LAST ACCESS AT 16:32:29 ON TUESDAY, AUGUS
 16.43.15 JOB07315  $HASP373 PITRES2  STARTED - INIT 1    - CLASS A - SYS SAP6
 17.09.09 JOB07315  -                                        --TIMINGS (MINS.)-
 17.09.09 JOB07315  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOC
 17.09.09 JOB07315  -PITRES2           RESTORE   0004    220   3.28    .00   25.
 17.09.09 JOB07315  -PITRES2  ENDED.  NAME-                    TOTAL CPU TIME=
 17.09.09 JOB07315  $HASP395 PITRES2  ENDED


IEF375I  JOB/PITRES2 /START 2004237.1643
IEF376I JOB/PITRES2 /STOP 2004237.1709 CPU  3MIN 17.13SEC SRB  0MIN 00.10SIEF375I JOB/PITRES2 /START 2004237.1643
IEF376I  JOB/PITRES2 /STOP  2004237.1709 CPU    3MIN 17.13SEC SRB    0MIN 00.10SEC
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RESTORE
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  RESTORE SYSTEM
DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BBB7AC91D252A504BBB788D18DD4'.
DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$P870$DB
                       TOKEN = X'D7F8F7F0BBB7AC91D252A504BBB788D18DD4'
                       ELAPSED TIME = 00:00:18.
DSNU1604I -P870 DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT = X'BBB788D18DD4'.
DSNU1629I -P870 DSNUVARL - DB2 PUT ONE OR MORE OBJECTS INTO THE RECOVER-PENDING STATE, THE REBUILD-PENDING STATE,
                       OR THE LOGICAL PAGE LIST DURING THE LOG APPLY PHASE.
DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME = 00:25:34.
```

The Fast Log Apply execution is confirmed by a message in your system log similar to what we show in Example 19-25.

Example 19-25 System log for RESTORE with FLA

```
$HASP373 PITRES2  STARTED - INIT 1    - CLASS A - SYS SAP6
DSNI029I  -P872 DSNIDEFR THE FAST LOG APPLY PROCESS
IS ACTIVE AND THE AVAILABLE P872DBM1 STORAGE IS 1F000000 BYTES
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY
PROCESSING LOG RANGE FROM LRSN BBBB7260C3DE TO BBC2DAAD56E4
```

At the end of the RESTORE job, DB2 issues some messages to the MVS console with statistics about the FLA execution. See Example 19-26.

Example 19-26 FLA informational messages to the console

```
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY 634
PROCESSING LOG RANGE FROM LRSN BBBCDA9B9DFC TO BBC99E2C1012
DSNI040I  -P872 DSNIRSTR RESTORE SYSTEM UTILITY 635
PROCESSING LOG RANGE FROM LRSN BBBCDC5FA53B TO BBC99E2C1012
DSNI028I  -P872 DSNIFLAF THE NUMBER OF QUALIFIED
LOG RECORDS READ DURING THE FAST LOG APPLY PROCESS IS 13803142
AND THE NUMBER OF FAST LOG APPLY BUFFERS PROCESSED ARE 209
-                                          --TIMINGS (MINS.)--
 ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
   PAGE   SWAP    VIO SWAPS
-PITRES2           RESTORE   0004    492  10.38    .01  194.0 61273K   0
      0      0      0     0
ESL02 PITRES2             RESTORE   DSNUTILB  SYSADM   0004
-PITRES2  ENDED.  NAME-                     TOTAL CPU TIME= 10.38  TOTAL
 ELAPSED TIME= 194.0
$HASP395 PITRES2  ENDED
```

In Example 19-27 we show a full sequence of MVS console messages for the execution of the Restore utility, which is summarized in Table 19-1.

Example 19-27 MVS console - RESTORE SYSTEM messages with FLA

```
17:33:52.16 STC00983 00000010  DSNI029I  *DJ1G DSNIDEFR THE FAST LOG APPLY PROCESS 400
                 400 00000010  IS ACTIVE AND THE AVAILABLE DJ1GDBM1 STORAGE IS 1F400000 BYTES
17:33:52.18 STC00983 00000010  DSNI040I  *DJ1G DSNIRSTR RESTORE SYSTEM UTILITY 401
                 401 00000010  PROCESSING LOG RANGE FROM RBA 00000EDE9AE3 TO 000043E9F41A
17:33:52.21          00000210  IEF196I IEF237I 8B0D ALLOCATED TO SYS00017
17:33:52.60 STC00989 00000010  IEF352I ADDRESS SPACE UNAVAILABLE
17:33:52.60 STC00989 00000010  $HASP395 IEESYSAS ENDED
17:33:52.79 STC00988 00000010  IEF352I ADDRESS SPACE UNAVAILABLE
17:33:52.79 STC00988 00000010  $HASP395 IEESYSAS ENDED
17:33:53.05          00000210  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0195.
17:33:53.05          00000210  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=019C.
17:35:27.12 STC00983 00000010  DSNI040I  *DJ1G DSNIRSTR RESTORE SYSTEM UTILITY 409
                 409 00000010  PROCESSING LOG RANGE FROM RBA 000010E770D5 TO 000043E9F41A
17:35:40.60 STC00983 00000010  DSNI040I  *DJ1G DSNIRSTR RESTORE SYSTEM UTILITY 410
                 410 00000010  PROCESSING LOG RANGE FROM RBA 000012D430A0 TO 000043E9F41A
17:35:54.84 STC00983 00000010  DSNI040I  *DJ1G DSNIRSTR RESTORE SYSTEM UTILITY 411
                 411 00000010  PROCESSING LOG RANGE FROM RBA 0000156D6548 TO 000043E9F41A
17:36:56.07 STC00983 00000010  DSNI040I  *DJ1G DSNIRSTR RESTORE SYSTEM UTILITY 412
                 412 00000010  PROCESSING LOG RANGE FROM RBA 000026D15000 TO 000043E9F41A
17:37:47.08 STC00983 00000010  DSNI040I  *DJ1G DSNIRSTR RESTORE SYSTEM UTILITY 413
                 413 00000010  PROCESSING LOG RANGE FROM RBA 000037D50238 TO 000043E9F41A
17:38:14.19 STC00983 00000010  DSNI028I  *DJ1G DSNIFLAF THE NUMBER OF QUALIFIED 414
                 414 00000010  LOG RECORDS READ DURING THE FAST LOG APPLY PROCESS IS 3772081
                 414 00000010  AND THE NUMBER OF FAST LOG APPLY BUFFERS PROCESSED ARE 17
17:38:14.38 JOB00986 00000010  SMF000I  RSTRUTIL    RESTOR      DSNUTILB    0004
17:38:14.38 JOB00986 00000010  $HASP395 RSTRUTIL ENDED
```

In the MVS log we have set in bold the relevant FLA information.

- Message DSNI029I indicates the start of the utility, the usage of Fast Log Apply, and the amount of storage utilized. for reading and applying the log records. The value 1F400000 BYTES is the HEX correspondent to the system assumed default of 500 MB, based on our non zero definition in DSNZPARM LOGAPSTG (100 MB).
- Message DSNI040I issued repetitively keeps you informed of the RBA progress in applying the log records.

- Message DSNI028I indicates the end of the utility with the total of the log records processed and the buffers used.

FLA impact

A performance test was implemented, after applying a fix for PQ95164, with the objective of comparing Restore runs with and without FLA by way of setting a non zero value for LOGAPSTG (to enable FLA with 500 MB default) and 0 (to disable FLA).

The chosen workload was the IBM Relational Warehouse Workload (IRWW). Query results after the Restore were also compared for verification.

Table 19-1 summarizes the measurements.

Table 19-1 Restore with and without FLA

Fast Log Apply	Log records processed (million)	Elapsed time (sec.)	Rate (log records per sec.)
Disabled	3.800	1.320	2,900
Enabled	3.772	262	14.400

Without FLA, DB2 processed 3.800 million log records in 1.320 seconds; this corresponds to a process rate of 2,900 log records per second.

With FLA, and the default 500 MB buffer, DB2 processed 3.772 million log records in 262 seconds, about *five times faster*; this corresponds to about 14.400 log records per second.

We have seen that for this workload, which is typical of warehouse applications, Restore with FLA enabled is about five times faster than the normal log apply. Since the functionality of FLA with Restore is the same as Recovery (as well as DB2 restart, and START DATABASE for LPL and GRECP), we can qualitative assume that previous general performance considerations still apply.

FLA was introduced with DB2 V6. The redbook, *DB2 UDB for OS/390 Version 6 Performance Topics*, SG24-5351, contains detailed performance information on its previous use. From a very general point of view, the more sequential was the update processing reflected in the log, the less evident will be the advantage of using the FLA option. Conversely, typical transactional environments, with prevalent random updates, are expected to show improvements even higher than the one measured with IRWW, up to a maximum of 10 times.

19.6.4 Reducing objects recovery status

We have seen that BACKUP SYSTEM is implicitly handling the sequences of Set log suspend - FlashCopy - Set log suspend, saves the RBLP as the oldest system checkpoint or GBP Checkpoint value of all the SSIDs and GBPs, and records the starting point for the Log Apply phase.

However, if an object is being REORG'd **and**, either started after the RBLP, or had updated logs after it, **and** the REORG job had completed before the time the BACKUP SYSTEM starts, then the object would be marked in RECP/RBDP unnecessarily because DB2 is not able to locate the old data set while processing a log for it. The BACKUP SYSTEM initiated system checkpoints and GBP checkpoints derive an RBLP value not close enough to the time when the CopyPool backup starts.

In general, this condition would be rare, because online REORG will most likely take a longer time to complete.

However, it is advisable to avoid running *short* online REORG altogether while taking a system level backup. If short online REORGs cannot be avoided at BACKUP time, it could be advisable to increase the frequency of checkpoints by:

1) Forcing a checkpoint on each member of a data sharing group:

```
-SET LOG LOGLOAD(0)
```

2) And forcing more frequent (such as every minute) group buffer pool checkpoints:

```
-ALT GBPOOL(GBP0) GBPCHKPT(1)
-ALT GBPOOL(GBP1) GBPCHKPT(1)
-ALT GBPOOL(GBP2) GBPCHKPT(1)
-ALT GBPOOL(GBP3) GBPCHKPT(1)
-ALT GBPOOL(GBP4) GBPCHKPT(1)
-ALT GBPOOL(GBP8) GBPCHKPT(1)
-ALT GBPOOL(GBP16) GBPCHKPT(1)
-ALT GBPOOL(GBP40) GBPCHKPT(1)
-ALT GBPOOL(GBP32K) GBPCHKPT(1)
```

In this unlikely situation where online Reorg started after BACKUP SYSTEM and ended sometime before the PITR point (when the “I” objects become “J”), if there are no logs between RBLP and Reorg End against the “I” data set, then the object will not be marked in RECP/RBDP.

The online Reorg will always have logs against “I” until the switch phase.

```
Reorg Start               RBLP       Reorg End          Backup System
|---------------------------|------------|--------------------|-------------------> DB2 log
d/s=I                                    d/s=J
```

So the log processed was for the “I” data sets and since Reorg had ended and the data sets are now “J”, it put the “J” objects in RECP/RBDP.

If there is a log against “I” between RBLP and Reorg End, then a DSNP* message will be issued to indicate that the RESTORE SYSTEM is unable to locate the “I” data set.

20

Restart using tape dump of copy pools

z/OS V1R5 DFSMShsm manages the use of volume level Fast Replication functions, such as the FlashCopy function supported by the IBM TotalStorage Enterprise Storage Server (ESS) disk and the SnapShot function supported by the IBM RAMAC® Virtual Array (RVA) disk. These functions provide point-in-time copy services that can quickly copy data from a source location to a target location. Using the new SMS copy pool construct, you can specify the pool storage groups that you want DFSMShsm to process collectively for Fast Replication.

This chapter shows how to extend your local DFSMShsm Fast Replication backups to a remote location by using tape dump of the copy pool backup storage group.

The following publications should be referenced to complement the information presented in this chapter:

- *DFSMShsm Fast Replication Technical Guide*, SG24-7069
- *z/OS 1.5 DFSMShsm Storage Administration Guide*, SC35-0421
- *z/OS 1.5 DFSMShsm Storage Administration Reference*, SC35-0422

This chapter covers the following topics:

- Extending Fast Replication backups via tape dumps
- Description of the scenarios
- Description of the test environment
- Setting up the SMS environment
- Creating the backup
- Dumping the backup
- Restoring the dump at the recovery system
- Restarting DB2

20.1 Extending Fast Replication backups via tape dumps

z/OS V1R5 DFSMShsm provides enhanced management functions for taking volume level backups. With this capability, a set of SMS storage groups can be defined as a copy pool. All the volumes in this pool are processed collectively, and you can use the DFSMShsm Fast Replication function to create backup versions managed by DFSMShsm. Recovery at the copy pool level (or at the volume level) is also fast and easy to perform. Copy pools and Fast Replication provide a fast, easy-to-use volume level backup and recovery solution, which is exploited by the new DB2 V8 utility BACKUP SYSTEM.

For more information on the DFSMShsm Fast Replication function and how it is used by DB2 V8 new utilities, refer to Chapter 6, “SMS copy pools and DB2 point in time recovery” on page 79.

One benefit to using copy pools and the DFSMShsm Fast Replication function is that you can request a Fast Replication of a set of volumes (using either the BACKUP SYSTEM utility or DFSMShsm commands), follow this backup with a dump of each target volume to produce a dump tape, and this dump tape can be restored directly back to the source volumes or, in the case of a disaster recovery scenario, to the remote site. By doing this, you enable the dump function to be performed without a significant outage.

Restriction: DFSMShsm currently does not support creating a dump copy of a target volume by using either the autodump function or the BACKVOL DUMP command. You must use DFSMSdss or an equivalent product to create a dump copy of a target volume.

20.2 Description of the scenarios

In this section, we summarize the steps necessary to use the DFSMShsm Fast Replication function and create consistent backups of an entire DB2 subsystem or data sharing group. The target volumes are then dumped to tape. The tape can be restored later at the remote site and the DB2 subsystem restarted. The procedure varies depending on whether you are using DB2 data sharing or not.

20.2.1 Non-data sharing

If you are *not* using DB2 data sharing, use the following procedure:

Preparation

1. Define two copy pools:
   - One for DB2 logs and BSDSs (and associated ICF catalog)
   - One for DB2 catalog/directory and user databases (and associated ICF catalogs)
2. Define at least one copy pool backup storage group.
3. Validate the Fast Replication environment.

Backup

If you are using the BACKUP SYSTEM utility, use the following procedure:

1. Use BACKUP SYSTEM FULL to take the system backup.
2. Make tape copies and transport to the remote site.

If you are *not* using the BACKUP SYSTEM utility, use the following procedure:

1. Issue the DB2 command SET LOG SUSPEND to suspend logging and update activity, and to quiesce 32 KB page writes and data set extensions.
2. Use the DFSMShsm command FRBACKUP to copy both the data *and* log copy pools.

```
FRBACKUP COPYPOOL(DSN$locn$DB) NOVTOCENQ TOKEN(token)
FRBACKUP COPYPOOL(DSN$locn$LG) NOVTOCENQ TOKEN(token)
```

3. Issue the DB2 command SET LOG RESUME to resume normal DB2 update activity.
4. Use DFSMSdss to dump the disk copies you just created to tape, then transport this tape to the remote site.

Restore at the remote site and restart

1. Unallocate the ICF catalogs that reside on the target volumes.
2. Use DFSMSdss to restore the tape dump to disks.
3. If using DDF, update LOCATION and LUNAME in the BSDS (Change Log Inventory).
4. Start DB2 by using the local DSNZPARM member (normal restart).
5. Use the DISPLAY UTIL command to see if any utilities are in stopped status. If utilities are displayed, use the TERM UTIL command to end them.
6. Execute the following display commands to verify that no object is in a restricted state:

```
-DIS DB(*) SPACE(*) LIMIT(*) RESTRICT
```

20.2.2 Data sharing

If you are using DB2 data sharing, use the following procedure:

Preparation

1. Define two copy pools:
   - One for DB2 logs and BSDSs of all members (and associated ICF catalog),
   - One for DB2 catalog/directory and user databases (and associated ICF catalogs).
2. Define at least one copy pool backup storage group.
3. Validate the Fast Replication environment.

Backup

If you are using the BACKUP SYSTEM utility, use the following procedure:

1. Use BACKUP SYSTEM FULL to take the system backup.
2. Make tape copies and transport to the remote site.

If you are *not* using the BACKUP SYSTEM utility, use the following procedure:

1. Issue the DB2 command SET LOG SUSPEND to *each member* of the group, in order to suspend logging and update activity, and to quiesce 32 KB page writes and data set extensions.
2. Use the DFSMShsm command FRBACKUP to copy both the data *and* log copy pools.

```
FRBACKUP COPYPOOL(DSN$locn$DB) NOVTOCENQ TOKEN(token)
FRBACKUP COPYPOOL(DSN$locn$LG) NOVTOCENQ TOKEN(token)
```

3. Issue the DB2 command SET LOG RESUME to *each member* of the group, in order to resume normal DB2 update activity.

4. Use DFSMSdss to dump the disk copies you just created to tape, then transport this tape to the remote site.

Restore at the remote site and restart

You must have an identical data sharing group at the recovery site. It must have the same name, the same number of members, and the same member names as the data sharing group at the local site. Before starting with disaster recovery, you should ensure that none of the members at the recovery site is active:

1. Unallocate any ICF catalogs that reside on the target volumes.
2. Use DFSMSdss to restore the tape dump to disks.
3. If using DDF, update LOCATION and LUNAME in the BSDS (Change Log Inventory).
4. Delete all CF structures owned by the group.
5. Start all active members, using the local DSNZPARM members.
6. Execute the following display commands to check for restricted objects:

   ```
   -DIS DB(*) SPACE(*) LIMIT(*) RESTRICT
   ```

7. Execute the GRECP and LPL recovery, which recovers the changed data that were stored in the coupling facility at the time of the backup. If you have LPL and GRECP status conditions for page sets of the DB2 catalog and directory, you must issue the appropriate START DATABASE commands for these page sets before you issue the commands for the other page sets or partitions.
8. Use the DISPLAY UTIL command to see if any utilities are in stopped status. If utilities are displayed, use the TERM UTIL command to end them.

20.3 Description of the test environment

The tests were conducted in a *non-data sharing environment*, using the following hardware and software:

- Hardware:
  - zSeries 990 processor Model B16
  - 1 Logical Partition (LPAR) with 2 dedicated CPUs
  - 2 GB real memory
- Disks:
  - ESS Model 800 with FICON channels – FlashCopy V2
  - ESS Model F20
- Software:
  - z/OS V1R5, service level RSU0402
  - DB2 for z/OS V8

20.4 Setting up the SMS environment

In order to use the DFSMShsm Fast Replication function, the storage system administrator needs to establish the copy pools environment for the DB2 system and the user databases, as described in B.1, “Setting up copy pools for PITR” on page 464.

20.4.1 Preparing for using the Fast Replication function for DB2

If you want to exploit the DFSMShsm Fast Replication function in a DB2 environment, the following requirements should be met:

- All volumes containing DB2 data sets, including the DB2 catalog and directory, the logs, and the BSDS, must be SMS managed.
- The DB2 logs and BSDS should be separated from the rest of the DB2 data sets: they should reside in their own pool of volumes, have their own ICF catalog, and be defined in a separate SMS storage group.
- Since a Fast Replication backup is a physical copy of the data, special care should be taken to ensure that the ICF catalogs stay synchronized with the data. ICF catalogs should be copied with the data that they represent so that both can be recovered to the same point-in-time.
- DFSMS does not verify that the storage groups listed in a copy pool include any associated extend and overflow storage groups. Therefore, if you have implemented extend or overflow storage groups, you must ensure that you include them in the appropriate copy pools.

In our test environment, we defined two separate ICF catalogs:

- UCAT.DB8XL for the DB2 logs, BSDS and system executable libraries.
- UCAT.DB8XD for the DB2 catalog and directory, and the user databases.

Note that we could have separated the DB2 catalog and directory from the user databases by defining additional ICF catalogs, although we chose not to do so for this implementation.

We defined the following SMS storage groups:

- DB8XD (volumes DB8X00 and DB8X01) for the user databases
- DB8XC (volume DB8X04) for the DB2 catalog and directory
- DB8XA (volume DB8X02) for the DB2 archive logs
- DB8XL (volume DB8X03) for the DB2 logs and BSDS and system executable libraries

Example 20-1 shows the filter lists that we used when defining the SMS storage classes in the ACS routines.

Example 20-1 Filter lists for SMS storage classes

```
FILTLIST DB8XC    INCLUDE(DB8XD.DSNDBC.DSNDB06.**,
                          DB8XD.DSNDBC.DSNDB01.**)
FILTLIST DB8XD    INCLUDE(DB8XD.DSNDBC.**)
FILTLIST DB8XA    INCLUDE(DB8XU.ARCHLOG*.**)
FILTLIST DB8XL    INCLUDE(DB8XU.LOGCOPY*.**,
                          DB8XU.BSDS*,
                          DB8XU.**,
                          DB8X8.**)
```

Figure 20-1 illustrates the SMS definitions for the source volumes.

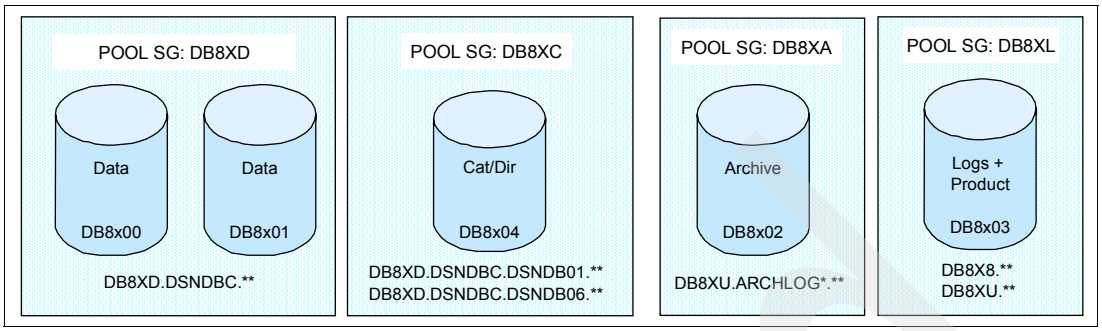


Figure 20-1 SMS definitions for source volumes

20.4.2 Defining the copy pools for DB2 logs and data

Your storage system administrator needs to define the *data* and *log* copy pools and associated source storage groups. The *data* copy pool should contain all volumes that contain the DB2 catalog and directory, all user databases, and all associated ICF catalogs. The *log* copy pool should contain the active logs, the BSDS, and the associated ICF catalog. You may also want to include the system executable libraries in the *log* copy pool.

The naming convention for the copy pools is dictated by the DB2 V8 requirements. In our environment, the following copy pools are defined:

- DSN$DB8X$LG associated to the storage groups DB8XA and DB8XL
- DSN$DB8X$DB associated to the storage groups DB8XC and DB8XD

Figure 20-2 shows the definitions for the first of the two copy pools that we have defined. To display this information, choose option P from the ISMF primary option menu, enter the name of your copy pool, select option 2 and press Enter.

```
                                COPY POOL DISPLAY
Command ===>

CDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name: DSN$DB8X$LG

Description : DB8X LOGS,BSDS,CODE,OTHER

Number of Recoverable DASD Fast Replicate Backup Versions : 2
Storage Group Names :  DB8XA     DB8XL
```

Figure 20-2 Copy pool DSN$DB8X$LG

Figure 20-3 shows the definitions for the second copy pool.

```
                                COPY POOL DISPLAY
Command ===>

CDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name: DSN$DB8X$DB

Description : DB8X DATABASE,CATALOG,DIRECTORY

Number of Recoverable DASD Fast Replicate Backup Versions : 2
Storage Group Names :  DB8XC     DB8XD
```

Figure 20-3 Copy pool DSN$DB8X$DB

Note: When defining a copy pool, you also need to specify the number of backup versions that you want to keep for this copy pool. It is recommended to keep at least two versions (default). We used that value in our implementation.

20.4.3 Defining the copy pool backup storage group

Your storage system administrator now has to define a copy pool backup storage group. This is used to specify the candidate target volumes where the Fast Replication backup versions will be kept.

There is no specific naming convention for the copy pool backup storage group. In our environment, we chose to define a copy pool backup storage group called DB8XCPB, as shown in Figure 20-4.

```
                      COPY POOL BACKUP STORAGE GROUP DEFINE
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : DB8XCPB

To DEFINE Storage Group, Specify:

 Description ==> COPY POOL BACKUP STORAGE GROUP FOR DB8X DATA + LOG
             ==>
```

Figure 20-4 Copy pool backup DB8XCPB

Once the copy pool backup storage group is defined, you need to alter the definition of all the storage groups that you want to associate with the copy pool backup storage group, as shown in Figure 20-5. A storage group can only be associated with one copy pool backup storage group, but many storage groups can be associated with the same copy pool backup storage group.

```
                          POOL STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : DB8XL
To ALTER Storage Group, Specify:
 Description ==> DB8X ACTIVE LOGS, BSDS, AND CLONED LIBRARIES
             ==>
 Auto Migrate . . N  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . N  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . N  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . . . .
                                      Copy Pool Backup SG Name . . . DB8XCPB
 Dump Class . . .                     (1 to 8 characters)
 Dump Class . . .                     Dump Class . . .
 Dump Class . . .                     Dump Class . . .
 Allocation/migration Threshold: High . . 85  (1-99)      Low  . .     (0-99)
 Guaranteed Backup Frequency  . . . . . .          (1 to 9999 or NOLIMIT)

ALTER     SMS Storage Group Status . . . N   (Y or N)
Use ENTER to Perform Verification and Selection;
```

Figure 20-5 Storage group DB8XL

The copy pool backup storage group must contain enough eligible target volumes to satisfy the needs of the number of specified backup versions. For example, in our case, the copy pool backup storage group must contain at least 5 (number of source volumes) x 2 (number of versions) = 10 volumes.

Since we are using FlashCopy V2, the target volumes can reside in separate LSS from the source volumes. Note that to be eligible, the target volumes must also meet the following requirements:

- Have the same track form as the source volume
- Be the exact size of the source volume
- Not be a primary or secondary volume in an XRC or PPRC volume pair
- Not be in another FlashCopy relationship

Figure 20-6 illustrates the SMS definitions for the source and target volumes.

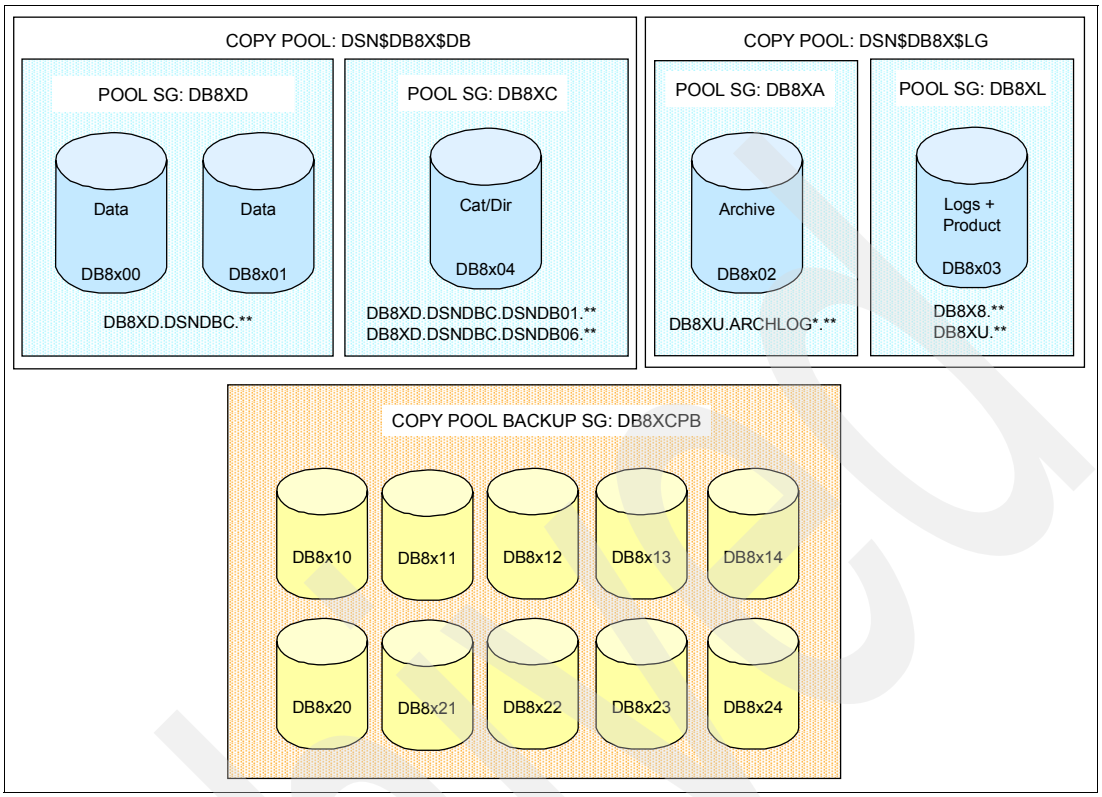


Figure 20-6 SMS definitions for source and target volumes

20.4.4 Validating the Fast Replication environment

The DFSMShsm command FRBACKUP with the PREPARE parameter should be used against each copy pool to validate the Fast Replication environment. You can invoke the FRBACKUP command from a system console, with a TSO command (HSENDCMD), with a batch job, or by using the ARCHSEND macro interface.

When using the PREPARE keyword, DFSMShsm assigns specific target volumes to each source volume for each version of the copy pool. The PREPARE processing does not create a backup of the copy pool. Specifying this keyword is not required, but is recommended when the environment is set up and every time the environment is changed, for example:

- When the number of volumes in a source or backup storage group changes
- When there is a change in the number of versions to maintain
- When the storage groups defined to a copy pool have changed

Example 20-2 shows the DFSMShsm commands that we used to validate our environment. These commands were issued from a TSO terminal by an authorized storage administrator.

Example 20-2 FRBACKUP command – PREPARE parameter

```
HSENDCMD FRBACKUP COPYPOOL(DSN$DB8X$LG) PREPARE
HSENDCMD FRBACKUP COPYPOOL(DSN$DB8X$DB) PREPARE
```

The output of the FRBACKUP command is sent to the DFSMShsm activity log, as shown in Example 20-3.

Example 20-3 FRBACKUP command – PREPARE parameter – DFSMShsm activity log output

```
ARC1801I FAST REPLICATION PREPARE IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8X$LG, AT 12:32:44
ARC1801I (CONT.) ON 2004/07/19
ARC1802I FAST REPLICATION PREPARE HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8X$LG, AT 12:32:46
ARC1802I (CONT.) ON 2004/07/19, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
ARC1801I FAST REPLICATION PREPARE IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8X$DB, AT 12:32:50
ARC1801I (CONT.) ON 2004/07/19
ARC1802I FAST REPLICATION PREPARE HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8X$DB, AT 12:32:53
ARC1802I (CONT.) ON 2004/07/19, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

The DFSMShsm command LIST with the COPYPOOL parameter can be used as shown in Example 20-4, to request detailed information for each version of the specified copy pool. This information includes source volume to target volume pairings. You can invoke the LIST command from a system console, with a TSO command (HSENDCMD), with a batch job, or by using the ARCHSEND macro interface. The following LIST commands are issued from a TSO terminal by an authorized storage administrator.

Example 20-4 LIST command – COPYPOOL parameter

```
HSENDCMD LIST COPYPOOL(DSN$DB8X$LG)
HSENDCMD LIST COPYPOOL(DSN$DB8X$DB)
```

You can decide to let DFSMShsm print the output of the LIST command at your terminal (parameter TERMINAL), send it to a SYSOUT data set (default), or send it to an alternative output data set (parameter OUTDATASET(dsname)).

Example 20-5 is a sample printer list of the copy pool DSN$DB8X$LG. Notice that the pairs source-target for each version have been defined, this is the result of the FRBACKUP command with PREPARE parameter.

Example 20-5 LIST command – COPYPOOL parameter – DSN$DB8X$LG output

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING -- AT 12:34:12 ON 04/07/19 FOR SYSTEM=SC64

COPYPOOL = DSN$DB8X$LG

VERSION  VALID VTOCENQ      DATE           TIME
  ***      *      *      **********      ********
TOKEN(C)=C''
TOKEN(H)=X''

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XA    DB8X02 - DB8X12
DB8XL    DB8X03 - DB8X13

VERSION  VALID VTOCENQ      DATE           TIME
  ***      *      *      **********      ********
TOKEN(C)=C''
TOKEN(H)=X''

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XA    DB8X02 - DB8X10
DB8XL    DB8X03 - DB8X11
```

```
----- END OF -- COPY POOL -- LISTING -----
```

Example 20-6 is a sample printer list of the copy pool DSN$DB8X$DB.

Example 20-6 LIST command – COPYPOOL parameter – DSN$DB8X$DB output

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING -- AT 12:34:16 ON 04/07/19 FOR SYSTEM=SC64

COPYPOOL = DSN$DB8X$DB

VERSION  VALID VTOCENQ      DATE           TIME
  ***      *      *      **********      ********
TOKEN(C)=C''
TOKEN(H)=X''

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XC    DB8X04 - DB8X22
DB8XD    DB8X00 - DB8X23  DB8X01 - DB8X24

VERSION  VALID VTOCENQ      DATE           TIME
  ***      *      *      **********      ********
TOKEN(C)=C''
TOKEN(H)=X''

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XC    DB8X04 - DB8X14
DB8XD    DB8X00 - DB8X20  DB8X01 - DB8X21

----- END OF -- COPY POOL -- LISTING -----
```

20.5 Creating the backup

In this section, we describe two methods of creating the backup, with or without BACKUP SYSTEM utility. However, wherever available, BACKUP SYSTEM should be the method of choice as it has a minimal impact on the system availability.

20.5.1 Using the BACKUP SYSTEM utility

DB2 V8 provides a fast, easy, minimally disruptive way to create volume level backups with the new BACKUP SYSTEM utility. For more information on this utility, refer to 6.4.1, “DB2 BACKUP SYSTEM utility” on page 86.

For the purpose of our test, we use BACKUP SYSTEM FULL (default) to take a full system backup of both data *and* log copy pools, as shown in Example 20-7.

Example 20-7 BACKUP SYSTEM utility

```
//BACKSYSF EXEC DSNUPROC,PARM='DB8X,BACKSYSF'
//STEPLIB  DD  DISP=SHR,DSN=DB8X8.SDSNLOAD
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  BACKUP SYSTEM FULL
//*
```

The output of the BACKUP SYSTEM utility is show in Example 20-8. Notice that the *data* copy pool DSN$DB8X$DB is copied first, followed by the *log* copy pool DSN$DB8X$LG.

Example 20-8 BACKUP SYSTEM utility – Output

```
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = BACKSYSF
DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I    DSNUGUTC -  BACKUP SYSTEM FULL
DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA STARTING,
                       COPYPOOL = DSN$DB8X$DB
                       TOKEN = X'C4C2F8E7BB8A571EFF6BE31000007F94C090'.
DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$DB8X$DB
                       TOKEN = X'C4C2F8E7BB8A571EFF6BE31000007F94C090'
                       ELAPSED TIME = 00:00:03.
DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS STARTING,
                       COPYPOOL = DSN$DB8X$LG
                       TOKEN = X'C4C2F8E7BB8A571EFF6BE31000007F94C090'.
DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$DB8X$LG
                       TOKEN = X'C4C2F8E7BB8A571EFF6BE31000007F94C090'
                       ELAPSED TIME = 00:00:00.
DSNU1602I   DSNUVBBD - BACKUP SYSTEM UTILITY COMPLETED, ELAPSED TIME = 00:00:04.
DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

The token used by DB2 to identify the backup is stored by DFSMShsm in its BCDS records. It is also stored by DB2 in the BSDS. For more information on how to decrypt the DB2 token, see “Use of DB2 tokens” on page 84.

Example 20-9 is an extract of the LIST COPYPOOL command output after BACKUP SYSTEM. It shows that the backup was taken as a Version 001 backup and gives the associated token chosen by DB2.

Example 20-9 LIST COPYPOOL output after BACKUP SYSTEM

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING -- AT 13:08:10 ON 04/07/19 FOR SYSTEM=SC64

COPYPOOL = DSN$DB8X$LG

VERSION  VALID VTOCENQ      DATE           TIME
  001      Y      N      2004/07/19      13:04:27
TOKEN(C)=C'DB8X|«ï..,T..."mé['
TOKEN(H)=X'C4C2F8E7BB8A571EFF6BE31000007F94C090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XA    DB8X02 - DB8X10
DB8XL    DB8X03 - DB8X11

VERSION  VALID VTOCENQ      DATE           TIME
  ***      *      *      **********      ********
TOKEN(C)=C''
TOKEN(H)=X''

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XA    DB8X02 - DB8X12
DB8XL    DB8X03 - DB8X13

----- END OF -- COPY POOL -- LISTING -----

-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING -- AT 13:08:19 ON 04/07/19 FOR SYSTEM=SC64
```

```
COPYPOOL = DSN$DB8X$DB

VERSION  VALID VTOCENQ      DATE           TIME
  001      Y      N      2004/07/19      13:04:24
TOKEN(C)=C'DB8X|«ï..,T..."mé['
TOKEN(H)=X'C4C2F8E7BB8A571EFF6BE31000007F94C090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XC    DB8X04 - DB8X14
DB8XD    DB8X00 - DB8X20  DB8X01 - DB8X21

VERSION  VALID VTOCENQ      DATE           TIME
  ***      *      *      **********      ********
TOKEN(C)=C''
TOKEN(H)=X''

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XC    DB8X04 - DB8X22
DB8XD    DB8X00 - DB8X23  DB8X01 - DB8X24

----- END OF -- COPY POOL -- LISTING -----
```

20.5.2 Without the BACKUP SYSTEM utility

First, we issue the DB2 command SET LOG SUSPEND to suspend logging and update activity, and to quiesce 32 KB page writes and data set extensions, as shown in Example 20-10.

DB2 V8 has enhanced the SET LOG SUSPEND/SET LOG RESUME commands to quiesce 32 KB page writes and data set extends. When the SET LOG SUSPEND command is issued, DB2 waits for any in-progress 32 KB database writes and data set extends to complete, and quiesces any additional writes or data set extends as long as the command is in effect. The SET LOG RESUME command resumes these activities. This helps guard against the possibility of partial page writes taking place during backup processing.

Example 20-10 SET LOG SUSPEND

```
-DB8X SET LOG SUSPEND
DSN9022I  -DB8X DSNJC001 '-SET LOG' NORMAL COMPLETION
DSNJ372I  -DB8X DSNJC09A UPDATE ACTIVITY HAS BEEN 952
SUSPENDED FOR DB8X AT RBA 000081663190, LRSN 000081663190, PRIOR
CHECKPOINT RBA 00008165FC8F
```

Once SET LOG SUSPEND is submitted, we use the DFSMShsm command FRBACKUP with the parameter EXECUTE (default) to create a Fast Replication backup version for each volume in the *data* and *log* copy pools.

Example 20-11 gives the commands that we used in our environment. These commands are issued from a TSO terminal by an authorized storage administrator. The order in which you issue these commands is not important as SET LOG SUSPEND prevents update activity on both copy pools.

Example 20-11 FRBACKUP command – EXECUTE parameter

```
HSENDCMD FRBACKUP CP(DSN$DB8X$LG) EXECUTE NOVTOCENQ TOKEN(DB8X-SETLOG)
HSENDCMD FRBACKUP CP(DSN$DB8X$DB) EXECUTE NOVTOCENQ TOKEN(DB8X-SETLOG)
```

We use the following optional parameters:

- NOVTOCENQ – Used to request DFSMShsm to not serialize on the VTOCs of the volumes being processed. We can only use this parameter because SET LOG SUSPEND ensures that no update activity will occur to the volumes being processed.
- TOKEN – Used to reference this particular backup version. We recommend that you use unique tokens for each version, so that you can easily reference the correct version.

With FRBACKUP, DFSMShsm invokes the DFSMSdss COPY FULL function. The operation is considered successful after the Fast Replication relationship for each source volume is made with a target volume, and a completion message (ARC1000I) is written indicating success or failure of the request. For success see Example 20-12.

Example 20-12 FRBACKUP command – Completion message – Success

```
ARC1000I COPY POOL DSN$DB8X$LG FRBACKUP PROCESSING ENDED
```

For failure, see Example 20-13.

Example 20-13 FRBACKUP command – Completion message – Failure

```
ARC0570I FRBACKUP FOR DB8X12 VOLUME(S) TERMINATED, RC=5 REASON=4
ARC1001I COPY POOL DSN$DB8X$LG FRBACKUP FAILED, RC=0008, REAS=0000
ARC1808E ONE OR MORE VOLUMES FAILED DURING FAST REPLICATION BACKUP OF COPY
ARC1808E (CONT.) POOL DSN$DB8X$LG
```

The output of the FRBACKUP commands is sent to the DFSMShsm activity log, as shown in Example 20-14. More detailed information is sent to the DFSMShsm backup activity log, to which you can refer for problem determination purposes.

Example 20-14 FRBACKUP command – EXECUTE parameter – DFSMShsm activity log output

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8X$LG, AT 13:12:15
ARC1801I (CONT.) ON 2004/07/19
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION BACKUP OF COPY POOL
ARC1805I (CONT.) DSN$DB8X$LG
ARC1805I (CONT.) DB8X02
ARC1805I (CONT.) DB8X03
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8X$LG, AT 13:12:16
ARC1802I (CONT.) ON 2004/07/19, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8X$DB, AT 13:12:36
ARC1801I (CONT.) ON 2004/07/19
ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION BACKUP OF COPY POOL
ARC1805I (CONT.) DSN$DB8X$DB
ARC1805I (CONT.) DB8X04
ARC1805I (CONT.) DB8X00
ARC1805I (CONT.) DB8X01
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8X$DB, AT 13:12:38
ARC1802I (CONT.) ON 2004/07/19, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

Once we have received these messages and verified that the Fast Replication backup has completed successfully, we can issue the SET LOG RESUME command, as shown in Example 20-15, to resume the DB2 logging and update activity.

Example 20-15 SET LOG RESUME

```
-DB8X SET LOG RESUME
DSN9022I  -DB8X DSNJC001 '-SET LOG' NORMAL COMPLETION
DSNJ373I  -DB8X DSNJC09A UPDATE ACTIVITY HAS BEEN RESUMED FOR DB8X
```

Example 20-16 is an extract of the LIST COPYPOOL command output after FRBACKUP command. It shows that the backup was taken as a Version 002 backup and gives the associated token.

Example 20-16 LIST COPYPOOL output after FRBACKUP

```
-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING -- AT 13:24:25 ON 04/07/19 FOR SYSTEM=SC64

COPYPOOL = DSN$DB8X$LG

VERSION  VALID VTOCENQ      DATE           TIME
  002      Y      N      2004/07/19      13:12:15
TOKEN(C)=C'DB8X-SETLOG'
TOKEN(H)=X'C4C2F8E760E2C5E3D3D6C7'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XA    DB8X02 - DB8X12
DB8XL    DB8X03 - DB8X13

VERSION  VALID VTOCENQ      DATE           TIME
  001      Y      N      2004/07/19      13:04:27
TOKEN(C)=C'DB8X|«ï..,T..."mé['
TOKEN(H)=X'C4C2F8E7BB8A571EFF6BE31000007F94C090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XA    DB8X02 - DB8X10
DB8XL    DB8X03 - DB8X11

----- END OF -- COPY POOL -- LISTING -----

-- DFSMShsm CONTROL DATASET --COPY POOL--LISTING -- AT 13:24:31 ON 04/07/19 FOR SYSTEM=SC64

COPYPOOL = DSN$DB8X$DB

VERSION  VALID VTOCENQ      DATE           TIME
  002      Y      N      2004/07/19      13:12:36
TOKEN(C)=C'DB8X-SETLOG'
TOKEN(H)=X'C4C2F8E760E2C5E3D3D6C7'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XC    DB8X04 - DB8X22
DB8XD    DB8X00 - DB8X23  DB8X01 - DB8X24

VERSION  VALID VTOCENQ      DATE           TIME
  001      Y      N      2004/07/19      13:04:24
TOKEN(C)=C'DB8X|«ï..,T..."mé['
TOKEN(H)=X'C4C2F8E7BB8A571EFF6BE31000007F94C090'

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
DB8XC    DB8X04 - DB8X14
DB8XD    DB8X00 - DB8X20  DB8X01 - DB8X21

----- END OF -- COPY POOL -- LISTING -----
```

Figure 20-7 illustrates the status of the volumes after the two backups.

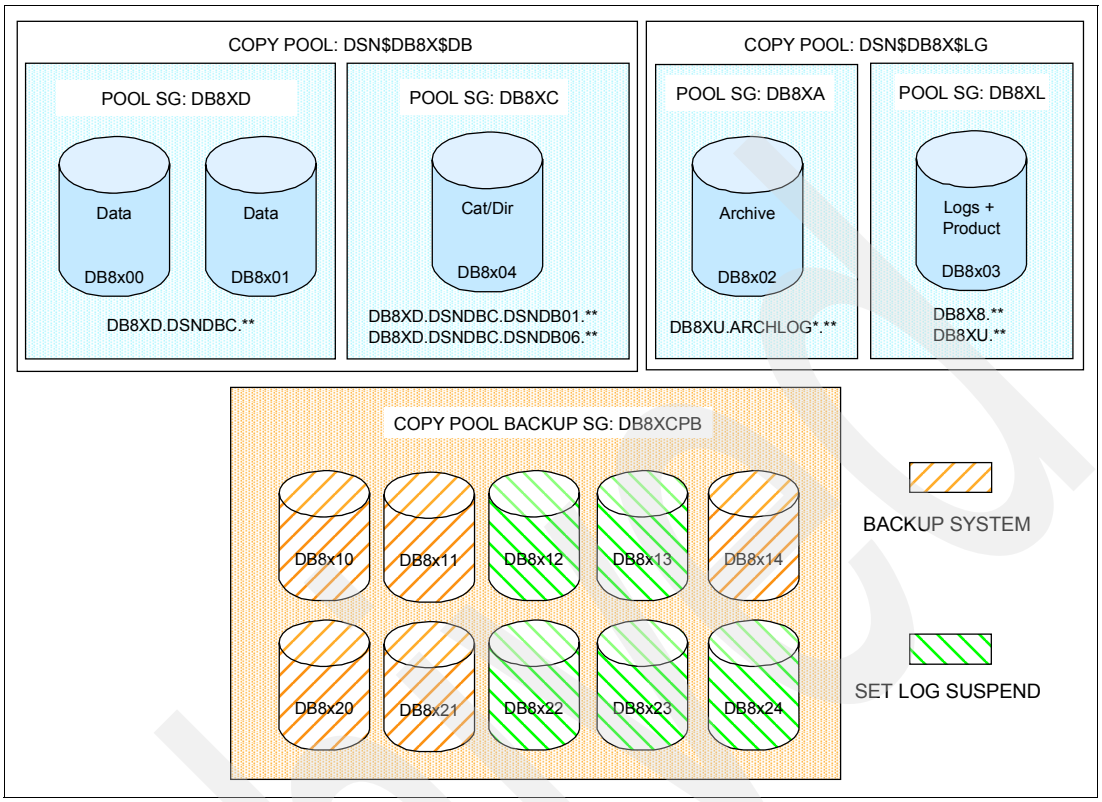


Figure 20-7 Status of the volumes after the two backups

20.6 Dumping the backup

In this section, we describe how to dump the copy pool backup volumes.

20.6.1 What to dump?

The copy pool backup storage group contains several backup versions – two in our case. Using the ISMF interface, we could easily retrieve the list of all the target volumes in the copy pool backup storage group. But we only want to dump the target volumes that belong to a specific version. So we need to establish a list of the target volumes that have to be processed, out of all the volumes in the copy pool backup storage group.

The list of the source-target pairs for each version is stored by DFSMShsm. To retrieve this information, we can use the LIST COPYPOOL command. However, we cannot restrict the scope of the command to a specific version, and the output shown in Example 20-16 is not usable as such, and needs to be processed, for example by a REXX procedure.

First, we want to identify the backup version to be dumped. For the purpose of our test, we need to dump the entire DB2 subsystem, including data *and* logs. Since we have two copy pools, we must make sure that we dump both at the same level.

- We cannot use the version numbers, as they are incremented every time a Fast Replication backup is taken for a given copy pool. The version numbers for data and logs could easily get out of sync. For example, if we used the DB2 utility BACKUP SYSTEM DATA ONLY for the next backup, only the data copy pool would be copied and the version

number for the data would be incremented to Version 003. Then if we took a backup using BACKUP SYSTEM FULL, both copy pools would be copied and the version numbers would be Version 003 for the logs and Version 004 for the data.

- If tokens have been carefully used, we can rely on that information to identify the backup levels. If the backup was taken using the BACKUP SYSTEM FULL utility, DB2 will use the same token for the data and the logs. If the backup was taken using the FRBACKUP command, it is the user's responsibility to define a token to identify the backup version.
- As a last resort, we can use the date and time information, based on the assumption that the backups of the data and the logs were taken almost at the same time.

Once we have identified a valid match, we need to verify that both are valid backup versions (identified by Y in the VALID column). If this is the case, we can extract the list of the target volumes and corresponding source volumes.

We provide you with a sample REXX procedure that issues the LIST COPYPOOL command, extracts the information from the output, builds the list of volumes, and prepares the dump and restore JCL statements. For more information, refer to A.1, "CPBULIST: JCL to dump and restore backup disk volumes" on page 454.

20.6.2 When to dump?

DFSMShsm returns control to the system after a FlashCopy relationship for each volume has been established. The target volumes are available, and you can start dumping them to tape as soon as the Fast Replication backup has completed successfully.

However, the background copies for these volumes continue for several minutes or even hours longer, depending on criteria such as the number of background copies being processed, or the amount of other higher priority activity in the storage subsystem. During this time, the storage subsystem must handle the I/O requests issued by the on-going workload, as well as the background copy processing. Executing the dump jobs concurrently will further increase the load on the storage subsystem. You might experience slight performance degradation and elongated background copy time, and therefore choose to wait until the background copies are finished before starting to dump the backup to tape.

The DFSMShsm command QUERY with the COPYPOOL parameter can be used to request a list of the source volumes that have an outstanding FlashCopy relationship, as shown in Example 20-17. You can invoke the QUERY command from a system console, with a TSO command (HSENDCMD), with a batch job, or by using the ARCHSEND macro interface. The following commands are issued from a TSO terminal by an authorized storage administrator.

Example 20-17 QUERY command – COPYPOOL parameter

```
HSENDCMD QUERY CP(DSN$DB8X$LG)
HSENDCMD QUERY CP(DSN$DB8X$DB)
```

Example 20-18 shows the output of the QUERY command when some source volumes have an outstanding FlashCopy relationship. In this example, the volumes containing the DB2 catalog and directory as well as the user databases, are still being copied.

Example 20-18 QUERY command – COPYPOOL parameter – Output (1/2)

```
ARC1820I THE FOLLOWING VOLUMES IN COPY POOL DSN$DB8X$DB ARE IN AN ACTIVE
ARC1820I (CONT.) FLASHCOPY RELATIONSHIP
ARC1820I (CONT.) DB8X04
ARC1820I (CONT.) DB8X00
ARC1820I (CONT.) DB8X01
```

Example 20-19 shows the output of the QUERY command when there are no volumes in an active FlashCopy relationship. The background copy processing is finished.

Example 20-19 QUERY command – COPYPOOL parameter – Output (2/2)

```
ARC1821I NONE OF THE VOLUMES IN COPY POOL DSN$DB8X$DB ARE IN AN ACTIVE
ARC1821I (CONT.) FLASHCOPY RELATIONSHIP
```

20.6.3 How to dump?

To know how to dump the target volumes, we first need to understand how the backup was created.

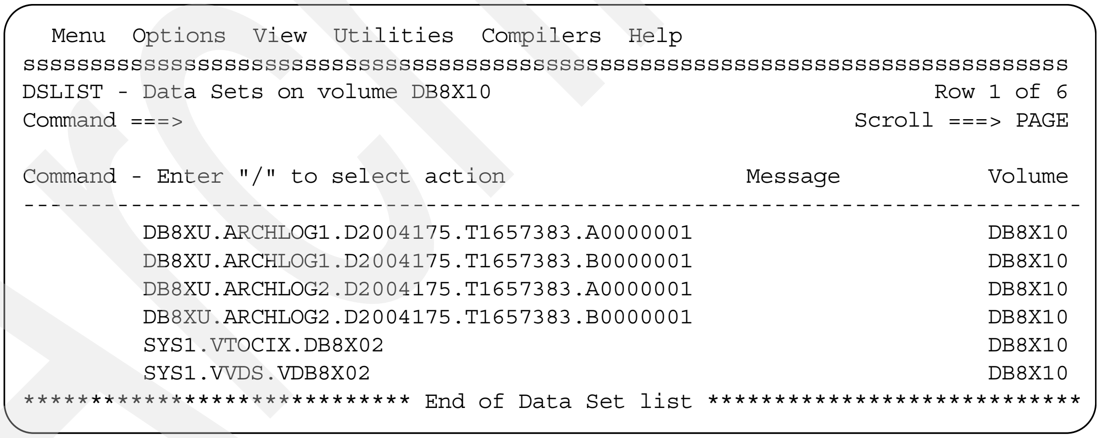
When invoking the Fast Replication function, either directly via the FRBACKUP command, or indirectly via the BACKUP SYSTEM utility, DFSMShsm calls DFSMSdss to perform a COPY FULL operation. Example 20-20 is an extract of the DFSMShsm backup activity log, which shows the command that is issued to copy the source volume DB8X02 on the target DB8X10.

Example 20-20 COPY FULL

```
ARC0640I ARCFRTM -  COPY FULL IDY(DB8X02) ODY(DB8X10) ALLX ALLD(*) DUMPCOND FR(REQ) PUR
```

Since the keyword DUMPCONDITIONING is specified, DFSMSdss copies volume DB8X02 to volume DB8X10, but renames both the VTOC index and the VVDS to match volume DB8X02, as shown in Figure 20-8. Although this means that all data are inaccessible, it will allow the dump of the target volume, since it remains online.

For more information about the DFSMSdss COPY command and the DUMPCONDITIONING keyword, see 5.3.2, “DUMPCONDITIONING parameter” on page 72.



```
  Menu  Options  View  Utilities  Compilers  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
DSLIST - Data Sets on volume DB8X10                                 Row 1 of 6
Command ===>                                                  Scroll ===> PAGE

Command - Enter "/" to select action                  Message          Volume
-------------------------------------------------------------------------------
         DB8XU.ARCHLOG1.D2004175.T1657383.A0000001                      DB8X10
         DB8XU.ARCHLOG1.D2004175.T1657383.B0000001                      DB8X10
         DB8XU.ARCHLOG2.D2004175.T1657383.A0000001                      DB8X10
         DB8XU.ARCHLOG2.D2004175.T1657383.B0000001                      DB8X10
         SYS1.VTOCIX.DB8X02                                             DB8X10
         SYS1.VVDS.VDB8X02                                              DB8X10
***************************** End of Data Set list ****************************
```

Figure 20-8 DUMPCONDITIONING effect

The target volume is not usable by any user, including the DFSMShsm autodump function or the BACKVOL DUMP command. Therefore, the dump must be created outside of DFSMShsm, by using the DFSMSdss DUMP command.

Example 20-21 shows sample JCL statements to dump the target volume to tape.

Example 20-21 DFSMSdss DUMP command

```
//VDB8X10 EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X10
//OUTDS    DD  UNIT=TAPE,DISP=(,KEEP),LABEL=1,
//             VOL=(,RETAIN,,99),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X02
  DUMP FULL INDDNAME(INVOL) OUTDDNAME(OUTDS) ALLDATA(*) -
  ALLEXCP CANCELERROR OPTIMIZE(4) ADMINISTRATOR
```

Because our level of z/OS includes APAR OW48234, the DUMP FULL of the target device modifies the tape records so that it appears as if the tape was created directly from the source volume DB8X02, allowing the volume to be restored without requiring ICKDSF to re-label it.

20.7 Restoring the dump at the recovery system

In a disaster recovery situation, we assume that the volumes at the remote site have been initialized with the same volume serial numbers (VOLID) as the ones used at the local site. In this case, the RESTORE command can be used without the COPYVOLID parameter, as the VOLID is the same on DASD as the one on the tape.

Before restoring the dump, we must unallocate the ICF catalogs that reside on the volumes where the dump will be restored. We use the MODIFY CATALOG command as shown in Example 20-22.

Example 20-22 MODIFY CATALOG command to unallocate ICF catalogs

```
F CATALOG,UNALLOCATE(UCAT.DB8XL)
F CATALOG,UNALLOCATE(UCAT.DB8XD)
```

Example 20-23 shows sample JCL statements to restore a tape dump of a target volume.

Example 20-23 DFSMSdss RESTORE command

```
//VDB8X12  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X02
//INDS     DD  UNIT=TAPE,DISP=(,KEEP),LABEL=1,OLD
//             VOL=(,RETAIN,,99),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X02
  RESTORE FULL INDDNAME(INDS) OUTDDNAME(OUTVOL) -
          CANCELERROR PURGE ADMINISTRATOR
```

The PURGE parameter is required in the restore job because the VVDS name on the target volume does not match the target volume serial number (as a result of the DUMPCONDITIONING parameter that was specified for the full volume COPY).

20.8 Restarting DB2

Once the tape dump is restored at the recovery site, we can restart DB2 (normal restart).

DB2 goes through the usual “crash recovery” processing and detects one inflight transaction. It is automatically backed out during the backward recovery phase, as shown in Example 20-24.

Example 20-24 DB2 restart messages – Syslog

```
-DB8X STA DB2
....
DSNJ127I  -DB8X SYSTEM TIMESTAMP FOR BSDS= 04.201 13:04:18.19
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 1 ACTIVE LOG
DATA SET IS DSNAME=DB8XU.LOGCOPY1.DS01,
STARTRBA=00007E900000,ENDRBA=000080ABFFFF
DSNJ001I  -DB8X DSNJW007 CURRENT COPY 2 ACTIVE LOG
DATA SET IS DSNAME=DB8XU.LOGCOPY2.DS01,
STARTRBA=00007E900000,ENDRBA=000080ABFFFF
DSNJ099I  -DB8X LOG RECORDING TO COMMENCE WITH
STARTRBA=000080676000
...
DSNR001I  -DB8X RESTART INITIATED
DSNR003I  -DB8X RESTART...PRIOR CHECKPOINT RBA=00007F94C090
DSNR004I  -DB8X RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=1, IN ABORT=0, POSTPONED ABORT=0
DSNR007I  -DB8X RESTART...STATUS TABLE
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
B TSO      PAOLOR1      PAOLOR1  DSNESPCS F 00007FDF9E22 201 12:57:12
IEF196I IEF237I 8011 ALLOCATED TO SYS00099
DSNR005I  -DB8X RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR018I  -DB8X RESTART...BACKWARD RECOVERY PROCESSED
 FROM RBA 000080675000 TO RBA 00007FDF9E22
DSNR006I  -DB8X RESTART...COUNTS AFTER BACKWARD
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR002I  -DB8X RESTART COMPLETED
-DB8XRECOVER POSTPONED
DSNV434I  -DB8X DSNVRP NO POSTPONED ABORT THREADS FOUND
DSN9022I  -DB8X DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
```

After restart, we use the -DISPLAY UTILITY(*) command to see if there are any stopped utilities. In our case, no utility was running at the time of the backup, as shown in Example 20-25.

Example 20-25 -DISPLAY UTILITY(*) output

```
-DB8X DIS UTIL(*)
DSNU112I  -DB8X DSNUGDIS - NO AUTHORIZED UTILITY FOUND FOR UTILID = *
DSN9022I  -DB8X DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
```

Finally, we use the -DISPLAY DB(*) SPACE(*) RESTRICT command to check that no objects are in a restricted state, as shown in Example 20-26.

Example 20-26 -DISPLAY DB(*) SPACE(*) RESTRICT LIMIT(*) output

```
-DB8X DIS DB(*) SP(*) RES LIMIT(*)
DSNT365I  -DB8X    NO DATABASES FOUND
DSN9022I  -DB8X DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

Part 5

Additional considerations

In this part of the book, which provides additional considerations, we summarize hints and tips for Recovery best practices, and complete the picture with information specific to data sharing environments.

Part 5 contains the following chapters:

- Chapter 21, “Data sharing” on page 411
- Chapter 22, “Validation and performance” on page 431

21

Data sharing

In this chapter we provide considerations specific to data sharing.

This chapter covers the following topics:

- Data sharing overview
- Data sharing and disaster recovery
- Planning for disaster recovery
- ARCHIVE LOG command for disaster backup
- Recovery procedures in data sharing environment
- Considerations

For more information on the data sharing disaster recovery process, refer to *Data Sharing: Planning and Administration,* SC18-7417.

21.1 Data sharing overview

DB2 data sharing, introduced with DB2 Version 4, is one of the most significant enhancements to DB2 availability. Data sharing runs in a parallel sysplex environment, a set of z/OS systems that communicate and cooperate with one another using specialized hardware and software. A parallel sysplex contains, along with the processors that are running the DB2 subsystems, one or more processors that act as *coupling facilities (CF)*. Figure 21-1 shows the basic components of a DB2 data sharing environment running in a parallel sysplex.

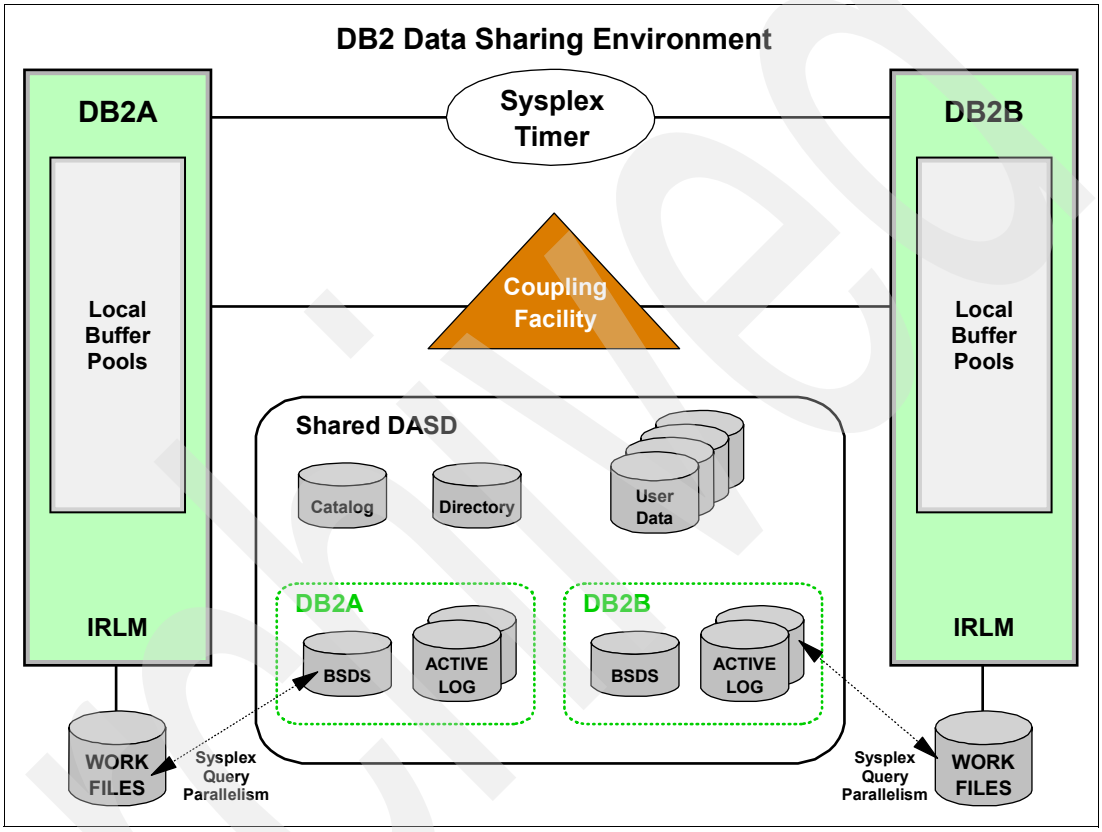


Figure 21-1 The data sharing environment

The coupling facilities run specialized control code that coordinates activity across the multiple subsystems (called members) of the data sharing group. The coupling facilities contain three major structures that are utilized by DB2, as shown in Figure 21-2:

- Group Buffer Pools: These are cache structures containing data and index pages for DB2 objects that are being accessed across the data sharing group. The group buffer pools, which usually have a one-to-one correspondence with the local buffer pools on each member, monitor which members are accessing which DB2 data sets and thereby keep track of any data that is being updated by one member and is subsequently accessed by a second member. This is called inter-DB2 read/write interest in a data set, and causes the pages being updated to be written from the local buffer pools to the respective group buffer pools. The member that is reading the data then reads the data from the group buffer pools, ensuring that the most recent data is read and data integrity is maintained.
- Lock Structure: The lock structure is used to ensure that data that is being updated by one member cannot be updated by another member until the first member commits or rolls back, thus releasing the locks that it held. the lock structure ensures that the same data is not updated by multiple members at the same time.

- Shared Communications Area (SCA): The SCA is used to coordinate recovery and startup across the group. The SCA contains:
  - DB2 member names
  - BSDS names for each member
  - Database exception status conditions for DB2 objects and DB2 members
  - Recovery information, such as log data set names and information about indoubt transactions

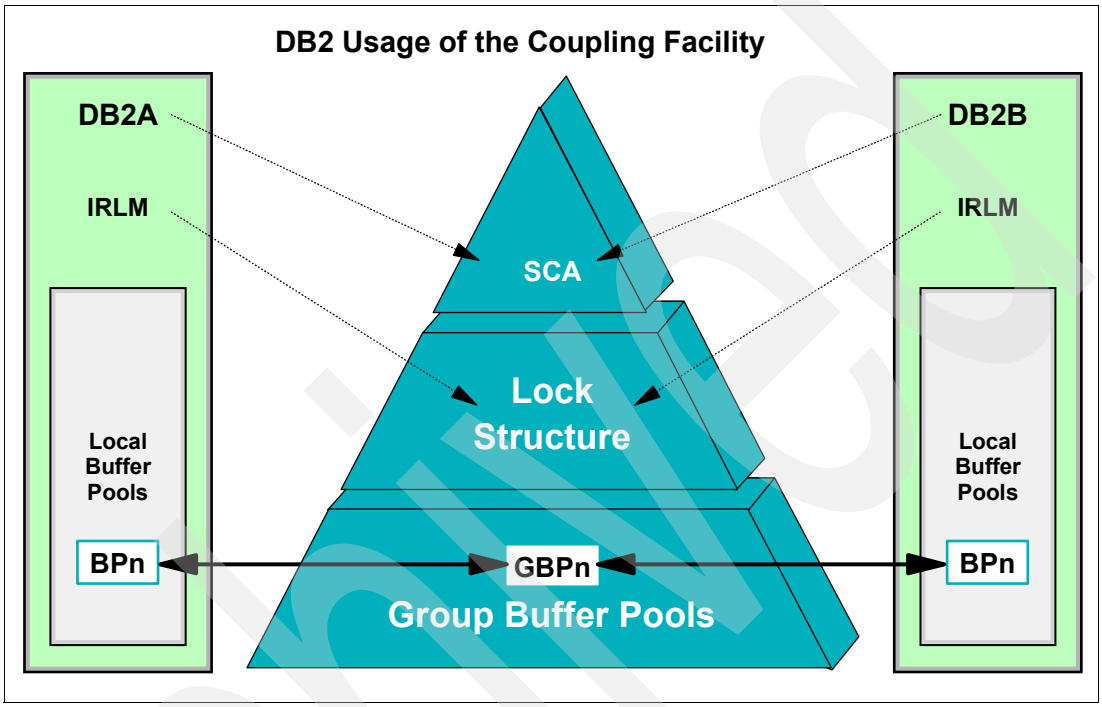


Figure 21-2 DB2 and coupling facility

The above structures reside on one or more coupling facilities. Ideally you want at least two coupling facilities with failover capability from each to the other. In addition to the coupling facilities a parallel sysplex includes one or more sysplex timers, which synchronize the system clocks of all the members of the data sharing group. Since multiple members can update data from the same set of tables, and since recovery could be issued from any one of those members, the log data on each member uses a common log record sequence number (LRSN) that is derived from the store clock timestamp and synchronized across all members of the group by the sysplex timer. Multiple sysplex timers are recommended for failover capability.

DB2 restart and recovery

Consider using Automatic Restart Manager (ARM) in conjunction with the RETAINED LOCK TIMEOUT option (ZPARM RETLWAIT) of installation panel DSNTIPI to ensure that retained locks are released in a timely manner. See *DB2 UDB for z/OS Version 8 Data Sharing: Planning and Administration*, SC18-7417 for more information on creating an ARM policy.

If you intend to use Restart Light at some point, you should plan for it ahead of time by creating an ARM policy for the DB2 group that specifies LIGHT(YES) within the RESTART_METHOD(SYSTERM) keyword for the DB2 element name. For example:

```
RESTART_METHOD(SYSTERM,STC,'cmdprfx STA DB2,LIGHT(YES)')
```

If you require faster restarts, we recommend that you take more frequent checkpoints.

Starting with Version 7 you can specify the checkpoint frequency in terms of the number of minutes between checkpoints in addition to the traditional number of log records. The checkpoint frequency and frequency type (log records or minutes) are set on panel DSNTIPL at installation time. Refer to *DB2 UDB for z/OS Version 8 Installation Guide*, GC18-7418 for instructions on setting these values. The checkpoint frequency can impact how long it takes to restart DB2. When choosing a value, you must consider the trade-off between the overhead that is needed for frequent checkpoints and the time that is needed to restart a DB2 subsystem. If your primary concern is DB2 restart time, we recommend you use a use a checkpoint frequency between 50,000 and 1,000,000 log records. Otherwise, use a checkpoint frequency of 2 to 5 minutes.

You can also enable Fast Log Apply to speed up restarts, assuming that you have adequate DBM1 storage.

Specify AUTOREC(YES) for each of your group buffer pools to enable automatic recovery of your GBPs. You do not need to specify AUTOREC(YES) for any GBPs which are defined with GBPCACHE(NO) since data will be written directly to DASD.

Data sharing generally requires more disk space for the active log, and you need to archive the logs more frequently. See Chapter 5 of *DB2 UDB for z/OS Version 8 Data Sharing: Planning and Administration*, SC18-7417 for more information.

21.2 Data sharing and disaster recovery

When comparing data sharing to non-data sharing disaster recovery, the major differences are related to the structural components of the data sharing **group**, already briefly described:

- Each member of the data sharing group has its own log, and the logs are controlled by LRSN, rather than RBA. This adds complexity in managing the log data sets during recovery and in defining the “common” recovery point.
- The Coupling Facility and its structures must be cleaned up and rebuilt Group restart needs to be performed, locks released, and structures rebuilt. Restart light can help in some situations.

In this section, we describe how your recovery site must be set up for data sharing to be able to recover your local data sharing group. We present an overview of how to recover a data sharing group at a recovery site.

After data sharing is enabled for a DB2® subsystem, you need to run data sharing for the rest of the subsystem's life, including execution of the DB2 at a disaster recovery site. Proper planning of software as well as hardware is necessary — for example, you will need coupling facilities (CFs) at the disaster recovery site.

The general rule at the disaster recovery site is that all members must be started successfully at least once.

The exception to this rule is a DB2 subsystem that was quiesced at the time of the failure, is no longer in use, or is a read-only DB2 member. If you are unsure about the status of your DB2 members, restart them all. DB2 needs restarting at the disaster recovery site to perform any recovery actions that are required to bring the system to a current and consistent state. After they are finished with restart and recovery, the DB2 subsystems can be shut down cleanly and do not need to run at the disaster recovery site.

The recovery site must have a data sharing group that is identical to the data sharing group at the local site. It must have the same group name and the same number of members and the names of the members must be the same as shown in Figure 21-3.

The active CFRM policy at the recovery site must define coupling facility structures for the SCA, the DB2 lock structure, and the needed group buffer pools with the same names as for the local site. The size of the structures may be different, as may be the preference and exclusion lists for the structures. It is not necessary to define group buffer pools which are not needed at the recovery site.

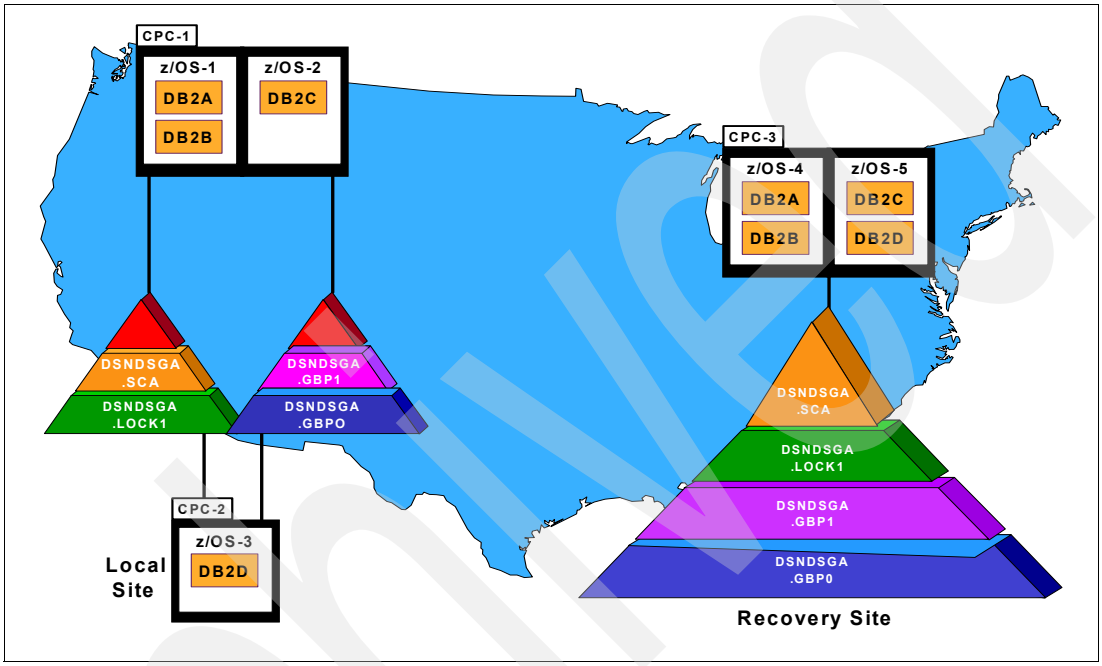


Figure 21-3 Recovery site requirements

During the data sharing disaster recovery:

- The member specific tasks must be done for each member.
- You must specify LRSNs when adding archive log data sets.
- You can specify common ENDLRSN for CRCRs.
- You must use common ENDLRSN for CRCRs for SCOPE (GROUP).
- You must perform group restart.

For a detailed description of the non-data sharing environment, refer to Chapter 3, “Traditional recovery” on page 43.

21.3 Planning for disaster recovery

There are multiple ways of preparing for disaster recovery. Independent of the way you choose, you should plan and practice your recovery procedures before the disaster strikes. Some of the popular options for disaster backup and recovery are:

21.3.1 Dump and restore of DB2 environment

With this option, you dump the entire DB2 environment for your data sharing group at the local site and reestablish it at the recovery site as follows:

1. Copy everything periodically with a DFSMSdss volume dump.
2. Send it to the recovery site, and then restore it there.

This is an inexpensive and easy backup option, but it can only be used to recover to the point-in-time when the dump was taken.

A disadvantage of this option is that, to obtain data consistency, you must bring down *all DB2 members* of your data sharing group for the length of time it takes to dump the DB2 environment.

Depending on the frequency with which you can afford to dump the entire DB2 environment (typically, once per week), the potential loss of data and work in case of a disaster may be very large.

21.3.2 Consistent Point-in-Time Recovery

With this approach, you must take image copies of the DB2 Catalog, the DB2 Directory, and the user table spaces at the local site. Thereafter, you must issue the ARCHIVE LOG command to create a consistent set of archive log data sets for all members of the data sharing group.

At the recovery site, the DB2 environment is reestablished, whereby the DB2 Catalog, the DB2 Directory, and the user tables spaces are recovered to the point-in-time when the ARCHIVE LOG command created the archive logs.

The frequency with which you issue the ARCHIVE LOG command determines how much data is lost in a disaster recovery. You can only recover to the *point-in-time when the latest ARCHIVE LOG command was issued*.

For ARCHIVE LOG with MODE(QUIESCE), data consistency is achieved at the local site and for ARCHIVE LOG with SCOPE(GROUP), it is achieved at the recovery site. For more detailed information, refer to 21.4, “ARCHIVE LOG command for disaster backup” on page 417.

This is the approach recommended by DB2. In 21.5, “Recovery procedures in data sharing environment” on page 419, we will concentrate on this approach and discuss the data sharing considerations for it.

Important: Planning the timing of backup is essential for a successful recovery. You can back up in this order:

1. Take image copies of the user table spaces.
2. Take image copies of the DB2 catalog and DB2 directory.
3. Truncate the active log.
4. Perform MVS catalog backup.

Copy and send the following objects that you need to recover at the recovery site:

1. Image copies of the DB2 catalog, the DB2 directory, and the user table spaces:

   Run COPY utility with RECOVERYDDN (in most cases, on tape).

2. Force archiving:

   You must issue the ARCHIVE LOG command to create a consistent set of archive log data sets for all member of the data sharing group. When using the ARCHIVE LOG command with SCOPE(GROUP), there is no impact on your applications. Also, you have to use MODE(QUIESCE), if you plan on the remote site, a PIT Recovery to the archive log point.

3. BSDS list:

   You will use the DSNJU004 utility to determine the fully qualified name of the archive log you need to restore from and the RBA you will need to give in CRESTART statement of DSNJU003.

21.3.3 Continuous archive log transfer

With this approach, you continuously send copies of the image copies for the DB2 Catalog, the DB2 Directory, and the user tables spaces to the recovery site. Whenever an archive log data set is created for a member, you send a copy of it and BSDS list to the recovery site.

Sending the archive logs as they are created is not sufficient to achieve data consistency at the recovery site. In addition, you must, from time to time, issue the ARCHIVE LOG command for each member in order to create a consistent set of archive log data sets for all members of the data sharing group. For more detailed information on ARCHIVE LOG command, refer to 21.4, “ARCHIVE LOG command for disaster backup” on page 417.

At the recovery site, you reestablish the DB2 environment in a similar fashion as was indicated for the previous methods. However, there are some problems with finding a restart point to recover to during disaster recovery and achieving data integrity. Considerable analysis and cleanup has to be done in this case to obtain consistency for your data. The potential loss of data for this method corresponds to the size of an active log data set.

Copy and send the following objects to the recovery site *as you produce them*:

1. Image copies of the DB2 catalog, the DB2 directory, and the user table spaces:

   You can back up using COPY with RECOVERYDDN (in most cases, on tape) and the copies can be sent to the recovery site.

2. Archive logs:

   You usually use dual archive logging. The second archive log (in most cases, on tape) is intended as a backup or it can be sent to the recovery site in preparation for disaster recovery.

3. BSDS lists:

   You will use the report to determine the fully qualified name of the archive log you need to restore from and the RBA you will need to give in CRESTART statement of DSNJU003. Your system should usually schedule a search for the message DSNJ139I LOG OFFLOAD TASK ENDED. Once the message is issued, you can release the batch job to run your DSNJU004 report.

21.4 ARCHIVE LOG command for disaster backup

As mentioned before, you must issue the ARCHIVE LOG command with the options MODE(QUIESCE) or SCOPE(GROUP) in order to create archive log data sets for all members of the data sharing group. The two options are mutually exclusive.

21.4.1 MODE(QUIESCE) versus SCOPE(GROUP)

With MODE(QUIESCE), you create a *group wide point of consistency* (GPOC) at the local site before the current active log data sets of all members are truncated and archived. Thus, conceptually, the active log data sets are truncated at the same point-in-time.

With SCOPE(GROUP), you merely ensure that the current active log data sets of all members are truncated and archived, but you do not ensure that they are truncated at the same consistent point-in-time. DB2 Version 5 or later versions allow you to establish data consistency at the recovery site.

For more detailed information on how to find the RBA that you want to recover to consistent point-in-time, refer to 21.6.2, “Determine ENDLRSN for the conditional restart record” on page 423.

21.4.2 Advantage and disadvantage of MODE(QUIESCE)

MODE(QUIESCE) halts all new update activity for a specified period, referred to as the quiesce period. Ongoing units of recovery are not interrupted. If all ongoing units of recovery for all members are completed within the quiesce period, that is, a group wide point of consistency is reached, the current active log data sets of all members are truncated and offloaded. If some members do not reach a point of consistency within the quiesce period, none of the current active log data sets are truncated, and normal processing is resumed, that is, update activities are allowed to proceed.

In Example 21-1, MODE(QUIESCE) has the disadvantage that the ARCHIVE LOG command fails if the group wide point of consistency cannot be achieved within the quiesce period for the command and, further, that the ARCHIVE LOG command must be repeated until it successfully creates the archive log data sets for all members. For a busy data sharing group, this may be a challenge.

Example 21-1 ARCHIVE LOG MODE(QUIESCE) command fail

```
DSNJ310I  -D8F2 ASYNCHRONOUS ARCHIVE LOG COMMAND 647
QUIESCE PROCESSING STARTING FOR MAXIMUM OF   5 SECONDS
DSNJ310I  -D8F1 ASYNCHRONOUS ARCHIVE LOG COMMAND 880
QUIESCE PROCESSING STARTING FOR MAXIMUM OF   5 SECONDS
DSNJ357E  -D8F2 DSNJC007 SYSTEM QUIESCE FAILED
DSNJ312I  -D8F2 ARCHIVE LOG QUIESCE ENDED.  UPDATE 649
ACTIVITY AGAINST DB2 RESOURCES IS NOW RESUMED.
DSNJ317I  -D8F1 ARCHIVE LOG QUIESCE PERIOD EXPIRED. 881
NUMBER OF OUTSTANDING UR'S =    1.  ARCHIVE LOG PROCESSING WILL BE
TERMINATED, AND UPDATE ACTIVITY AGAINST DB2 RESOURCES WILL BE
RESUMED.  23:26:00
DSNJ312I  -D8F1 ARCHIVE LOG QUIESCE ENDED.  UPDATE 882
ACTIVITY AGAINST DB2 RESOURCES IS NOW RESUMED.
DSNJ355I  -D8F2 DSNJC005 ARCHIVE LOG:  ONE OR MORE 650
ACTIVE MEMBERS FAILED TO RESPOND OR START OFFLOAD TASKS
DSNJ359I  -D8F2 MEMBER RESPONSES:  651
DSNJ351I  -D8F2 MEMBER D8F1  STATUS= Time Exceeded
DSNJ351I  -D8F2 MEMBER D8F2  STATUS= Quiesced
DSNJ361I  -D8F2 DSNJC005 END OF MEMBER RESPONSE LIST.
DSNJ357E  -D8F2 DSNJC005 SYSTEM QUIESCE FAILED
```

21.4.3 ARCHIVE LOG command verification

You can run the Print Log Map (DSNJU004) utility to print the Archive Log Command History for each member of the data sharing group as shown in Example 21-2.

Example 21-2 Archive Log Command History on Print Log Map

```
                    ARCHIVE LOG COMMAND HISTORY
                             MEMBER  D8F2
          DATA SHARING GROUP DB8FU    CONTAINS 2 MEMBERS
                     03:36:18 AUGUST 03, 2004
 DATE/SDATE   TIME/STIME      RBA        MODE    WAIT  TIME   SCOPE   CMD ORIGININ   STATUS        ACTIVE
------------  ----------  ------------  -------  ----  -----  -----   ------------  -----------   ------
AUG 03, 2004  03:35:07.9  000012ED7D36  QUIESCE   NO     5 D    G      D8F2          ORIGINATOR      2
AUG 02, 2004  22:47:01.9  00000E0C5162                           G      D8F1          PARTICIPANT     2
AUG 02, 2004  21:49:30.3  0000036C5756  QUIESCE   NO     5 D    G      D8F2          ORIGINATOR      2
```

You have now the necessary information to check the following situations:

- The Archive Log Command History for each of the members should contain an entry for the time when the ARCHIVE LOG command was issued. If it does not, active log truncation and archiving has not or has not yet been performed for the member. The reason can be that the ARCHIVE LOG command failed or that the member was inactive when the command was issued and has not been restarted since.
- The number in the ACTIVE column of the entry for the ARCHIVE LOG command should be equal to the number of members in the data sharing group. If the number is smaller than expected, some of the members were not active when the ARCHIVE LOG command was issued and archiving has not yet been done for these members. You must determine these members and start them if they have not yet been started in the meantime.
- The Archive Log Command History of members that were inactive when the ARCHIVE LOG command was issued and have not been restarted since does not contain an entry for the ARCHIVE LOG command.

  When the members are restarted, an entry is created in their Archive Log Command History with a STATUS of STOPPED, a blank RBA column, and the date and time of the ARCHIVE LOG command. In addition, a second entry with a STATUS of STARTED, a valid RBA, and the date and time when the member was started is created to indicate that the active log truncation and the offload processing were triggered by starting the member.

  The RBA value in an Archive Log Command History entry specifies the position at which the current active log data set for the member was truncated.

21.5 Recovery procedures in data sharing environment

When performing a disaster recovery using the Consistent Point-in-Time Recovery approach (refer to 21.3.2, “Consistent Point-in-Time Recovery” on page 416), you must follow the steps listed in this section. Basically, the steps are the same for the data sharing environment as for the non-data sharing environment. Subsystem-specific steps, such as the adaptation of DSNZPxxx in order to defer restart processing for databases, must be performed for every member of the data sharing group.

We give you an overview of the steps involved. We do not discuss the steps in detail unless there are differences for data sharing. Rather, we concentrate on the changes introduced by data sharing. For a detailed description of the steps for the non-data sharing environment, refer to 3.3, “Recovery procedures” on page 47.

21.5.1 Establish environment

This section shows the preparatory steps that are necessary before the actual data recovery can begin.

1. If an ICF catalog does not already exist at the recovery site, run job DSNTIJCA to create a user catalog.
2. Use the access method services IMPORT command to import the any ICF catalogs.
3. Restore DB2 libraries.

   Include DB2 RESLIBs, SMP libraries, user program libraries, user DBRM libraries, CLISTs, SDSNSAMP, or where the installation jobs are, JCL for user defined table spaces, and so on.
4. Use IDCAMS DELETE NOSCRATCH to delete all catalog and user objects

   Because step 2 imports a user ICF catalog, catalog reflects data sets that do not exist on disk.
5. Define required data sets.

   During this step, job DSNTIJIN should be modified and run to define the VSAM data sets for the DB2 Catalog and the DB2 Directory databases for the data sharing group (step DSNTDIR) and, for all members, the VSAM data sets for the BSDSs and the active log data sets (step DSNTDBL) at the recovery site. Of course, the VSAM data sets need only be defined if they have not yet been defined.
6. Change ZPARM needed for disaster recovery.

   During this step, the DSNZPxxx subsystem parameter load module is modified for all members as follows:

   a. The first two entries in the DSN6SPRM macros should be **DEFER** and **ALL**. This will indicate to defer recovery of all objects, including DB2 catalog objects on DB2 start up.
   b. ARC2FRST should be **YES** to indicate to DB2 to read COPY2 of the archive log first.
   c. SITETYP should be **RECOVERYSITE** to indicate to the RECOVER utility that it should use image copies registered to RECOVERYDDN.

21.5.2 Recover the BSDS

For each member, both the primary and the alternate BSDSs are restored (using VSAM Access Method Services REPRO with the REUSE option) from the BSDS copy sent with the last archive log data set copy for the member. Since the last archive log data set is not recorded in the BSDS, it must be added using the Change Log Inventory (DSNJU003) utility. The active log data sets recorded in the BSDS are those of the local site. They must be deleted and active log data sets for the recovery site must be added using the Change Log Inventory utility.

1. Review the DSNJU004 report.

   > **Attention:** The assumption is that we have sent BSDS list on recovery site to determine the name and the RBA of the archive log once the message DSNJ139I LOG OFFLOAD TASK ENDED is issued at the local site.

   Determine the fully qualified name of the archive log you need to restore from and the RBA you will need to give in CRESTART statement of DSNJU003.
2. Copy the BSDS from the last archive log tape.

3. (Optionally) copy archive log to disk.

   This eliminates multiple tape mounts and speeds up recovery.

4. Update the BSDS:

   a. Delete the active logs and add new active logs in the BSDS. Do not specify a STARTRBA or ENDRBA in NEWLOG statement.

   b. Register the last archive log data set in the archive log inventory of the BSDS just restored.

      When you register the last archive log data set in the BSDS, you must provide its starting (STARTRBA) and ending RBAs (ENDRBA) and its starting (STARTLRSN) and ending LRSNs (ENDLRSN) in the NEWLOG statement for the Change Log Inventory utility.

5. If you are using the DB2 distributed data facility, run the DSNJU003 to update the LOCATION and LUNAME in the BSDS.

6. Use the print log map utility (DSNJU004) to ensure that the BSDS correctly reflects the active and archive log data set inventories.

7. If you are using dual BSDSs, make a copy of the newly restored BSDS data set to the second BSDS data set.

8. Run DSN1LOGP to show any activity that was ongoing during the archive processing:

   a. Input is the latest archive log (file 2).

   b. STARTRBA is the starting RBA of the last complete checkpoint on the last archive log.

21.5.3 Restart DB2 (Conditional Restart)

1. Use DSNJU003 to create conditional restart record.

   Specify CRESTART CREATE, ENDLRSN=nnnnnnnnnnnn, FORWARD=YES, BACKOUT=YES. To determine the ENDLRSN, refer to 21.6.2, “Determine ENDLRSN for the conditional restart record” on page 423.

2. Run DSNJU004 to verify BSDS contents.

   Verify that the conditional restart control record that you created in the previous step is active with correct ENDLRSN and checkpoint.

3. Set up or clean up the coupling facility (CF) structure:

   – As noted before, the coupling facility structures for the data sharing group at the recovery site must have the same names as the corresponding structures at the local site. If the structures have not yet been defined in the active CFRM policy, they must be defined. Group buffer pools that are not needed at the recovery site need not be defined.

   – Remove old information from the coupling facility, if you have information in your coupling facility from practice startup. For more information, refer to 21.6.1, “Cleaning up coupling facility structures” on page 423.

4. Start all members with ACCESS(MAINT) to perform group restart:

   – Start all members at the recovery site. Because there is no SCA or no DB2 lock structure, a group restart is performed reconstructing the SCA and the DB2 lock structure from information in the BSDSs and on the logs of the members of the data sharing group.

   – Units of recovery that are incomplete as the consequence of the log truncation specified via the conditional-restart control records for the members and that must be rolled back are backed out on the log, that is, redo compensation log records are

written for them to the log during the group restart. Remember that the restart processing for all databases has been deferred. Later, when the table spaces are recovered, the redo compensation log records will ensure that the changes for these units of recovery are not reflected in the table spaces.

Attention: You need to log on with INSTALL SYSADM TSO user id.

5. Recover indoubt URs.

   If you used ARCHIVE LOG command with SCOPE(GROUP), you may have indoubt units of recovery. Determine the proper action to take (commit or abort) for each such unit of recovery. The RECOVER utility which will be used by the subsequent steps fails on table spaces that have indoubt units of recovery.

21.5.4 Recover DB2 system data

You must recover the DB2 catalog and DB2 directory before recovering user table spaces.

1. Recover DB2 system data that must be recovered in a particular order as follows:
   a. The SYSUTILX table space must be first and alone.
   b. The SYSUTILX indexes must be alone.
   c. Recover the rest of the DB2 catalog and DB2 directory objects starting with DBD01, in the order shown in the description of the *RECOVER utility in Part 2 of DB2 Utility Guide and Reference*.
2. Verify your subsystem.
   a. Display DSNDB01 and DSNDB06.
   b. DSN1COPY with CHECK to do a validity check on each page.
   c. DSN1CHKR to verify the integrity of the DB2 directory and DB2 catalog table spaces.
   d. Run CHECK INDEX to verify that the DB2 directory and DB2 catalog indexes are consistent with the data they index.

21.5.5 Prepare the subsystem for use

1. Define and initialize the work file database.

   For each started member, its work file database must be created and initialized. Thereafter, you must start the work file database by means of the START DATABASE command.
2. Modify DSNZPxxx for all members.

   The first two entries in the DSN6SPRM macros should be RESTART and **ALL** to allow restart for all databases.
3. Stop and start DB2 members.

   You are not required to restart all members. You can only start the members you need at the recovery site. However, some of the subsequent steps may complete faster if you start more members and distribute the work.
4. Make a full image copy of the DB2 catalog and DB2 directory.
5. Recover user data.

   The recovery process for your application data should be the same as you use at your local site, except that you will be using your RECOVERYDDN data sets.

6. Make a full image copy of all table spaces and indexes with the COPY YES.

21.6 Considerations

The procedure for data sharing at the recovery site differs in that extra steps exist for cleaning out old information in the coupling facility. Old information exists in the coupling facility from any practice startups. In addition, you must prepare each subsystem (rather than just a single system) for a conditional restart.

21.6.1 Cleaning up coupling facility structures

Defined structures must not yet be allocated in the coupling facility. If they are allocated, they must be deleted. This is necessary to cause a group restart which reconstructs the SCA and the DB2 lock structure at the recovery site. The group buffer pools must be deleted to avoid the usage of obsolete buffer information.

For each allocated structure, check to see if it has failed-persistent connections. Because none of the members is active, the structures cannot have active connections. You need not check the SCA for connections. It can never have failed-persistent connections because it is allocated with connection disposition DELETE, and the members always disconnect from it with disconnection reason NORMAL.

Only after all connections have been deleted, you can delete the structure. If the structure is a group buffer pool, it is automatically deleted when the last connection is deleted. Therefore, you do not have to delete group buffer pools.

To remove old information from coupling facility, do the followings:

1. To ensure that none of the structures is allocated, issue this command:

   *D XCF,STRUCTURE,STRNAME=group-name**

   which displays the status of all structures of the data sharing group. Group-name is the name of the data sharing group.
2. If a structure has failed-persistent connections, delete them by means of this command:

   SETXCF FORCE,CONNECTION,STRNAME=structure-name,CONNAME=ALL
3. To delete an allocated SCA or DB2 lock structure, issue this command:

   SETXCF FORCE,STRUCTURE,STRNAME=structure-name

After all allocated structures have been deleted, you are ready to proceed to next step, that is, the group restart for the data sharing group at the recovery site.

21.6.2 Determine ENDLRSN for the conditional restart record

When you restart the members of the data sharing group at the recovery site, you must truncate the logs of the members at the same “logical” time. In DB2 Version 5 and later versions, you can use an LRSN to accomplish this.

Determine the ENDLRSN value using one of the following methods:

ENDLRSN with SCOPE(GROUP) on print log map

You can specify an LRSN instead of an RBA for the CRESTART statement. Because the ENDLRSN for the CRESTART statement need not be rounded, you can specify a common LRSN for all members. This LRSN can be any LRSN that is within the LRSN ranges of the latest archive log data sets of all members.

Mathematically speaking, this means that the LRSN must be contained in the intersection of the LRSN ranges (of the latest archive log data sets) for all members. To lose as little work as possible, you will want to make this LRSN as large as possible, namely, by choosing the smallest ending LRSN of any of these archive log data sets.

Compare the ending LRSN values for all members' archive logs, and choose the lowest LRSN as the truncation point. Example 21-3 shows the log map for the first member.

Example 21-3 Print Log Map - member D8F1

```
ARCHIVE LOG COPY 2 DATA SETS
  START RBA/LRSN/TIME   END RBA/LRSN/TIME      DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
   00000573F000           000005789FFF       2004.215  18:47 DSN=DB8FU.D8F1.ARCLG2.A0000003
     BB9C30F4D93F           BB9C3DD08828                     PASSWORD=(NULL) VOL=SBOXE2 UNIT=3390
 2004.215  21:49:30.3  2004.215  22:47:01.9
                                                             CATALOGUED
```

Example 21-4 shows the log map for the second member.

Example 21-4 Print Log Map - member D8F2

```
ARCHIVE LOG COPY 2 DATA SETS
  START RBA/LRSN/TIME   END RBA/LRSN/TIME      DATE    LTIME DATA SET INFORMATION
 --------------------  --------------------  --------  ----- --------------------
  00000DF86000           00000E0C5FFF       2004.215  18:47 DSN=DB8FU.D8F2.ARCLG2.A0000005
    BB9C3DCA49BB           BB9C3DD08896                     PASSWORD=(NULL) VOL=SBOXE2 UNIT=3390
2004.215  22:46:55.3  2004.215  22:47:01.9
                                                             CATALOGUED
```

The lowest LRSN is **BB9C3DD08828**. To get the last complete log record, *you must subtract 1 from that value*, so you would enter **BB9C3DD08827** as the ENDLRSN value in the CRESTART statement of the DSNJU003 utility for each of the members at the remote site. All log records with a higher LRSN value are discarded during the conditional restart.

ENDLRSN with SCOPE(GROUP) on DSN1LOGP output

To determine the common ENDLRSN value for the CRESTART statements of all members, you can use the DSN1LOGP utility with the SUMMARY option to create summary reports for the latest archive log data sets of the members of the data sharing group.

Message DSN1213I for a member at the end of the Summary of Completed Events section returns the proper ending LRSN for the printed log data set. Thus, you can derive the common ENDLRSN value for the CRESTART statements by choosing *the minimum LRSN value of all messages DSN1213I*. Because the LRSN displayed by DSN1LOGP is the LRSN of the actual last log record for the log data set, you need not subtract one from the minimum.

Example 21-5 shows the log output for the first member.

Example 21-5 DSN1LOGP output for member D8F1

```
DSN1213I DSN1LGRD LAST LOG  RBA ENCOUNTERED 0000057892AC


DSN1213I DSN1LGRD LAST LOG LRSN ENCOUNTERED BB9C3DCDB4FF
```

Example 21-6 shows the log output for the second member.

Example 21-6 DSN1LOGP output for member D8F2

```
DSN1213I DSN1LGRD LAST LOG  RBA ENCOUNTERED 00000E0C41A2


DSN1213I DSN1LGRD LAST LOG LRSN ENCOUNTERED BB9C3DD08788
```

The minimum LRSN is **BB9C3DCDB4FF**. To get the actual last log record, you would enter **BB9C3DCDB4FF** as the ENDLRSN value in the CRESTART statement of the DSNJU003 utility for each of the members at the remote site.

You can print the summary reports for the latest archive log data sets of all members with a single run of the DSN1LOGP utility by using MxxARCHV DD statements to identify the archive log data sets.

21.6.3 Choosing an LRSN

Similarly to the initial steps described previously to choose an RBA, for data sharing you want to use an LRSN at the end of the archive logs that is common to all. The procedure is the same, but the number of BSDSs you must examine is the number of members in the group.

You repro the last BSDS archive for each member to its “real” BSDS. You then run Print Log Map specifying the Group option and examine each set of logs. If this point had been created when you issued the command -ARC LOG GROUP, the logs for the entire group will be close in time (and LRSN). Otherwise you must run Print Log Map on each one and determine the most current LRSN common to all members. We illustrate with an example similar to 2.5.1, “Choosing an RBA for normal conditional restart” on page 35.

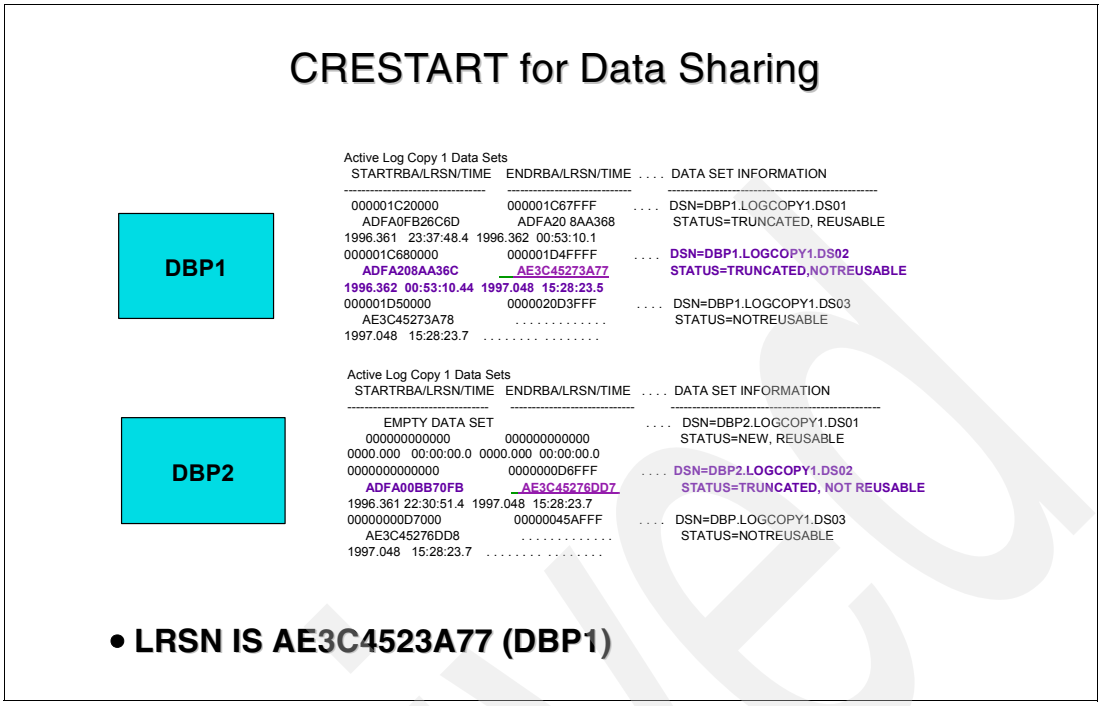


Figure 21-4 Finding the ENDLRSN to use in a conditional restart

Relative to the abbreviated Print Log Map output shown on this figure, we look down the right side for the DBP1 active log section. We look at the status. The first one is truncated, but reusable. This log was truncated the last time it was used, but it has already been archived (since it is reusable). It is not the one we want. The status of DS02 is truncated, not reusable. It meets both our criteria: truncation as a result of -ARC LOG, and not reusable, since the BSDS was offloaded while this log (now file 2 of the archive) was still active. DS03 is not reusable and it is the current active log (at the local site). So DS02 is the log we want and we record its ENDLRSN value. Note if two data sets are marked truncated, not reusable, the one which should be chosen is the one with the lowest ENDLRSN.

We go through the same process for the active logs for DBP2 and find that DS02 is the active log which meets our criteria. We record its ENDLRSN and compare the values from DBP1 and DBP2. The lowest one is that of DBP1. We must truncate both logs to that value.

Before doing that, we must subtract 1 from the ENDLRSN. We do this because when the -ARC LOG is issued, it writes a log record which is greater than 4 KB in order to force the write and then the offload. If the subtraction were not performed, DB2 would try to access a spanned log record whose range is not at the recovery site.

The conditional restart record looks like Example 21-7 and you must run Change Log Inventory on every non-dormant DB2 member using the same record.

Example 21-7 Conditional restart record for data sharing

```
CRESTART CREATE,FORWARD=YES,BACKOUT=YES,ENDLRSN=AE3C45273A76
```

DB2 restart will truncate any members' logs which have an ENDLRSN equal to or less than

This log has not yet been added to the BSDS. The data set name is the same one as is in the BSDS you recovered, with the exception of the low order qualifier, which is Annnnnnn instead of Bnnnnnnnn. This addition is shown in Example 21-8.

Example 21-8 Adding the last archive log to each BSDS for data sharing

```
DBP1:
NEWLOG DSNAME=hlq.ARCLG1.Dyyddd.Thhmmsst.Annnnnnn, COPY1VOL=xxxxxx,CATALOG=YES,
STARTRBA=00001C68000,ENDRBA=000001D4FFFF
STARTLRSN=ADFA209AA36C,ENDLRSN=AE8C45278A77

DBP2:
NEWLOG DSNAME=hlq.ARCLG1.Dyyddd.Thhmmsst.Annnnnnn, COPY1VOL=xxxxxx,CATALOG=YES,
STARTRBA=00000000000,ENDRBA=00000000D6FFF
STARTLRSN=ADFA00BB70FB,ENDLRSN=AE3C45276DD7
```

You must add the second copy of the archive log and active logs to the BSDS. This activity is not shown.

All the Change Log Inventory statements can be added in one invocation of the utility.

After the updates have been made, you can REPRO each BSDS to its second copy. If you prefer, you can REPRO before the update and include both BSDS data sets in Change Log Inventory job.

Many users restore some number of archives to DASD. This has two advantages: faster recovery due to direct reads of the archive, and parallel recoveries, as multiple jobs can share one DASD archive. Another benefit is that a long running unit of recovery can progress much faster if it doesn't have to read a tape backward (this only comes into play during restart).

If you are going to restore TAPE archives to DASD for disaster recovery, make sure the BLKSIZE specified in DSNZPARM is 24576 default with V8 when the archive was created, so that you can fit two blocks per track (the default is 28 KB and wastes DASD at a time when you are more likely to be constrained). The archive log block size cannot be changed after the data set was created.

21.6.4 Restart versus Restart Light

DB2 Version 7 introduced a new flavor of DB2 restart called Restart Light. This type of restart is useful in data sharing if you have an LPAR failure and the DB2 member that was running on that LPAR has retained locks in the coupling facility. Restart Light allows you to restart the DB2 member on another LPAR for the purpose of freeing the retained locks, then the member is terminated. The “light” restart requires a much smaller amount of storage than is usually required to restart a DB2 member, thus the member can be restarted on a different LPAR that would not normally be able to handle that DB2 member. No new work is accepted; the purpose of the restart is to free the retained locks. Once the locks are freed, the member is automatically terminated by DB2. You can now bring that member up later on the original LPAR once it is available again.

Restart Light It is best employed in an ARM environment to restart a DB2 member on a failed image on another (or any other) image. The ARM advantage is speed of execution in accomplishing the restart. The JCL or option to be specified is LIGHT(YES). See Example 21-9.

Example 21-9 ARM policy statement coded for Restart Light option

```
RESTART_METHOD(SYSTERM,STC,'-crc STA DB2,LIGHT(YES)')
Where -crc represents the command recognition character
```

The DB2 member starts quickly with a minimum footprint in MAINT mode, releases all retained locks except for indoubt units of recovery, and remains active, but without accepting new work. The only work that can be performed is that of the commit coordinators to resolve the indoubt URs (DDF is started for that purpose only). SQL codes are issued if an attempt is made to form new connections. The DB2 is skinny; it needs no pools and consumes minimal buffer pool storage. You can read about it in *DB2 UDB for z/OS Version 8 Data Sharing: Planning and Administration*, SC18-7417. It shuts down automatically when all indoubt URs have been resolved. You may stop DB2 manually before that time if you wish. It shuts down with CASTOUT(NO) for best performance.

One other activity it does not perform is to close out SYSLGRNX entries for all open data sets for that member (whether shared or not). This activity can have some repercussions that are not so obvious, so we shall discuss them.

SYSLGRNX entries are used to “help” DB2 determine where updates for a given object are found on the DB2 log when an object is recovered. They are employed for the Recover Utility, not GRECP/LPL recovery. Without them, DB2 would pass all the log since an image copy because it didn’t know where the updates were.

Let us now look at the implications of Recovery following Restart Light or DEFER. From the above discussion, you may have discerned that neither Restart Light or DEFER closes out SYSLGNRX entries for a given data set. Is this a serious problem?

With “classic” disaster recovery, the main activity is the Recover utility, which produces very little log. After recovery, usually all objects are image copied, thus rendering meaningless any open SYSLGRNX entries. Therefore the burden of open ended SYSLGRNX entries for a data set which passes extra log is minimized.

The same observation can be made of most of the mirroring scenarios.

In one case we know, a customer wanted to use Restart Light (to avoid taking the time to close SYSLGRNX entries for thousands of open data sets) followed by normal restart to remove the remaining retained locks (V7). The customer was able to prioritize some 25 data bases out of 400 that were mirrored and needed to be active quickly (thus the mirroring). The remainder were to be recovered using standard image copy and recovery activities over the next few days in the event of disaster. This division reduced the total cost of the disaster recovery solution.

Restart Light does not close out open SYSLGRNX entries, but it does write a system checkpoint, as a normal restart does. The following normal restart occurs, but as a result of the checkpoint, there an no open data sets except those with any remaining retained locks.

The open SYSLGRNX entries remain zeros for all data sets from the original crash or disaster. As a result of activity on the critical data base data sets, significant DB2 log will be produced. Recovery for those second priority DB2 objects will now pass an extra amount of log from what they would if they had gone through a normal restart initially after the disaster, and this amount of log may be significant.

21.6.5 GRECP/LPL recovery recommendations

When GRECP/LPL conditions are set during DB2 restart, you must remove them. You do this normally by issuing START DATABASE commands, as it is much faster than alternate methods (Recover, Load etc.). The command reads a few minutes of the active log and applies all records that had not been externalized to DASD.

Best performance is obtained by the use of *Fast Log Apply,* a process enabled by the DSNZPARM LOGAPSTG in the DSN6SYSP macro, or the Log Apply Storage option of the DSNTIPL install panel. The default is 0, which means *do not use FLA*. The maximum value that can be specified is 100, which refers to megabytes of storage reserved in the DBM1 address space for FLA events. You should consider at least having a non zero value in a separate DSNZPARM for disaster for LOGAPSTG.

Each START command acquires a 10 MB log buffer and sorts the list of objects in the START command in database/table space/page/LRSN sequence. One subtask is created for each of the objects in the list, up to 98 (More than 98 objects will wait until one the subtasks are available). Not only are the log records ready to be applied in sequence, but also the pages of the data set are acquired through a process called list prefetch. This capability allows a recovery that is up to 10 times faster than without FLA. Without FLA, each log record is read sequentially, the applicable page is read synchronously into the buffer pool and applied. The next log record, which is likely for a different table space, is read, and the process continues.

Create your START commands so that you process multiple objects per command. Issue no more than 10 START DB commands per DB2 member (the 11th etc. will not use FLA). We recommend the commands shown in Example 21-10.

Example 21-10 Issuing START commands

```
-START DB(DSNDB01) SP(*)
-START DB(DSNDB06) SP(*)
-START DATABASE(db1) SPACE(*)
-START DATABASE(db2) SPACE(abc*)
```

You should always start the DB2 Directory (DSNDB01)and DB2 Catalog (DSNDB06) first, as we show in Example 21-10. Following their successful restart, issue the commands for other databases in priority sequence if you have one. If the number of objects in a data base is fewer than 90 or so, you can issue the database form of the command. If you have several hundred objects in a data base, then you should split them as shown in the last form of the command, making sure you don't exceed the limit of 98. In fact, it is better to restrict it to about 50 objects, which allows for future growth without your recalculating the number each time.

Note: While you predefine the list, at the time of the disaster the number of objects actually in GRECP/LPL will vary. In a list of 50, there may be none, or 5, but that is OK.

You should have Install SYSADM authority for the commands. If you do not, DB2 will check for your authorization, and if the Directory objects are in GRECP, it is likely to fail. Since Install SYSADM is defined in DSNZPARM, it is validated by that means.

21.6.6 CLOSE (YES) versus CLOSE (NO) data sets

The default for DB2 for CREATE or ALTER TABLESPACE has been CLOSE(YES). Historically, users have been reluctant to specify CLOSE(YES) for performance reasons, both the overhead of physically closing a data set and the overhead of physically opening the data set in the first transaction that accesses it. CLOSE(NO) is almost universally practiced by DB2 users.

DB2 V8 behavior (and DB2 V7 after the PTF for PQ69741 is applied) is such that if CLOSE(YES) is specified, the result is that many fewer data sets are likely to be open during restart after an abnormal termination of DB2, and thus the end restart phase will complete more quickly. If you can identify non-critical page sets, you may alter them to CLOSE(YES) to

obtain a faster restart after a crash or for disaster recovery. When the effort in a mirroring scenario is to restart quickly in order to meet a Service Level Agreement (SLA), this approach can reduce restart time.

When there is GBP dependency (1 updater, 1 reader) the following different behaviors occur:

- CLOSE=YES: When the next PCLOSET for the reader, there is a physical close of the object and it ceases to be GBP-dependent. The advantage is that the remaining member can update the data set in its local buffer pool. The disadvantage is that the next access for the reader goes through physical open.
- CLOSE=NO: When the next PCLOSET occurs for the reader, no physical close is performed. The data set updated by the remaining member stays GBP-dependent (SIX against reader IS page set P-lock) until it is stopped or until DB2 is stopped normally. The advantage is that there is no physical open on the next access from the reader.

APAR PQ69741 has implemented a change in the CLOSE YES/NO behavior to reduce the degradation caused by the need to re-open data sets which had been previously closed due to being inactive for a period of time.

In data sharing, any table space or index which has remained in a read-only state with no activity for a period of time longer than the pseudo-close interval (PCLOSET zparm) will be physically closed. This is done in an attempt to reduce data sharing overhead (the object can become non-GBP-dependent if it is closed on all but one member), but it has the disadvantage of requiring that the next access to the object must go through dynamic allocation and data set open again. This can be a significant amount of overhead, particularly if a large number of objects are accessed at the same time after a period of inactivity (for example, at the beginning of the business day).

The code which determines whether an idle page set should be physically closed has been modified to honor the CLOSE YES or NO attribute of the table space or index. The physical close will now only take place for CLOSE YES objects, while CLOSE NO objects may remain open indefinitely. This allows the user to control the behavior on an object-level basis (formerly, the only means of control was through the pseudo-close ZPARMs, and lengthening the pseudo-close interval to keep data sets open could have other undesirable effects.

CLOSE(YES) — This now means less impact on performance.

CLOSE(NO) — No really means no now, with this APAR, in a data sharing environment. If you want things to behave as they did prior to this APAR (in a datasharing environment), you need to change your CLOSE(NO) objects to CLOSE(YES).

Consider CLOSE(NO) for objects that you expect to be GBP-dependent most of the time and you do not want the extra OPEN/CLOSE overhead associated with CLOSE(YES). For objects that are infrequently GBP-dependent, CLOSE(YES) will likely provide better performance.

Once GBP-dependent, CLOSE(NO) objects will stay GBP-dependent until either DSMAX is hit and all CLOSE(YES) objects have been closed or until the CLOSE(NO) objects are manually closed as a result of a DB2 command or a DB2 shutdown. This can have an effect on the amount of data sharing overhead experienced for objects defined as CLOSE(NO).

Users will need to consider data sharing overhead versus open/close overhead and tune appropriately using the CLOSE() parameter.

22

Validation and performance

This chapter provides health checking suggestions for keeping the system stable and for continuous validation of the system set up for disaster recovery.

We also provide recommendations regarding how to set up your system to handle mass recovery in the most efficient manner. We mention the latest maintenance level we used at the time of this project.

This chapter covers the following topics:

- Health checking
- Mass recovery best practices
- Maintenance

22.1 Health checking

Continuous evaluation of system set-up and disaster recovery procedures is required in order to:

- Maximize the stability of the system
- Ensure that the procedures for disaster recovery are capable of recovering the system in accordance with the defined recovery time objective (RTO) and recovery point objective (RPO).

In this section we look at these topics:

- What is health checking?
- Monitoring system availability
- Testing recovery procedures
- Identifying single points of failure

22.1.1 What is health checking?

It is all very well to define and set up a system for continuous availability. However, without continuous monitoring of the system as well as periodic tests of the defined recovery procedures, there can be no certainty that the system is providing the continuous availability required by the business.

These monitoring and procedural validation activities are what is collectively called health checking. Also with IT, prevention is both better and cheaper than cure, hence the term *health check*.

It is obvious that health checking is something that is not peculiar to disaster recovery. But it should be equally obvious that if your company is dependent on a capability to restore system services at a certain level should disaster strike, then you must validate regularly that the recovery time and point objectives can indeed be met.

With any type of health checking, human or machine, you will be looking for two things. Searching for any signs of bad health is one aspect, and the other is looking for indicators or triggers that may over time lead to bad health. The most important things to look for are Single Points of Failure (SPoF).

So health checking consists of:

- Monitoring of system availability and factors impacting availability
- Regular testing of recovery procedures
- Regular system reviews to identify Single Points of Failures (SPoF)

22.1.2 Monitoring system availability

Apart from whatever other monitoring is performed, it is essential that factors which either directly or indirectly impact availability are continuously monitored. Some contributors impact availability immediately, while others may have to reach a certain level before availability starts suffering.

Loss of hardware or failing application programs are examples of events that impact availability immediately, while locking problems or intermittent I/O errors are examples of situations that will have a creeping impact on availability.

Here is a list of entities to monitor for availability:

- **DB2 objects in restricted states:**

  DB2 objects in restricted states clearly present an availability issue. Apart from the immediate problems associated with resources not being available, there may also be an impact on your disaster recovery setup or point-in-time recovery procedures.

  You may find it very difficult to investigate why some DB2 objects are in restricted states, having recovered your system to some prior point-in-time without access to the system log or DB2 job log.

  We recommend that you either remove any restricted states for all DB2 objects or carefully document why these objects are in the respective states so that proper action can be taken to remove the restricted states.

- **Table spaces with LPL entries and error ranges:**

  Pages registered either on the Logical Page List (LPL) or on an error range should be recovered as soon as possible by entering the START DATABASE command. With DB2 UDB for z/OS Version 8 this happens automatically.

  Pages should not be left in error for too long and should be recovered before backups for disaster or point-in-time recoveries are taken. As for DB2 object in restricted states, diagnosing why pages are on the LPL or in an error range may be close to impossible after system recovery unless the system log and/or the DB2 job log is kept or the situation resulting in the pages needing recover is well documented in some other fashion.

- **Long running URs:**

  Long running URs pose several problems to availability. Lock contention due to extended lock duration is the obvious one.

  Another implication of long running URs is potential long backout times in case of application abends. Even URs that do infrequent updates may cause problems during backout due to the amount of log data that will have to be traversed to read the few log records that are to be backed out. This will get further aggravated in the case where the log records have to be read from archive logs.

  Furthermore, any repositioning of updated non-DB2 data sources by the UR may also impact availability.

  Long running URs generally come in two flavors:

  - One possibility is the case where few updates are done throughout the URs execution. The problem is not the number of updates that need to be backed out should the UR get cancelled, but rather, the amount of needless log data that needs to be traversed to roll back the few updates.
  - The other version is the UR that does massive updates without intermittent commits. Not only will there be numerous log records to read, but also, many updates will obviously need to be rolled back.

  DB2 for z/OS provides three parameters in DSNZPARM to control the monitoring of these two types of URs:

  - The URCHKTH parameter (also referred to as “UR CHECK FREQ” on the DSNTIPL installation panel).

    This parameter specifies the number of DB2 system checkpoints that is allowed to be taken before a UR is expected to commit. For instance, if this parameter is specified as 5, then a warning will be issued for every UR that have not committed for the last 5 system checkpoints. The purpose of this parameter is to catch long running URs doing only a few updates without committing.

The warning issued takes two forms. Message DSNR035I is issued identifying the offending UR and showing the number of system checkpoints that have been taken since the UR last issued a commit. If system checkpoints are taken based on elapsed time, it is easy to calculate the time since the last commit. Additionally, DB2 will write an IFCID 0313 trace record if Statistics Class 3 is turned on. This record can then either be read after the fact as part of the normal post-processing of SMF data or it might get picked up immediately by the on-line monitor you have available for prompt reaction.

We recommend URCHKTH be set to 5 as a starting point. In order to set a sensible URCHKTH, it helps if DB2 system checkpoints are taken based on time, rather than number of log records. If locking is your primary concern, you may consider lowering the value to catch URs that result in many lock waits or time-outs. If the roll back time is your main concern then you could increase the value.

– URs that perform massive updates without committing can be caught using the URLGWTH parameter of DSNZPARM, also known as “UR LOG WRITE CHECK” on the DSNTIPL installation panel.

  This parameter specifies the number of DB2 log records that you will allow before DB2 should issue a warning. For instance, if you specify 20 (the value is specified as a multiple of 1000), this requests DB2 to issue a warning message whenever a UR has caused more than 20,000 log records to be generated without issuing a commit.

  The warning information is the same as described above. Message DSNJ031I is issued identifying the offending UR together with the number of minutes since the last commit. Again, if Statistics Class 3 is turned on, DB2 will also produce an IFCID 0313 trace record.

  We recommend that you set URLGWTH to 10 as a starting point. Assuming the average update generates 4 log records (1 for the data row plus 3 for index changes, free space management etc.) then 10,000 log records represent 2,500 updates and at least the same number of locks.

– The LRDRTHLD parameter in DSNZPARM, introduced with DB2 V8, provides an additional threshold for catching Reader URs holding claims for a long time.

  It is also known as ‘LONG-RUNNING READER” in the DSNTIPE installation panel, and specifies the number of minutes that a UR is allowed to hold a read claim without committing. DB2 will not issue a message to identify Reader URs exceeding the threshold. You should consider specifying this threshold is you have problems getting DB2 utilities executed due to long running Reader URs holding Read Claims on resources.

  As opposed to the previously described two thresholds, DB2 will not issue a warning message. With Statistics Class 3 turned on, DB2 will write a IFCID 0313 trace record. If you are in a SAP installation, you can use the SAP transaction DB2 to have this trace record formatted and displayed. If you are not a SAP user, you may either catch any IFCID 0313 trace records as part of your daily batch post-processing of SMF data or you may consider writing an Instrumentation Facility Interface (IFI) program to format and display the IFCID 0313 trace records as they are produced to allow you immediate reaction.

Having set these thresholds is not enough, though. It is essential that you monitor the warning messages that are generated should either of the thresholds be exceeded. You can either catch the messages issued by DB2 (DSNB260I, DSNJ031I and DSNR035I) with an automation tool for someone to react to, or have your on-line monitor catch the IFCID 0313 trace record. Either way, it is important that all cases of exceeding any of these thresholds be investigated and action taken to correct the offending programs.

- **Virtual storage usage:**

  Over-utilization of virtual storage is currently probably the most serious threat to system stability. It is therefore essential that you perform capacity planning for virtual storage just like you do for CPU usage, real storage etcetera. It is important to understand that virtual storage usage is not confined to looking exclusively at DB2.

  You need to make a total system virtual storage budget that defines how you want to allocate virtual storage to the various users of this critical resource. Virtual storage should then be monitored regularly to verify that the different users stay within their defined boundaries and to catch any growth tendencies that may need a reevaluation of the virtual storage budget. DB2 should be regularly monitored using z/OS related tools like RMF together with the IFCID 0225 trace records.

  The *DB2 UDB for z/OS Version 8 Installation Guide,* GC18-7418 contains detailed information on how to calculate DB2 virtual storage usage in Chapter 2-1. This in combination with the data contained in the IFCID 0225 trace record will allow you to evaluate how DB2 virtual storage usage is made up and take corrective action.

  Check APAR II10817 frequently as this APAR contains a storage usage fixlist for DB2 Versions 6, 7 and 8.

  DB2 Version 8 is the first version of DB2 to provide 64-bit addressing. This has allowed some storage areas to be moved above the 2 GB bar. This includes the database buffer pools, EDM pool, Sort pool RID pool, castout buffers and compression dictionaries. The exact savings will depend on how your DB2 system is currently set up. If you are using dataspace buffer pools then clearly your savings will not be as great as for an installation using virtual buffer pools. Similarly, no savings is to be gained from moving the compression dictionaries above the 2 GB bar if you are not presently using data compression.

- **Maximum number of threads:**

  It is very important that you do not allow more threads to be active than DB2 can actually handle with the amount of virtual storage and pools available.

  The two parameters to watch out for are CTHREAD and MAXDBAT of DSNZPARM. They are also known as “MAX USERS” and “MAX REMOTE ACTIVE” respectively from the DSNTIPE installation panel. If the sum of these two parameters is set too high, DB2 will allow threads to get allocated even though virtual storage might not be available to support these threads. it is important that you monitor virtual storage usage and particularly keep a close eye on the amount of available storage. You want to keep about 200 MB free as a storage cushion so that DB2 can allocate temporary storage when needed.

- **Preventive maintenance:**

  You should closely follow what is being made available through the maintenance stream. Certainly any HIPERs against DB2, z/OS, DFSMS, CICS etc. should be applied fairly quickly to your test or maintenance system. This way, if a given fix need to be applied to the production system quickly for whatever reason, it should already have been tested on the test system.

  Apart from HIPERs, any fixes that have to do with data corruption or integrity and availability or system availability should be carefully evaluated for inclusion in the next maintenance cycle.

- **Automation:**

  Automation is key to monitoring of messages issued by the various systems. It is simply not feasible to sift through the massive amount of messages manually.

In most cases, a list of known message identifiers is defined to the automation tools. These messages are then processed by automation. But what message identifiers should be included in the list? Any message with a suffix of “E” (for Error) would be a good candidate. However, often warnings or informational messages may require action immediately or at least an follow on investigation. The messages produced by the long running URs described above are examples of messages that require some action.

We suggest that two separate lists be defined. One list is used to contain identifiers of messages that need action or investigation by the operational staff, DBAs or systems programmers. The other list is used to contain identifiers of messages that we know we are positively not interested in from a monitoring point of view.

Any given message will either belong to one of these lists or neither. If they are on one of the lists, it is defined what to do with the message: process or ignore. If they are not on either list, then the message should be routed to a queue that should subsequently be investigated to determine which of the two lists the identifier should be added to. This is the only way to ensure that new messages don’t fall through the cracks.

Automation should either initiate corrective action or leave the message on a log for further investigation. However, if the number of messages left is very large it is possible that the person(s) reviewing the log may either miss messages or maybe even give up. It is therefore very important that some scheme be set up that will allow the reviewing team some sort of structured approach in analyzing the log. A summary of the messages on the log with the ability to drill down for more detailed analysis of individual messages would be one way.

Identifying problems is the first step, but not enough. It is imperative that someone be responsible for fixing the problem. If several URs exceed one or more of the long running UR thresholds, something must be done to fix the problem.

22.1.3 Testing recovery procedures

Recovery procedures need to be developed and documented for at least three reasons:

- The first reason is that reviewing the procedures for completeness and correctness obviously requires some sort of documentation to review.
- The second reason is that when the recovery procedures are to be used in real life it is not given that the person skilled in this type of activity is available. This is particularly true in case of disaster recovery. Therefore, well documented recovery procedures provide the only way to ensure that even non-specialists can perform the recovery.
- The third reason is that without a documented procedure, there is no way to check that the procedures actually work and cover all aspects of recovery.

To paraphrase a famous Goldwynism: undocumented recovery procedures are not worth the paper they are written on.

Assume that the recovery procedures are documented. How do you verify that they are complete and correct? Or that the Recovery Time (RTO) and Point (RPO) objectives are met? The only answer is by testing the recovery procedures using the documentation as the script for the process. So verifying the documentation by actually trying to recover using it is the first step.

The follow-on steps are the verification that the recovery procedures keep working with changing system set-up, new software releases, new recovery utilities and tools, new applications, changing priorities etcetera.

Just testing the procedures may not be enough. Tuning of the procedures may be required to meet the RTO. Adding new applications or increasing the amount of data stored may also require the procedures to be tested to verify that they can still meet the required RTO.

Finally, assuming that you need to continue working at the recovery site, you need to add procedures for maintenance, development, and testing.

22.1.4 Identifying single points of failure

Single points of failure (SPoFs) are inherent dangers to your system's health. It is therefore of the outmost importance to identify any SPoFs you might have in your system set-up. Having identified the SPoFs, you now have the choice of either removing them or, if it is either impossible or too expensive to remove them, to carefully document the exposure and design efficient procedures to recover from a particular SPoF.

For many years now, technological advances have made it possible to remove a large number of what were previously SPoFs. This whole redbook is all about some of these new technologies, which give you options to change what might otherwise for your company be a life threatening situation, into what might be called an extended recovery situation. DB2 itself provides many functions and features for removing SPoFs. We just mention a few: dual logging, data sharing, multiple stored procedure address spaces and, of course, all the recovery/restart functions.

It is important, however, to understand that DB2 cannot prohibit you from creating SPoFs in your application design. Next we list some pitfalls in application design and set-up that may lead to SPoFs:

- **System affinities:**

  Suppose that you have a DB2 data sharing group set up and a set of CICS subsystems such that you can run your transactions anywhere. This will give all transactions the theoretical ability to run anywhere and, should one CICS or DB2 fail, there would be another pair to take over the workload.

  All it takes for this *not* to be true, is a situation where a given transaction must run on a single or a subset of all pairs due to some affinity in the application set-up. One such example would be an application that needs access to a WebSphere MQS that does not run on all available CICSs.

  Another less likely case would be an application that accesses a DB2 table which cannot be shared across DB2 subsystems.

- **Data affinities:**

  Some applications may require access to data outside DB2 that cannot be shared across z/OS systems. Examples of this would be sequential data sets used by batch applications.

- **System symmetry:**

  The ability to freely move subsystems and application systems around is dependent on some level of identical set-up of the systems to which these subsystems are being moved.

  Assume, for instance, that one of your z/OS systems has less private storage, SQA, etc. available than the other participating z/OS systems because some subsystem like WebSphere is running on that z/OS system. The implication might be that not all applications or subsystems can be moved to this z/OS due to storage limitations.

  Asymmetrical system set-ups may be another example. Moving an application from one member of a DB2 data sharing group might not be immediately possible for reasons like the size of the DB2 virtual buffer pools or the maximum number of threads defined for the target DB2 subsystem.

22.2 Mass recovery best practices

Mass recovery is a term used to describe recovery of a large number of DB2 objects using the standard DB2 Recovery utility, as opposed to employing the techniques described in other parts of this book.

We also discuss considerations for setting your system up to allow optimal DB2 log based analysis for application recovery and data diagnosis. Some suggestions are of a kind that will be contrary to what is considered best practices for other aspects of system setup. As one example, one consequence of using DB2 compression techniques is that DB2 log data will be compressed; this complicates log analysis or makes it downright impossible.

You may consider mass object recovery in a situation where you are either not using the various hardware and software duplication techniques described in this redbook, or where these techniques for one reason or another do not work.

You may consider log-base recovery in a situation where the standard DB2 recovery facilities, for one reason or another, do not work.

22.2.1 Introduction

Recovery of DB2 table spaces consists of two phases:

- The first phase of recovery is the restore of a full Image Copy followed by any available incremental Image Copies.

  The full Image Copy may either be a normal DB2 Image Copy or a DFSMSdfp Concurrent Copy. Either one must have been produced using the DB2 COPY utility. Incremental Image Copies are not available with DFSMSdfp Concurrent Copy.
- The second phase of recovery is to apply any updates to the table space done after the last incremental Image Copy was taken.

  These updates are available on the DB2 log. The log will need to be scanned starting at the time the Image Copy was initiated until the end of the log unless a point-in-time (PIT) recovery is wanted.

There are basically three ways to improve the performance of the restore phase:

- Increase the speed with which the Image Copy is read. One example is data striping of the Image Copy.
- Decrease the size of the Image Copy data set. One example is compression.
- Restore Image Copies in parallel.

Similarly, the time to read log records can be reduced by:

- Increasing the log transfer rate. One example is data striping of the log.
- Reducing the amount of log data to be read in order to get the relevant records. One example is DB2's registration of log record ranges in the SYSIBM.SYSLGRNX.
- Reading log records only once while using them for multiple table space or partition recoveries. One example is Fast Log Apply.

Applying the log records is a question of normal DB2 performance tuning even though some special considerations do apply.

Several features and options are available to reduce the elapsed time of the recovery. These include:

- **Parallel restore of Image Copies:**

  DB2 supports parallel restore of Image Copies, limited only by Image Copies devices and virtual memory.

  You can either let DB2 determine the optimal number of parallel restores or specify the number wanted when requesting DB2 to recover table spaces or partitions in parallel.

  Also, the maximum number of tape devices to be used at any point during restore can be explicitly specified, thus prohibiting DB2 from monopolizing the tape device usage.

- **Skipping log data sets without relevant data:**

  The SYSIBM.SYSLGRNX table contains information identifying down to the partition level the log ranges with updates. This in combination with the bootstrap data set (BSDS) will allow recovery to skip portions of the log that do not contain data for the table spaces or partitions being recovered.

  Various options give the user some control over the granularity with which the log ranges are recorded.

- **Taking Image Copies of indexes:**

  DB2 allows indexes to be copied; this provides a fast recovery alternative to the traditional index rebuild.

22.2.2 Recommendations for fast recovery

Here is a list of options available to obtain the fastest possible recovery:

- **Allow the Recovery utility to restore table spaces in parallel:**

  Specifying the PARALLEL parameter will cause DB2 to restore table spaces in parallel. This is something you definitely want to exploit. To get full advantage of this option you will have to image copy to disk or at least make sure that multiple image copies do not go to the same tape.

  Image Copies resident on tape may require special considerations. You may either let DB2 calculate the number of tapes drives required or specify a number yourself. Letting DB2 do the calculation may lead to overcommitment of tape drives particularly in the case where several restore jobs are executed concurrently. For this reason, you should calculate the number of tape drives you have available to the individual job yourself and specify that with the TAPEUNIT subparameter of the PARALLEL keyword.

  Refer to *DB2 UDB for z/OS V8 Utility Guide and Reference*, SC18-7427 for syntax and details.

- **Consider using Image Copy to disk:**

  Image Copies on disk allow for the highest degree of parallelism. This is particularly true where multiple Image Copies are written to the same tape volume. In this case, the Image Copies can only be restored serially.

  It is important to remember that on a VTS there is still a serialization taking place at the tape volume level so multiple Image Copies on a volume is also a problem for VTS. While VTS capabilities are designed for easy, inexpensive backup, it is less efficient for mass object recovery as there is likely to be far more data to be recovered than paths exist. While users *know* VTS is writing to DASD, the offload occurs to tape. If an archive log resides on a tape, the entire tape must be restored to the VTS *DASD* before any of it can be read. Additionally it is an actual tape UCB to DB2 and must be read serially, as a tape.

- **Consider generating dual Image Copies if using tape:**

  Truly tape-based Image Copies may for one reason or another be unreadable, which means that recovery will have to go back to the prior Image Copy and then additionally have to apply all log records between this prior Image Copy and the unreadable one. This will result in a longer log apply phase and thus an overall longer recovery time.

  To reduce the risk of this happening, you should write dual copies of each Image Copy.

  Writing dual copies gives you another possible benefit if you write more Image Copies to each tape volume. Stacking multiple Image Copies to the same tape volume means that restoring the Image Copies will be a serial process. However, with multiple tape volumes containing the same Image Copies you will be able to restore the Image Copies in parallel tracks.

- **Take frequent Image Copies:**

  The more frequently Image Copies are taken, the smaller the range of log records that need to be scanned during the Log Apply phase. Consider taking Image Copies frequently enough to ensure that whatever log range needs to be scanned will normally be available from the active log.

  The active log supports a higher degree of parallelism that the archive log both by implementation in DB2 of load balancing across the two active log data set copies but also since it is always disk based.

  The frequency of copying must then take into consideration the number and sizes of the active logs plus the amount of log data to be applied. The more log data to apply the more frequent the Image Copy rate. This is both to reduce the amount of log data to process and to ensure that the data is available on the active logs.

  As an example, assuming a set-up with 50 log data sets each 1500 cylinders equal to roughly 1 GB and a logging rate of 1.4 MB per second then it will take 15 minutes on average to fill up a single log data set. Wrap-around of the active log will happen after 12.5 hours. As long as the log read and apply time is less than 15 minutes per active log, Image Copy taken every 12 hours would ensure that the log data would be available on the active logs. Remember that full Image Copies would not be required for this to be true.

- **Consider using Incremental Image Copies:**

  Using Incremental Image Copies has a indirect impact on recovery. The purpose of Incremental Image Copies is to lower the cost of taking Image COpies by only copying those pages that have changed since the last Image Copy, whether full or incremental, was taken.

  However, lowering the cost of taking copies means that copies can be taken more frequently which again means that the amount of log data to process during restart can be decreased.

  Incremental Image Copies provide an attractive alternative to full Image Copies for the cases where the percentage of changed pages is low.

  Incremental Image Copies should preferably be written to disk for speed and parallelism and also to avoid the situation during mass recovery where there are not enough tape drives available for all the Incremental Image Copies required.

  Should the number of tape drives not be sufficient, then DB2 will revert to recovering the table spaces involved from the log.

  The recommendation for using Incremental Image Copies in stead of Full Image Copies has traditionally been when the number of updated pages is 10 percent or less of the total number of pages. However, especially to improve performance of recovering many table spaces and indexes in parallel, a higher percentage may be considered to ensure the log data is available from the active logs.

Using Incremental Image Copies imply maintaining the Modified Page bitmap in the spacemap pages which particularly in data sharing can be detrimental to performance. The maintenance of the Modified Page bitmap can be turned off on a table space or partition level by specifying TRACKMOD NO. Refer to *DB2 UDB for z/OS V8 SQL Reference*, SC18-7426 for syntax and details.

- **Use the Merge Copy utility when producing Incremental Image Copies:**

  DB2 needs concurrent access to the Full Image Copies as well as any Incremental Image Copies. This should not be a problem if Image Copies are written to disk. However, the number Incremental Image Copies becomes a concern as the number of tapes devices required increases.

  If DB2 cannot access all available Incremental Image Copies, DB2 will stop requesting access to subsequent Incremental Image Copies and start the Log Apply following the last RBA or LRSN found on the last Incremental Image Copy. This will most likely result in a longer recovery time.

- **Consider using Recover for large indexes:**

  Recovery of indexes using a method identical to recovering table spaces may be much faster than rebuilding the index. Exactly how much faster will depend on the frequency of Image Copies, the frequency and percentage of index pages being updated. The option to produce Incremental Image Copies and the Merge utility is also available to indexes.

  Allowing Image Copy of indexes does not require Image Copies to be taken. Recovery is not possible for an index without Image Copies. Rebuild is the method of restoring these indexes. This fact may be used to set up a recovery scheme for the partitioning index of partitioned table spaces where some partitions are being updated and others are not. For read-only partitions Image Copy is a good choice since one Image Copy of each of the read-only partitions will allow fast recovery of the index partition. Whether the partitions that are being updated will benefit from using Image Copy is a question of the frequency and time distribution of updates.

  If indexes are to be recovered simultaneous with the underlying table space or partition then specify all indexes together with the table space or partition in the same recover statement. This way you ensure the fastest possible recovery since the log will only be scanned once which is particularly important in this case since log records for updates to indexes will be found in close proximity to the log records for updating the base table.

  In order to be able to use the Recover utility for indexes, you must do the following actions:

  – Specify COPY YES when creating an index or use the ALTER INDEX statement for existing indexes.
  – Periodically create Image Copies of these indexes.

  Both of these actions are required.

  When there is a need to recover a table space and one or more of the indexes defined for tables in the table space, be sure to specify the table space and all the associated indexes in the same Recover utility control statement to ensure that the logs are only passed once.

> **Note:** If you specify COPY YES, SYSLGRNX entries will be created on behalf of the index just as it occurs for table spaces. Many users do this so that object level LOGONLY recovery can be implemented. If you never make copies, SYSLGRNX entries can never be deleted (MODIFY requires at least ONE image copy) and SYSLGRNX will continue to grow, eventually going into extents. For this reason, you should take an image copy occasionally, so SYSLGRNX entries can be pruned.

- **Consider running recovery jobs in parallel:**

  Separating recoveries into several jobs may be attractive from several viewpoints.

  Firstly, it gives the potential for an even higher degree of parallelism than using the PARALLEL keyword since not only is the Restore phase run in parallel but so will the Log Apply phase.

  Secondly, the sequence and priority of recovery can be controlled to reflect the business importance of the table spaces to be recovered. By splitting the recovery across several jobs you can define performance goals to the z/OS Work Load manager (WLM) for different categories of table spaces to be recovered.

  Thirdly, recovery jobs can be run across the entire group in a data sharing environment which adds CPU capacity to the recovery process. Furthermore, since each job will get its own set of Fast Log Apply buffers more buffer space will thus be available resulting in a better over-all performance.

  The above considerations all assume that the log is available on disk. Log data sets on tape cannot be read in parallel. This is even true for logs on VTS.

- **Exploit Fast Log Apply:**

  Set up DB2 to use Fast Log Apply for maximum speed of processing the log.

  The Fast Log Apply function is available during Recovery utility executions, DB2 restart, and, with DB2 V8, also during the RESTORE SYSTEM utility. FLA is enabled by default during the Forward Log Recovery phase of DB2 restart.

  The Fast Log Apply feature has to be explicitly activated by specifying the LOG APPLY STORAGE as a number greater than 0. We recommend that this parameter be set to 100 (equal to 100 MB of Fast Log Apply buffer storage).

  With a Fast Log Apply buffer storage of 100 MB and each recover job taking up 10 MB of buffer space this allows for 10 concurrent recover jobs.

  The Restore utility will use 500 MB by default, if enabled.

- **Exploit parallel index rebuild:**

  DB2 allows you to rebuild indexes in parallel. You definitely want to use this capability for the cases where you're not using recovery of indexes.

- **Archive logs to disk:**

  Archive logs on disk allows parallelism in reading the log during the Log Apply phase. So in order to be able to run multiple recovery jobs requiring access to archive log data sets concurrently, the archive logs need to be on disk.

  It is possible to let DB2 archive to disk and then migrate these data sets to tape using HSM. The migration to tape can be specified to take place after a certain time has expired. If, for example, Image Copies were produced every 24 hours, HSM could be set to keep archive logs on disk for 24 hours, allowing recovery to be entirely based on disk data sets.

- **Specify log data set block size as 24576 bytes:**

  Specify a block size of 24576 bytes for the log data set whether you archive to disk or tape. This block size allows two blocks to go on each track on disk.

  If, during the analytic phase of a massive object recovery, you decide to copy your tape-based archive logs then a block size of 24576 bytes will not waste any disk space. When you do archive to tape then use a block size of 24576 bytes.

  The log data set block size is specified with the "BLOCK SIZE" parameter of the DSNTIPA installation panel and defaults to 28672 bytes for DB2 versions prior to Version 8.

  With Version 8, the default is 24576 bytes.

- **Maximize data available from active logs:**

  The fact that active logs are always on disk guarantees the highest degree of parallel processing.

  Another aspect is DB2s way to access the active log. DB2 will, when using dual active logging, balance the workload across the two copies of the active log.

  If enough space is available for the active log to allow DB2 to recover any table space without having to go to the archive log, then that will mean the best possible log read time. It will also mean that archive to disk becomes much less of a concern.

  If Image Copies were taken every 24 hours maximum, then it would be optimal to have enough space defined for the active logs to hold 24 hours worth of log data.

- **Increase accuracy of the DB2 log range registration:**

  DB2 uses the information in the Directory table SYSIBM.SYSLGRNX to minimize the amount of log data to read. In this table DB2 registers the start and end RBAs/LRSNs for log activity against each table space or partition. Information about indexes will also be maintained if the index has been enabled for Image Copy.

  During recovery, DB2 will only read records within the log ranges found in the Log Range table and therefore potentially be able to perform the recovery faster.

  The start RBA and LRSN will be that of the first log record referring to the given table space, index or partition. The end RBA and LRSN will register the ceasing of update activity against the table space, index or partition. DB2 allows you to control when update activity is to be considered ceased.

  Two controls are available. You can either specify a number of system checkpoints (the PCLOSEN parameter of DSNZPARM also known as “RO SWITCH CHKPTS” on the DSNTIPL installation panel) or a time period (the PCLOSET parameter of DSNZPARM also known as “RO SWITCH TIME” on the DSNTIPL installation panel). If no update activity has taken place against a given DB2 object for either the specified number of checkpoints or the time period, DB2 will consider the table space, index or partition as read-only. This causes the Log Range table to be updated with the RBA address and LRSN of the last log record referring to the object and thus terminate the log range.

  The default for PCLOSEN is 5 checkpoints and for PCLOSET is 10 minutes. If DB2 system checkpoints are being taken based on time these two parameters overlap and either one may be used to drive the Read-Only Switching, also called Pseudo Close, process. Refer to “Pseudo Close” on page 33 for more details.

  What would be a good value to specify? Since a major purpose of these controls is to reduce the amount of log data needed to be read when recovering a table space, partition or index, the major benefit would be if entire log data sets could be skipped. This is particularly true for archive logs on tape. Hence, it would make sense to trigger the Read-Only Switching process with an interval that is at most equal to the shortest time it takes to fill an active log data set or even better half of that time.

  It would be a fairly simple task to write a REXX to post-process the report produced by the List Log Inventory utility, DSNJU004, to calculate how long it takes DB2 to fill an active log data set and based on this value work out what would be a good value to specify in your environment. Refer to Appendix A.2, “BSDSSTAT: Log fill time statistics from BSDS” on page 460 for an example of such a procedure.

  Every time the interval expires DB2 will check for tables not having been updated within the interval. For any table where this is the case the log range will be closed by updating the SYSIBM.SYSLGRNX table. So there is a fair amount of activity going on to both check all tables for their status plus the updating of the SYSIBM.SYSLGRNX table. Therefore, you want to set an interval where the incurred overhead is justified by the savings in recovery time. This means that you should not consider the interval as a permanent

specification, but regularly monitor both the number of log ranges actually being closed plus the time it take to fill your active log data sets and then adjust the interval accordingly.

Refer to the *DB2 UDB for z/OS V8 Installation Guide*, GC18-7418 for syntax details.

- **Consider using I/O striping for the active logs:**

  It is possible to use the DFSMS Data Striping facility with the DB2 active logs.

  DFSMS Data Striping is a technique for speeding up sequential processing by spreading a logical data set across several physical data sets and reading and writing these physical data sets in parallel. Splitting a data set up into two stripes for example will theoretically halve the read time of the data set.

  Up to 16 strips can be defined but tests have shown that depending on the environment, more than 7-8 stripes will not benefit performance.

  See the instructions for installation panel DSNTIPH in *DB2 UDB for z/OS Version 8 Installation Guide*, GC18-7418 for instructions on activating dual logging. Refer to *Appendix B: Enabling VSAM I/O Striping in OS/390 V2R10* in the redbook *DB2 for z/OS and OS/390 Version 7 Performance Topics*, SG24-6129 for instructions on enabling VSAM I/O striping. For possible performance enhancements when using FICON, striping, and dynamic PAV for DB2 log data sets, refer to Chapter 2 of the redbook *DB2 for z/OS and OS/390 Version 7 Selected Performance Topics*, SG24-6894.

- **Consider using the DB2 Log Accelerator tool to provide striping and compression support for the active logs:**

  DFSMS Data Striping is not directly available for use by archive logs due to the way DB2 access the data sets. However, the DB2 Log Accelerator tool provides the support needed to use Data Striping even for archive log data sets.

  The DB2 Log Accelerator tool also allows the archive log to be compressed either using DFSMS compression or compression delivered by the tool itself. Early measurements show that using DFSMS compression produces benefits in space (about 50% reduction) and recovery execution (about 30% elapsed time).

- **Evaluate Data Compression for reducing log data volumes:**

  The purpose of DB2 Data Compression is to reduce the amount of space being taken up by user data. The compressed data is what DB2 works with which means that the data on disk is compressed and, as a consequence, so is the data in the various database buffer pools and on the DB2 log.

  So compressing the data will also reduce the amount of log data that is written. This is particularly true for INSERTs and DELETEs where the entire compressed row is logged. This is also true for UPDATEs to tables with Changed Data Capture turned on

  To illustrate the effect, assume a row size of 750 bytes and a compression ratio of 33 percent. An INSERT into this table will cause a log record to be built that contains DB2 header information of 64 bytes plus 500 bytes for the compressed row image or a total of 564 bytes. Without compression, the same log record would have a length of 814 bytes (64 bytes header plus 750 bytes row image). The net effect of compression on the log for this one record is 30 percent saved space.

  Assuming an average saving of 25 percent on the database change log records and assuming that the database change records account for 60 percent of the total amount of log data, turning on compression for all table spaces would result in reduction of log data of 15 percent.

  The use of DB2 Data Compression is not free of charge since a compression dictionary is required for each table space or partition for which compression has been requested. This dictionary is 64 KB in size and prior to DB2 UDB for z/OS Version 8, the compression dictionaries were loaded below the 2 GB bar. So the deployment of DB2 Data

Compression has to be done judiciously with the potential performance improvements in terms of buffer pool hit ratios and log data reduction carefully weighted against increased usage of virtual storage and CPU.

- **Buffer pool tuning:**

  Tuning your buffer specially for massive recovery is a possibility that is also available to you. For instance, if you have a DB2 system that is heavily insert based, particularly if these inserts have a sequential insert pattern like time-series, reducing the write threshold may be a very good idea. The idea is to force pages to be written before the buffer space they occupy is needed so that as many writes as possible may be asynchronous.

  You may want to experiment with both the deferred and vertical deferred write thresholds. These can be set by the ALTER BUFFERPOOL command parameters DWQT and VDWQT respectively.

  In data sharing, you may additionally force buffer flushing by lowering the group buffer pool checkpoint interval.

  Do remember that these three threshold apply to a particular buffer pool, and that you therefore may have different thresholds for different buffer pools.

  Particularly, the usefulness of tuning the buffer pools for massive recovery greatly depends on the update patterns of the objects assigned to them. Also, if your DB2 system is not completely dedicated to the mass recovery task, performance requirements of the running applications may not allow you to change the buffer pool set-up.

  The point to be aware of is that massive object recovery will cause a lot of page writes over a short period of time. If large buffer pools are used, you will probably want to trigger page writes more frequently than with smaller pools to avoid big bursts of write activity. For these reasons, you may consider setting the DWQT to some value equal to or less than 10 and the VDWQT to 1 as a starting point during that time.

22.2.3 Special considerations for log-based recovery

Any time this type of recovery has to be undertaken, the situation is going to be different, and so is the actual recovery process. Again, it is beyond the scope of this book to provide suggestions on how to perform this type of recovery. Furthermore, since the scenarios that lead to this kind of recovery will most likely be of the kind “this should not happen”, it will be impossible to provide a standardized guide.

However, it is true to say that there are some general recommendations that will reduce the processing time for no matter what type of recovery you may need to perform, assuming it is log-based.

Typically, when you find yourself in this situation, you first have to diagnose the problem to determine what is wrong, what table spaces were hit and how extensive was the damage. Secondly, you have to decide on a recovery strategy, and finally you have to execute the chosen recovery.

The first phase will in all likelihood require the DB2 log as the base for the analysis. Furthermore, it is possible that a lot of log data have to be traversed to find the problem and determine the extent of the damage. Additionally, several runs may be required.

The net result is that the diagnostic phase may well take much longer than the actual recovery. For this reason, it is important to optimize the log reading time. The log reading time can be split in two. One part is positioning: finding the place where reading is to start; and the other part is the actual reading time.

It is important to understand that some of the following recommendations are specific to log-based recovery and may be in conflict with recommendations for achieving high performance or even mass recovery optimal performance.

22.2.4 Recommendations for log-based recovery

Here is a list of recommendations for this type of recovery:

- **Make archive logs disk resident:**

  As with mass recovery, logs on disk can be accessed in parallel, allowing several diagnostic runs simultaneously. This allows more people to work productively on problem determination.

  The argument is the same as for mass recovery in 22.2.3, “Special considerations for log-based recovery” on page 445, with the added concern that since DB2 might still be running and producing log records active logs might get archived. This means that it is possible that what started out as a completely active log based investigation may turn into requiring access to archive logs as time goes by.

- **Maximize availability of log data on active logs:**

  The argument is the same as for mass recovery in 22.2.3, “Special considerations for log-based recovery” on page 445. The major benefit is active log read versus archive log read (even for DASD archives), as the active log reads are much faster. They should have as much active log “capacity” as possible for better performance. However, since DB2 may still be active, there is no guarantee that log data will be available from the active log throughout the process.

  The size of the active log data sets is not an issue. A smaller log data set would reduce the binary search process a bit on the LRSN range positioning (first) read, while larger log data sets would be beneficial reducing the switching of data sets covering the required log range.

  DB2 V8 allows up to 93 active log data sets with a maximum of 4 GB each, where prior releases only supported a maximum of 31.

- **Consider using Changed Data Capture (CDC) more aggressively:**

  The logging of SQL UPDATEs will not necessarily contain an image of the complete row being updated. During recovery base on the log, this means that the recovery process will either have to go back to the log record for the initial insert of the row or pick up whatever image of the row can be found on the last valid Image Copy and then reconstruct the current image of the row by applying all subsequent log records to the row. Either way, this may be very time consuming and may require the recovery process to go back a long time on the log, maybe even several days depending of frequency of Image Copies.

  However, this will not be the case if Changed Data Capture (CDC) is requested for the table. Turning on CDC will cause DB2 to log the complete image of the row and thus remove the need to go back to rebuild the current row image.

  The CDC facility becomes even more interesting during the diagnostic phase if DB2 Data Compression was defined for the table space. Without CDC, it is not possible to decompress UPDATE log records which means that analysis will probably be impossible.

  Having both CDC and DB2 Data Compression active compared to both of them inactive will in many cases be log data neutral.

- **Consider the use of DB2 Data Compression:**

  The use of DB2 Data Compression makes the diagnostic phase more complex since decompression will have to be performed in order to analyze the user data in the log records.

Even with complete row images you might still not be able to print the uncompressed image of the user data, depending on the tool you use to print. DSN1LOGP, for example, does not decompress the row image when printing database change log records.

Also, the log analyzer tool you use to rebuild the tables might not support DB2 Data Compression which make it unusable for compressed tables.

So purely from the point of view of log-based analysis and recovery, DB2 Data Compression may not be a good thing.

You should carefully consider your strategy for building the dictionary used by DB2 Data Compression if you either use or are thinking of using DB2 Data Compression. Running DB2 Reorg without KEEPDICTIONARY specified will cause DB2 to generate a new compression dictionary as part of the Reorg process. This means that database change log records produced after the Reorg will require the new compression dictionary to be decompressed. This dictionary will be available from the Image Copy taken as part of Reorg or immediately after it. If an Image Copy containing this new dictionary is not available for whatever reason then any database change log record following Reorg cannot be decompressed and log-based recovery will thus not be possible.

It is for this reason that we recommend that Reorg always be run with KEEPDICTIONARY unless you have determined a need (such as change in the data) to create a new dictionary. It should be noted that the effect of generating a new compression dictionary may often not be very large unless the actual data content (the bit strings, if you like) changes a lot over time.

- **Tools can be essential:**

  DB2 does not come with utilities to perform a log-based recovery. The underlying assumption of DB2, very reasonably, is that at least one good Image Copy is available and that the DB2 log is available and usable. However, the reason for having to do a log based recovery is that one or both of these assumptions cannot be met. Therefore, tools available outside DB2 are very useful.

  Depending of the table and table space options chosen you may need both a tool to print the log to allow diagnosis of the activities recorded on the log and certainly a tool to extract row images from the log and then reapply these changes.

  The tools you need should as mentioned above support the table and table space options that you have chosen. So, for instance, if you have turned on DB2 Data Compression, then the tool must support that feature to be useful to you.

  Most of the log analysis tools available on the market have been developed to allow recovery of rows erroneously changed or removed by either bad application programs or accidental deletes or drop of tables or table spaces. The underlying assumption of most of these tools are that the amount of data to recover is small since if not, a point-in-time recover to a point before the erroneous process began would in most cases be faster.

22.2.5 Tools for processing the DB2 log

There are three tools that should be considered for analyzing the log and speeding reading of the log, particularly the archive log. These tools are:

- **IBM DB2 Log Analysis Tool for z/OS:**

  The IBM DB2 Log Analysis Tool for z/OS provides you with a powerful tool to ensure high availability and complete control over data integrity. It allows you to monitor data changes by automatically building reports of changes that are made to database tables. Latest enhancements include new advanced filters for more granularity and easier specification to output reports, the ability to audit table activity, and the ability to automatically select the most efficient way to run reports. In addition, the tool offers exclusion filters, Fast Path

processing, warning processing, and quiet time--a feature that allows you to determine periods of inactivity in your DB2 subsystem.

Refer to *DB2 Log Analysis Tool for z/OS Version 2.1 User's Guide*, SC18-9132 for a comprehensive description of the tool.

- **IBM DB2 Archive Log Accelerator for z/OS:**

  The IBM DB2 Archive Log Accelerator for z/OS reduces the overhead that is associated with database log management to balance the increases in archive log growth. The tool achieves this using a combination of data striping (for faster log volume through-put) and DFSMS hardware- or software-based log volume compression. The tool maximizes data availability by leveraging DASD storage that contains compressed archive logs.

  Refer to *DB2 Archive Log Accelerator for z/OS Version 2 User's Guide*, SC18-7405, for a comprehensive description of the tool.

- **IBM DB2 Change Accumulation Tool for z/OS:**

  The IBM DB2 Change Accumulation Tool for z/OS quickly restores database objects with precision and minimal disruption, setting the scope and specificity of image copy creation through the use of control cards. Latest enhancements include: ability to write changes to image copies and VSAM files, support for simultaneous creation of LP/LB and RP/RB image copies; ability to create image copies of catalog tables; a minilog control table maintenance function that enables selective deletion of rows.

  Refer to *DB2 Change Accumulation Tool for z/OS Version 1.3 User's Guide*, SC27-1191, for a comprehensive description of the tool.

22.3 Maintenance

In this section, we provide the most well-known current maintenance, as of the time of writing, applied on top of *our* DB2 level: the DB2 for z/OS V8 New Function Mode, service level PUT0406 (COR0407).

The APARs are listed in Table 22-1. The first APAR is for z/OS; all the others are applicable to DB2 V8.

Make sure to contact your IBM Service representative for the most current maintenance at the time of your own installation.

Table 22-1 Current disaster recovery related APARs

APAR #	Text	PTF and Notes
OW48234	This MVS APAR provides additional enhancements to the dump conditioning (DUMPCONDITIONING) support added to DFSMSdss by APAR OW45674. This support improves the usability of dump conditioning by eliminating the requirement to clip the volser following a full volume restore of a dump conditioned volume. This support also improves the capability of dump conditioning by allowing physical data set restore operations to be performed using a full volume dump of a dump conditioned volume as input.	UW79364, UW79365, UW79366 for DFSMSdss V1.4, V1.5, and V2 R10

APAR #	Text	PTF and Notes
PQ69741	Code changes to determine whether an idle page set in data sharing should be physically closed for CLOSE YES or NO attribute of the table space or index	Included in DB2 V8, applicable to V6 with UQ74865, and V7 with UQ74866
PQ86382	Abend04E RC00C90102 DSNIBHUN VRACE0C01 ERQUAL0C01 during LPL recovery	UQ86901
PQ86492	DSNIDBSE ERQUAL5006 RC00C00101 Abend04E	UQ87589
PQ87542	RC00C90101 DSNILGCL ERQUAL5001 during RESTORE SYSTEM	UQ89350
PQ88307	DSNB1PER + 0C1E RC00C002AA during DB2 Restart	UQ91265
PQ88728	DB2 hang, wait when a loop occurs in CSECT DSNB1CMS scanning the defer write queue (DWQ) after buffer manager clean up was run.	UQ89106
PQ89297	Fast log apply performance improvement by assigning each log apply task the same priority of the recover job. Also for Recovery in DB2 V7.	UQ92067
PQ89742	RC00C90094 during REBUILD INDEX(ALL) on SYSCOPY, RNA DSNDXX01	UQ92442
PQ90764	Abend04E RC00C90101 DSNKUNR2:5002 while running RESTORE SYSTEM LOGONLY	UQ91525
PQ90795	DSNU510I Error During SYSUTILX Recover. Recover is unable to find the correct SYSCOPY information for SYSUTILX	UQ91590
PQ91099	ABEND0C4 PIC10 IDCSA11 +4A8 while running DB2 in z/OS 1.5. ABEND0C4 PIC4 in DSNPAMS2 +0AB0 while running point-in-time RESTORE system. ABEND04E RC00C90101 DSNIDBSE ERQUAL5006 while running point-in- time RESTORE system.	UQ91718
PQ91102	Start of DB2 on recovery site DBET contains invalid UTRW states	UQ91315
PQ92187	DSNB232I unexpected data set level ID encountered	UQ92020
PQ93548	Scan the log for prior system checkpoint during a conditional restart of DB2 when the BSDS no longer contains system checkpoint information prior to the log truncation point.	UQ92875
PQ94793	ABEND04E RC00D10340 in DSNJR006 attempting to read/merge the logs in a data sharing environment. Also for DB2 V7.	UQ93407
PQ95164	ABEND04E RC00C90101 in DSNIFLWD ERQUAL5005 during restore system utility.	OPEN

Part 6

Appendixes

In this part of the book, we provide the following appendixes and other supplementary material:

- Appendix A, “REXX procedures” on page 453

  This appendix contains a description of two REXX execs that were developed during the project.

- Appendix C, “Additional material” on page 471

  Here we explain how to obtain the source code for the REXX execs from the ITSO Web site.

- “Abbreviations and acronyms” on page 473

  Abbreviations are spelled out here.

- “Glossary” on page 475

  Here we have tried to put in one place a list of explanations for the exotic (non-DB2) terms often used in the redbook.

- “Related publications” on page 483

  This reference contains pointers to the books we used, and some related Web sites.

A

REXX procedures

In this appendix we provide usage information about two REXX procedures that were developed during the writing of this redbook.

The first one, called CPBULIST, is a tool to assist in copying all backup volumes from a SYSTEM BACKUP utility to tape to be used for remote site disaster recovery.

The second one, called BSDSSTAT, produces a small set of statistics that shows the maximum, minimum, and average time it has taken to fill up the active and archive logs found in the BSDS that you point to.

They are available as described in Appendix C, “Additional material” on page 471.

A.1 CPBULIST: JCL to dump and restore backup disk volumes

The purpose of this REXX procedure is to produce the JCL needed to copy all or selected volumes defined in the DFSMS COPY POOL BACKUP storage group to tape for one given or all DB2 systems. These tapes may then be transported to a recovery site for subsequent disaster recovery.

Since the specific purpose is to copy backup disk volumes to tape for disaster recovery purposes, only COPY POOL BACKUP versions that have both a backup of the databases and the log will be considered by the CPBULIST procedure.

Also, only backup versions that registered under a name that conforms to the DB2 naming standard for the SYSTEM BACKUP utility backup copies are considered. The pools must be called DSN$xxxx$DB and DSN$xxxx$LG, where xxxx is the DB2 subsystem ID.

Various options are available to tailor the DUMP job to suite your requirements. You may for example want to dump only some of the volumes in the COPY POOL BACKUP definition or you may want to dump all volumes for a generation older than the newest one.

The procedure will not only produce a DUMP job, as mentioned, but also a RESTORE job and a report identifying the available copy generations for the selected DB2 subsystem.

Executing the CPBULIST REXX procedure

The CPBULIST REXX procedure may be executed either in TSO foreground or in TSO batch. To execute the procedure in batch you may use the JCL shown in Example A-1.

Example: A-1 Sample JCL to execute CPBULIST in batch

```
//PAOLOR2X JOB (),'DB2 DR RESIDENCY',
//             NOTIFY=PAOLOR2,
//             CLASS=A,MSGCLASS=S,REGION=0M
//CPBULIST EXEC PGM=IKJEFT01,DYNAMNBR=25,TIME=100,
// PARM='%CPBULIST SSID(DB8X) MAXVOLS(3)'
//SYSPROC  DD  DISP=SHR,DSN=&SYSUID..XTIAN.CLIST
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//DUMPJOB  DD  DISP=SHR,DSN=&SYSUID..XTIAN.CNTL(DUMPJOB),FREE=CLOSE
//RESTJOB  DD  DISP=SHR,DSN=&SYSUID..XTIAN.CNTL(RESTJOB),FREE=CLOSE
//JOBCARD  DD  DISP=SHR,DSN=&SYSUID..XTIAN.CNTL(JOBCARD),FREE=CLOSE
```

The JCL will execute the CPBULIST procedure and request a job to dump a maximum of three backup disk volumes per job.

The resulting JCL to dump the backup volumes from disk to tape will be put in member DUMPJOB of the JCL data set defined by the DUMPJOB DD statement.

The JCL to restore the dumped backup volumes from tape to disk will be put in member RESTJOB of the JCL data set defined by the RESTJOB DD statement.

The JOBCARD DD statement defines the JOB card to be used for the generated jobs.

Available parameters

Here is the list of the available parameters:

- SSID(xxxx)

  This parameter defines the specific DB2 system whose backup disk volumes you want to copy to tape. All available DB2 systems will be processed if this parameter is not specified or specified as '*'. The default is SSID(*).

- GEN(xxx)

  This parameter defines the backup generation you want copied to tape. 'xxx' may be specified as either '0', meaning the latest backup, or '-yy' where yy is a number indicating what back-level generation you want copied. So GEN(-3) means copy the 4th newest generation. The default is GEN(0) or the current generation.

- MAXVOLS(xxx)

  This parameter specifies the maximum number of backup volumes to be copied per job. Each backup volume is copied in a separate step and since a job can have a maximum of 255 steps, that is also the limit for this parameter. The default is MAXVOLS(255).

- TAPEUNIT(xxxxxx)

  This parameter specifies the information to go in the UNIT-parameter of the DD statement defining to output tape data sets. Any subparameter of the UNIT parameter may be specified. For example, TAPEUNIT(TAPE,3,DEFER) will result in a UNIT parameter of UNIT=(TAPE,3,DEFER). The default is TAPEUNIT(TAPE).

- DUMPOPT(xxx)

  This parameter specifies the OPTIMIZE parameter of the DUMP statement you want to use. The value must be between 1 and 4, both included. The default is DUMPOPT(4).

- VOLSERS(xxxxxx,yyyyyy,...,zzzzzz)

  With this parameter you specify the volume IDs of the tape volumes to receive the dumped copied of the backup disks. The default is to request scratch tapes.

- ? of HELP

  Specifying this option will cause the CPBULIST procedure to print a small summary of the available options and then terminate.

Specifying JCL

This is a description of the DD statements that you may specify besides the JCL required to execute any REXX procedure in batch. They are all optional.

- DUMPJOB

  This DD statement defines where CPBULIST is to put the generated JCL to dump the backup disk volumes to tape. If this DD statement is not defined, then the resulting JCL will be sent to the SYSTSPRT print data set.

- RESTJOB

  This DD statement defines where CPBULIST is to put the generated JCL to restore the backup disk volumes from tape. If this DD statement is not defined, then the resulting JCL will be sent to the SYSTSPRT print data set.

- JOBCARD

  This DD statement allows you to specify a data set containing a model JOB JCL statement. The jobname specified in the model JOB JCL statement will be treated as a prefix of the jobname that will be generated by the procedure. The suffix will be the number of the job in the generation process. Suppose the jobname in the JOB JCL

statement was specified as PAOLOR and suppose that there are 30 backup disk volumes to copy and that the MAXVOLS parameter was specified as 10.

With these assumptions three jobs would be generated, the first of which would have the jobname PAOLOR1, the second PAOLOR2 etc. Since a jobname cannot exceed 8 characters, it follows that the length of the jobname specified in the model JOB JCL statement plus the number of digits in the job number cannot be longer than 8 bytes. So if the model jobname has a length of 7 characters then no more than 9 jobs can be generated. A 6 character model jobname allows 99 jobs etc.

If the DD statement is not present, then a minimal JOB statement with just a jobname consisting of the user ID of the individual executing the CPBULIST procedure, followed by the job number, will be generated.

- INCLUDE

  This DD statement gives you the possibility of selecting the specific backup disk volumes you want copied to tape.

  If this DD statement is specified, then only volumes defined in the data set will get copied.

- EXCLUDE

  This DD statement allows you to specify backup disk volumes that should not be copied.

  If this DD statement is specified. then all backup disk volumes not explicitly defined in this data set will be copied.

- PRIORITY

  This DD statement allows you to specify the sequence in which you want the backup disk volumes copied to tape and hence which ones you want restored first.

  Let's assume that you want to restore the DB2 Catalog and Directory plus the DB2 logs in that order before everything else. The DB2 Catalog is on volume ID DB2CAT, the DB2 Directory on DB2DIR and the volumes DB2LG1 through DB2LG6 contain the DB2 logs. Let's furthermore assume that a total of 1000 volumes need to be copied and that MAXVOLS was set to 250. You do so by specifying the parameters shown in Example A-2.

Example: A-2 Sample specification of PRIOTITY DD statement input

```
// ..
// ..
//PRIORITY DD *
DB2CAT
DB2DIR
DB2LG1 DB2LG2 DB2LG3 DB2LG4 DB2LG5 DB2LG6
```

With MAXVOLS set to 250 and 1000 backup disk volumes to copy, four jobs will be generated. Job number 1 will as the first two steps copy the DB2CAT and DB2LG3 volumes followed by an additional 248 steps copying the first 248 volumes in sort sequence not including the 8 priority volumes. Job number 2 will copy DB2DIR and DB2LG4 as the first two volumes followed by the next 248 volumes in alphanumeric sort sequence n volume ID. Job number 3 would copy DB2LG1 and DB2LG5 first plus the next 248 volumes and finally job number 4 will copy DB2LG2 and DB2LG6 first followed by the last 248 volumes.

If this DD statement is not specified the backup disk volumes will be copied in the same sequence as the one the backup disk volumes were shown in the report, that is sorted in alphanumeric sequence on volume ID.

Sample output report

Example A-3 shows a report produced by the CPBULIST procedure.

Example: A-3 CPBULIST sample output report

```
CPBULIST - Rundate:6 Aug 2004 Runtime:14:46:34
CPBULIST - Version:2004.08.06 11:45
Job was run with the following parameters:
DB2 ssid            : DB8X
Tape unit           : TAPE
No. of vols per job : 3
DUMP OPTIMIZE       : 4
Generation requested: 0
Tape volsers        : **scratch**

List of priority volumes:
** none **

List of volumes to exclude:
** none **

List of volumes to include:
** none **

ENTRY (A) PAOLOR2.CPBULIST.LISTING DELETED

Available versions:    ssid Type      Ver Date------ Time---- Token(Hex)--------------------------
                       DB8X DATABASE    2 2004/08/06 12:08:27 C4C2F8E7BBA0EC2F6032BE5000008BFFE090
                       DB8X DATABASE    1 2004/08/04 18:51:03 C4C2F8E7BB9EC27212E9DC4C00008BFA0090
                       DB8X LOG         2 2004/08/06 12:08:31 C4C2F8E7BBA0EC2F6032BE5000008BFFE090
                       DB8X LOG         1 2004/08/04 18:51:05 C4C2F8E7BB9EC27212E9DC4C00008BFA0090

Chosen version was:    ssid Type      Ver Date------ Time---- Token(Hex)--------------------------
                       DB8X DATABASE    2 2004/08/06 12:08:27 C4C2F8E7BBA0EC2F6032BE5000008BFFE090
                            LOG         2 2004/08/06 12:08:31 C4C2F8E7BBA0EC2F6032BE5000008BFFE090
Target Source StoGroup ssid Type
DB8X20 DB8X04 DB8XC    DB8X DATABASE Included
DB8X21 DB8X00 DB8XD    DB8X DATABASE Included
DB8X22 DB8X01 DB8XD    DB8X DATABASE Included
DB8X23 DB8X02 DB8XA    DB8X LOG      Included
DB8X24 DB8X03 DB8XL    DB8X LOG      Included
```

The first part of the report shows the parameters used for this run. DB8X is the DB2 system for which we requested the dump operation and we requested the latest possible generation of backup. The unit name of the tape device to receive the dump is TAPE and we have requested 3 disk volumes dumped per job. The dump will be run with OPTIMIZE(4). Scratch tape volumes are requested as the target of the dump.

Neither the INCLUDE, EXCLUDE, nor PRIORITY volumes were defined.

The input parameter feedback section is followed by a section showing the possible generations available for the requested DB2 system. In this case there were two generations with generation 6 being the latest.

A section showing the generation chosen follows. This section documents all the volume defined in the COPY POOL BACKUP definition for the chosen generation. The list is sorted in ascending target volume ID sequence with an indication of the associated source volume ID, storage group, COPY POOL BACKUP type and an indication of whether the volume with be dumped (“included”) or not (“Excluded”).

Sample output DUMP job JCL

Example A-4 shows the JCL for listing a chosen generation.

Example: A-4 CPBULIST sample DUMP job JCL

```
//PAOLOR21 JOB (),'DB2 DR RESIDENCY',
//             NOTIFY=PAOLOR2,
//             CLASS=A,MSGCLASS=S,REGION=0M
//VDB8X12  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X12
//OUTDS    DD  UNIT=TAPE,DISP=(,KEEP),LABEL=1,
//             VOL=(,RETAIN,,99),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X02
   DUMP FULL INDDNAME(INVOL) OUTDDNAME(OUTDS) ALLDATA(*) -
   ALLEXCP CANCELERROR OPTIMIZE(4) ADMINISTRATOR
//VDB8X13  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X13
//OUTDS    DD  UNIT=TAPE,DISP=(,KEEP),LABEL=2,
//             VOL=(,RETAIN,,99,REF=*.VDB8X12.OUTDS),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X03
   DUMP FULL INDDNAME(INVOL) OUTDDNAME(OUTDS) ALLDATA(*) -
   ALLEXCP CANCELERROR OPTIMIZE(4) ADMINISTRATOR
//VDB8X22  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X22
//OUTDS    DD  UNIT=TAPE,DISP=(,KEEP),LABEL=3,
//             VOL=(,RETAIN,,99,REF=*.VDB8X13.OUTDS),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X04
   DUMP FULL INDDNAME(INVOL) OUTDDNAME(OUTDS) ALLDATA(*) -
   ALLEXCP CANCELERROR OPTIMIZE(4) ADMINISTRATOR
//PAOLOR22 JOB (),'DB2 DR RESIDENCY',
//             NOTIFY=PAOLOR2,
//             CLASS=A,MSGCLASS=S,REGION=0M
//VDB8X23  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X23
//OUTDS    DD  UNIT=TAPE,DISP=(,KEEP),LABEL=1,
//             VOL=(,RETAIN,,99),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X00
   DUMP FULL INDDNAME(INVOL) OUTDDNAME(OUTDS) ALLDATA(*) -
   ALLEXCP CANCELERROR OPTIMIZE(4) ADMINISTRATOR
//VDB8X24  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//INVOL    DD  UNIT=SYSDA,DISP=SHR,VOL=SER=DB8X24
//OUTDS    DD  UNIT=TAPE,DISP=(,KEEP),LABEL=2,
//             VOL=(,RETAIN,,99,REF=*.VDB8X23.OUTDS),
//             DSN=PAOLOR2.DB8X.D040728.T175023.DB8X01
   DUMP FULL INDDNAME(INVOL) OUTDDNAME(OUTDS) ALLDATA(*) -
   ALLEXCP CANCELERROR OPTIMIZE(4) ADMINISTRATOR
```

The data set name of the output data sets will consist of the user ID of the individual executing the CPBULIST procedure followed by the DB2 system name. Then follows the date and time of the LOG backup prefixed by a “D” and “T” respectively and ended by the volume ID of the original source disk volume whose backup is to be copied to tape.

The output data sets are not requested to be cataloged since this information will not be available anywhere on the backup tapes.

Sample output RESTORE job JCL

The procedure will also generate a job to restore the backup volumes from tape to disk for subsequent restart at the remote site.

Example A-5 shows the JCL produced to restore the backup volumes.

Example: A-5 CPBULIST sample RESTORE job JCL output

```
//PAOLOR21 JOB
//VDB8X20  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X04
//INDS     DD  UNIT=TAPE,DISP=(,KEEP),LABEL=1,OLD
//             VOL=(,RETAIN,,99),
//             DSN=PAOLOR2.DB8X.D040806.T120831.DB8X04
 RESTORE FULL INDDNAME(INDS) OUTDDNAME(OUTVOL) -
         CANCELERROR PURGE ADMINISTRATOR
//VDB8X21  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X00
//INDS     DD  UNIT=TAPE,DISP=(,KEEP),LABEL=2,OLD
//             VOL=(,RETAIN,,99,REF=*.VDB8X20.OUTDS),
//             DSN=PAOLOR2.DB8X.D040806.T120831.DB8X00
 RESTORE FULL INDDNAME(INDS) OUTDDNAME(OUTVOL) -
         CANCELERROR PURGE ADMINISTRATOR
//VDB8X22  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X01
//INDS     DD  UNIT=TAPE,DISP=(,KEEP),LABEL=3,OLD
//             VOL=(,RETAIN,,99,REF=*.VDB8X21.OUTDS),
//             DSN=PAOLOR2.DB8X.D040806.T120831.DB8X01
 RESTORE FULL INDDNAME(INDS) OUTDDNAME(OUTVOL) -
         CANCELERROR PURGE ADMINISTRATOR
//PAOLOR22 JOB
//VDB8X23  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X02
//INDS     DD  UNIT=TAPE,DISP=(,KEEP),LABEL=1,OLD
//             VOL=(,RETAIN,,99),
//             DSN=PAOLOR2.DB8X.D040806.T120831.DB8X02
 RESTORE FULL INDDNAME(INDS) OUTDDNAME(OUTVOL) -
         CANCELERROR PURGE ADMINISTRATOR
//VDB8X24  EXEC PGM=ADRDSSU
//SYSPRINT DD  SYSOUT=*
//OUTVOL   DD  UNIT=SYSDA,DISP=OLD,VOL=SER=DB8X03
//INDS     DD  UNIT=TAPE,DISP=(,KEEP),LABEL=2,OLD
//             VOL=(,RETAIN,,99,REF=*.VDB8X23.OUTDS),
//             DSN=PAOLOR2.DB8X.D040806.T120831.DB8X03
 RESTORE FULL INDDNAME(INDS) OUTDDNAME(OUTVOL) -
         CANCELERROR PURGE ADMINISTRATOR
```

This restore job assumes that the backups are to be restored on volumes with the same volume IDs as the original production volumes.

A.2 BSDSSTAT: Log fill time statistics from BSDS

The purpose of this REXX procedure is to produce statistics that show the minimum, average and maximum time it has taken to fill the active and archive log data sets registered in the BSDS you defined to the procedure.

You may use these values as one of the determinants for specifying the PCLOSET value as discussed in “Increase accuracy of the DB2 log range registration:” on page 443.

The procedure will execute the DB2 List Log Inventory utility program, DSNJU004, and post-process the report produced by this utility. If the report indicates that the BSDS is for a member of a data sharing group, then the DB2 List Log Inventory utility program, DSNJU004, will be executed once for each BSDS indicated by the first BSDS.

The BSDSSTAT procedure then reads all entries for active and archive COPY 1 data sets recorded in the processed BSDSs and calculates the minimum, average and maximum fill time for all of these logs. It would be a trivial change to let the procedure only look at the entries for the last week or month or some other interval.

Executing the BSDSSTAT REXX procedure

The BSDSSTAT REXX procedure may be executed either in TSO foreground or in TSO batch. To execute the procedure in batch, you may use the JCL in Example A-6.

Example: A-6 Sample JCL to execute BSDSSTAT in batch

```
//PAOLOR2X JOB (),'DB2 DR RESIDENCY',
//            NOTIFY=PAOLOR2,
//            CLASS=A,MSGCLASS=S,REGION=0M
/*JOBPARM T=001,L=0001,ROOM=TS2
/*JOBPARM  S=SC64
//BSDSSTAT EXEC PGM=IKJEFT01,DYNAMNBR=25,TIME=100,
// PARM='%BSDSSTAT DB8XU.BSDS01'
//STEPLIB  DD  DISP=SHR,DSN=DB2G7.SDSNEXIT
//         DD  DISP=SHR,DSN=DB2G7.SDSNLOAD
//SYSPROC  DD  DISP=SHR,DSN=&SYSUID..XTIAN.CLIST
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
```

The JCL will execute the BSDSSTAT procedure and run DSNJU004 against the BSDS data set named DB8XU.BSDS01. The report will be sent to SYSTSPRT.

Available parameters

The BSDSSTAT procedure only provides one parameter. This is a positional parameter that you use to specify the name of the BSDS you want to run against.

This BSDS data set will be dynamically allocated by the procedure and any further BSDSs indicated in the initial BSDS will also be dynamically allocated.

Sample output report

Example A-7 shows a report produced by the BSDSSTAT procedure.

Example: A-7 Sample output report from BSDSSTAT

```
BSDSSTAT - Rundate:6 Aug 2004 Runtime:13:21:58          *****   (c) Copyright IBM Corp. 2004   *****
BSDSSTAT - Version:2004.08.06 10:30

Non-Data Sharing

Processing DB8XU.BSDS01

Log fill time   Hours:mm Log data name
   Minimum          0:00 DB8XU.ARCHLOG1.D2004196.T1641240.A0000030
   Maximum        341:03 DB8XU.ARCHLOG1.D2004189.T2200376.A0000002
   Average         15:04
```

The report begins with a header followed by a line that tells you whether the owning DB2 subsystem of the initial BSDS is a member of a data sharing group. This line might either say “Non-Data Sharing” or “Data Sharing”.

After this, one or more lines will follow indicating the various BSDSs being processed by the BSDSSTAT procedure. The example is for a system that is not a member of a data sharing group so only one BSDS will be processed. The number of BSDSs processed will obviously be equal to the number of members in the data sharing group.

The report ends by reporting the minimum, maximum and average log fill time expressed in hours and minutes. For the minimum and maximum values, you can also see the log data set for which the minimum and maximum apply. So the maximum time it took to fill and DB2 active or archive log was 341 hours and 3 minutes, and that happened to be the archive log data set named DB8XU.ARCHLOG1.D2004189.T2200376.A0000002.

B

PITR definitions

In this appendix we provide the following information:

- Setting up copy pools for PITR:

  Usage information about setting up the definitions of copy pools for point-in-time recovery using Fast Replication. We show the sequence of ISMF panels your storage administrator needs to go through while defining data base and log copy pools mentioned in 6.2, “Preparing for DFSMShsm Fast Replication” on page 82.

- Sample scenario: Restoring to an arbitrary PITR:

  Sample scenario on Restoring to an arbitrary PITR with a methodology which could be used if one of the requirements for BACKUP SYSTEM and RESTORE SYSTEM is not in place.

B.1 Setting up copy pools for PITR

The following steps should be used to define the necessary copy pool, source storage group, and backup storage group structures for fast replication support:

- Start to define your copy pool definitions by selecting option P, Specify Storage Groups for Copies, from the ISMF Primary Option Menu as shown in Figure B-1.

```
ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R5
Enter Selection or Command ===> P

Select one of the following options and press Enter:
0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Criteria
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availability
6  Storage Group             - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
C  Data Collection           - Process Data Collection Function
L  List                      - Perform Functions Against Saved ISMF Lists
P  Copy Pool                 - Specify Pool Storage Groups for Copies
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                      - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure B-1 Select option P, Copy Pool

- Enter the database copy pool name using the required form of DSN$locn-name$DB and select option 3, Define a *database* copy pool, as shown in Figure B-2.

```
COPY POOL APPLICATION SELECTION
Command ===>

To perform Copy Pool Operations, Specify:
  CDS Name . . . . 'SMSCTL.SCDS'
                                (1 to 44 character data set name or 'Active' )
  Copy Pool Name   DSN$P870$DB                   (For Copy Pool List, fully
                                          or partially specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Copy Pools
     2. Display - Display a Copy Pool
     3. Define  - Define a Copy Pool
     4. Alter   - Alter a Copy Pool

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

Figure B-2 Define the database copy pool

- Enter the storage group name and number of backup versions for DFSMShsm to manage. Notice that DFSMShsm is asked to manage up to 15 copy pool backup versions on disk, as shown in Figure B-3.

```
COPY POOL DEFINE                  Page 1 of 3
Command ===>

SCDS Name  . . : SMSCTL.SCDS
Copy Pool Name : DSN$P870$DB

To DEFINE Copy Pool, Specify:
  Description ==> COPY POOL FOR P870
              ==>
  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 15       (1 to 85 or blank)
  Storage Group Names:  (specify 1 to 256 names)
   ==> P87VCAT
   ==>
   ==>
```

Figure B-3 Define the source storage group to the database copy pool

- If you wish to take full system backups, enter the *log* copy pool name using the required form of DSN$*locn-name*$LG and select option 3, as shown in Figure B-4.

```
COPY POOL APPLICATION SELECTION
 Command ===>

 To perform Copy Pool Operations, Specify:
   CDS Name . . . . 'SMSCTL.SCDS'
                                  (1 to 44 character data set name or 'Active' )
   Copy Pool Name   DSN$P870$LG                    (For Copy Pool List, fully
                                            or partially specified or * for all)

 Select one of the following options  :
   3  1. List    - Generate a list of Copy Pools
      2. Display - Display a Copy Pool
      3. Define  - Define a Copy Pool
      4. Alter   - Alter a Copy Pool

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

Figure B-4 Define the log copy pool

- Enter the storage group name and number of backup versions for DFSMShsm to manage, as shown in Figure B-5.

```
COPY POOL DEFINE                  Page 1 of 3
Command ===>

SCDS Name  . . : SMSCTL.SCDS
Copy Pool Name : DSN$P870$LG

To DEFINE Copy Pool, Specify:
  Description ==> COPY POOL FOR P870 BSDS + LOG DATASETS
              ==>
  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . .15       (1 to 85 or blank)
  Storage Group Names:  (specify 1 to 256 names)
   ==> DSNP870
   ==>
   ==>
```

Figure B-5 Define the source storage group to the log copy pool

- Connect the source storage groups with their associated backup storage groups using option 6, Specify Volume Names and Free Space Thresholds, as shown in Figure B-6.

```
ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R5
Enter Selection or Command ===> 6

Select one of the following options and press Enter:
0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Criteria
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availability
6  Storage Group             - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
C  Data Collection           - Process Data Collection Function
L  List                      - Perform Functions Against Saved ISMF Lists
P  Copy Pool                 - Specify Pool Storage Groups for Copies
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                      - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure B-6 Select the Storage Group

- Enter the source storage group name and select option 3, Alter a Storage Group, to associate the storage groups, as shown in Figure B-7.

```
STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . 'SMSCTL.SCDS'
                          (1 to 44 character data set name or 'Active' )
  Storage Group Name    P87VCAT          (For Storage Group List, fully or
                                         partially specified or * for all)
  Storage Group Type                     (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                         OBJECT, OBJECT BACKUP, or TAPE)

Select one of the following options  :
  3  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

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

Figure B-7 Alter by Storage Group Name

- Enter a description, the backup copy pool name in the *Copy Pool Backup SG Name* field, and ‘Y’ in the *SMS Alter Storage Group Status* field, as shown in Figure B-8.

```
POOL STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SMSCTL.SCDS
Storage Group Name  : P87VCAT
To ALTER Storage Group, Specify:
 Description ==> FOR P870 CONNECT P87VCATP TO P87VCAT
             ==>
 Auto Migrate . . N  (Y, N, I or P)   Migrate Sys/Sys Group Name . .
 Auto Backup  . . N  (Y or N)         Backup Sys/Sys Group Name  . .
 Auto Dump  . . . N  (Y or N)         Dump Sys/Sys Group Name  . . .
 Overflow . . . . N  (Y or N)         Extend SG Name . . . . . . . .
                                      Copy Pool Backup SG Name . . . P87VCATP
 Dump Class . . .                     (1 to 8 characters)
 Dump Class . . .                     Dump Class . . .
 Dump Class . . .                     Dump Class . . .
 Allocation/migration Threshold: High . . 85  (1-99)      Low  . .     (0-99)
 Guaranteed Backup Frequency  . . . . . .          (1 to 9999 or NOLIMIT)

 ALTER     SMS Storage Group Status . . . Y   (Y or N)
```

Figure B-8 Associating source and target storage groups

Associate source storage group P87VCAT with copy pool backup P87VCATP and set SMS Storage Group Status to ‘Y’. Do the same with the log storage groups.

Be sure to validate the backup environment each time it is changed — for example, when volumes in a source or backup storage group change, when the number of versions to maintain changes, or when the storage groups defined to a copy pool have changed. Be aware of how and when your system configuration has changed before you use a copy pool (with RESTORE SYSTEM or outside of DB2) to restore a system.

DB2 Administrator uses the DB2 utility to create fast replication backups of the data base.

B.2 Sample scenario: Restoring to an arbitrary PITR

This methodology could be used if one of the requirements for BACKUP SYSTEM and RESTORE SYSTEM is not in place.

- Start DB2. If data sharing, start all dormant members.
- Execute DDL to create a database, table space, and two tables each with one index.
- Take a system backup:
  - Execute SET LOG SUSPEND to stop update activity.
  - Take backups of “data” volumes using existing volume copy or Split Mirror solutions. If on z/OS V1R5 you can use copy pools to simplify the process.
  - Execute SET LOG RESUME to resume update activity.
- Execute DML to insert rows into one table, then update some of the rows.
- Use the LOAD utility with the LOG NO attribute to load the second table.

- Create another table space, table and index in an existing database.
- Use SET LOG SUSPEND/SET LOG RESUME to establish log truncation point *logpoint1*, the point to which you want to recover. If non-data sharing use the RBA, if data sharing, use the lowest LRSN among active members.
- Execute DML to insert rows into one of the tables, and to update and/or delete some rows.
- Stop DB2. If data sharing, stop all active members.
- Use DSNJU003 to create a SYSPITR CRCR record (CRESTART CREATE SYSPITR=*logpoint1*). This is the log truncation point established above. If data sharing, create a SYSPITR record for each member active member.
- Restore only the “data” volumes using an existing volume copy process, or if on z/OS V1R5 you can use copy pools to simplify the process.
- If data sharing, delete all coupling facility structures.
- Restart DB2. DB2 will start in system recovery-pending mode. If data sharing, restart all members.
- Execute the RESTORE SYSTEM utility with the LOGONLY keyword. If data sharing, the utility only needs to be executed on one member. If the utility terminates and must be restarted it can only be restarted on the member on which it was originally executed.
- After the utility ends successfully, stop DB2. If data sharing, stop all active members. This will reset system recovery-pending status.
- Restart DB2. If data sharing, restart all members.
- Execute the display command to check for active utilities or restricted objects. Terminate any active utilities. Recover any objects in RECP or RBDP status.
  - -DIS UTIL(*) and terminate any active utilities.
  - -DIS DB(DSNDB01) SP(*)
  - -DIS DB(DSNDB06) SP(*) LIMIT(*)
  - -DIS DB(*) SP(*) LIMIT(*) RESTRICT
- Validate that recovery was successful.

C

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/SG24????`

Alternatively, you can go to the IBM Redbooks Web site at:

`ibm.com/redbooks`

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG24-6370-00.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	*Description*
clst6370.zip	Zipped REXX execs described in Appendix A, “REXX procedures” on page 453.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 2 MB minimum
Operating System: Windows
Processor: Intel 386 or higher
Memory: 16 MB

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Abbreviations and acronyms

ACS	automatic class selection
AIX	Advanced Interactive eXecutive from IBM
APAR	authorized program analysis report
ARM	automatic restart manager
ASCII	American National Standard Code for Information Interchange
BCRS	business continuity recovery services
BLOB	binary large objects
BPA	buffer pool analysis
BCDS	DFSMShsm backup control data set
BSDS	boot strap data set
CCA	channel connection address
CCA	client configuration assistant
CCP	collect CPU parallel
CCSID	coded character set identifier
CD	compact disk
CEC	central electronics complex
CF	coupling facility
CFCC	coupling facility control code
CFRM	coupling facility resource management
CLI	call level interface
CLP	command line processor
CPU	central processing unit
CRCR	conditional restart control record
CRD	collect report data
CSA	common storage area
CTT	created temporary table
DASD	direct access storage device
DB2 PM	DB2 performance monitor
DBAT	database access thread
DBD	database descriptor
DBID	database identifier
DBRM	database request module
DCL	data control language
DDCS	distributed database connection services
DDF	distributed data facility
DDL	data definition language
DLL	dynamic load library manipulation language
DML	data manipulation language
DNS	domain name server
DRDA®	distributed relational database architecture
DSC	dynamic statement cache, local or global
DTT	declared temporary tables
DWDM	dense wavelength division multiplexer
DWT	deferred write threshold
EA	extended addressability
EBCDIC	extended binary coded decimal interchange code
ECS	enhanced catalog sharing
ECSA	extended common storage area
EDM	environment descriptor management
ELB	extended long busy
ERP	enterprise resource planning
ERP	error recovery procedure
ESA	Enterprise Systems Architecture
ESP	Enterprise Solution Package
ESS	Enterprise Storage Server
ETR	external throughput rate, an elapsed time measure, focuses on system capacity
FIFO	first in first out
FTD	functional track directory
FLA	fast log apply
FTP	File Transfer Program
GB	gigabyte (1,073,741,824 bytes)
GBP	group buffer pool
GRS	global resource serialization
GUI	graphical user interface
HPJ	high performance Java
I/O	input/output
IBM	International Business Machines Corporation
ICF	integrated catalog facility
ICF	integrated coupling facility

ICMF	internal coupling migration facility
IFCID	instrumentation facility component identifier
IFI	instrumentation facility interface
IFI	Instrumentation Facility Interface
IGS	IBM Global Services
IPLA	IBM Program Licence Agreement
IRLM	internal resource lock manager
ISPF	interactive system productivity facility
IRWW	IBM Relational Warehouse Workload
ISV	independent software vendor
IT	Information Technology
ITR	internal throughput rate, a processor time measure, focuses on processor capacity
ITSO	International Technical Support Organization
IVP	installation verification process
JDBC	Java Database Connectivity
JFS	journaled file systems
JNDI	Java Naming and Directory Interface
JVM	Java Virtual Machine
KB	kilobyte (1,024 bytes)
LOB	large object
LPAR	logical partition
LPL	logical page list
LRECL	logical record length
LRSN	log record sequence number
LRU	least recently used
LUW	logical unit of work
LVM	logical volume manager
MB	megabyte (1,048,576 bytes)
NPI	non-partitioning index
NVS	non volatile storage
ODB	object descriptor in DBD
ODBC	Open Data Base Connectivity
OP	Online performance
OS/390	Operating System/390®
PAV	parallel access volume
PDS	partitioned data set
PIB	parallel index build
PPRC	Peer-to-Peer Remote Copy
PSID	pageset identifier
PSP	preventive service planning
PTF	program temporary fix
PUNC	possibly uncommitted
PWH	Performance Warehouse
QA	Quality Assurance
QMF™	Query Management Facility
RACF	Resource Access Control Facility
RBA	relative byte address
RBLP	recovery base log point
RECFM	record format
RID	record identifier
RR	repeatable read
RRS	resource recovery services
RRSAF	resource recovery services attach facility
RPO	recovery point objective
RS	read stability
RTO	recovery time objective
SCUBA	self contained underwater breathing apparatus
SDM	System Data Mover
SMIT	System Management Interface Tool
SPL	selective partition locking
SU	Service Unit
UOW	unit of work
XRC	eXtended Remote Copy
WTO	write to operator

Glossary

A.

address space A range of virtual storage pages identified by a number (ASID) and a collection of segment and page tables which map the virtual pages to real pages of the computer's memory.

address space connection The result of connecting an allied address space to DB2. Each address space containing a task connected to DB2 has exactly one address space connection, even though more than one task control block (TCB) can be present. See allied address space and task control block.

allied address space An area of storage external to DB2 that is connected to DB2 and is therefore capable of requesting DB2 services.

alternate site An alternate operating location to be used by business functions when the primary facilities are inaccessible. 1) Another location, computer center or work area designated for recovery. 2) Location, other than the main facility, that can be used to conduct business functions. 3) A location, other than the normal facility, used to process data and/or conduct critical business functions in the event of a disaster.

application plan The control structure produced during the bind process and used by DB2 to process SQL statements encountered during statement execution.

application program interface (API) A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or licensed program.

ASCII (1) American Standard Code for Information Interchange.A standard assignment of 7-bit numeric codes to characters. See also *Unicode*. (2) An encoding scheme used to represent strings in many environments, typically on PCs and workstations. Contrast with EBCDIC.

attachment facility An interface between DB2 and TSO, IMS, CICS, or batch address spaces. An attachment facility allows application programs to access DB2.

authorization ID A string that can be verified for connection to DB2 and to which a set of privileges are allowed. It can represent an individual, an organizational group, or a function, but DB2 does not determine this representation.

automatic bind (More correctly, automatic rebind). A process by which SQL statements are bound automatically (without a user issuing a BIND command) when an application process begins execution and the bound application plan or package it requires is not valid.

automatic class selection (ACS) routine A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

B.

backup control data set (BCDS) A VSAM, key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

base table (1) A table created by the SQL CREATE TABLE statement that is used to hold persistent data. Contrast with result table and temporary table. (2) A table containing a LOB column definition. The actual LOB column data is not stored along with the base table. The base table contains a row identifier for each row and an indicator column for each of its LOB columns. Contrast with auxiliary table.

batch **(1)** An accumulation of data to be processed. **(2)** A group of records or data processing jobs brought together for processing. **(3)** Pertaining to activity involving little or no user.

binary large object (BLOB) A sequence of bytes, where the size of the sequence ranges from 0 bytes to 2 GB - 1. Such a string does not have an associated CCSID. The size of binary large object values can be anywhere up to 2 GB - 1.

bind The process by which the output from the DB2 precompiler is converted to a usable control structure called a package or an application plan. During the process, access paths to the data are selected and some authorization checking is performed.

built-in function A function that is supplied by DB2. Contrast with user-defined function.

business continuity planning (BCP) Process of developing advance arrangements and procedures that enable an organization to respond to an event in such a manner that critical business functions continue with planned levels of interruption or essential change. Other terms: Contingency Planning, Disaster Recovery Planning.

C.

call attachment facility (CAF) A DB2 attachment facility for application programs running in TSO or MVS batch. The CAF is an alternative to the DSN command processor and allows greater control over the execution environment.

call level interface (CLI) A callable application program interface (API) for database access, which is an alternative to using embedded SQL. In contrast to embedded SQL, DB2 CLI does not require the user to precompile or bind applications, but instead provides a standard set of functions to process SQL statements and related services at run time.

cast function A function used to convert instances of a (source) data type into instances of a different (target) data type. In general, a cast function has the name of the target data type. It has one single argument whose type is the source data type; its return type is the target data type.

casting Explicitly converting an object or primitive's data type.

catalog In DB2, a collection of tables that contains descriptions of objects such as tables, views, and indexes.

character large object (CLOB) A sequence of bytes representing single-byte characters or a mixture of single and double-byte characters where the size can be up to 2 GB - 1. Although the size of character large object values can be anywhere up to 2 GB - 1, in general, they are used whenever a character string might exceed the limits of the VARCHAR type.

cold site An alternate facility that already has in place the environmental infrastructure required to recover critical business functions or information systems, but does not have any pre-installed computer hardware, telecommunications equipment, communication lines, etc. These must be provisioned at time of disaster.

column function An SQL operation that derives its result from a collection of values across one or more rows. Contrast with scalar function.

commit The operation that ends a unit of work by releasing locks so that the database changes made by that unit of work can be perceived by other processes.

concurrent copy A method for creating a point-in-time copy in zSeries and S/390 environments, with the source data fully available for access and update after initiation of the copy operation.

contingency plan A plan used by an organization or business unit to respond to a specific systems failure or disruption of operations. A contingency plan may use any number of resources including workaround procedures, an alternate work area, a reciprocal agreement, or replacement resources.

cursor A named control structure used by an application program to point to a row of interest within some set of rows, and to retrieve rows from the set, possibly making updates or deletions.

D.

data backups The back up of system, application, program and/or production files to media that can be stored both on and/or offsite. Data backups can be used to restore corrupted or lost data or to recover entire systems and databases in the event of a disaster. Data backups should be considered confidential and should be kept secure from physical damage and theft.

data class A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

Data Facility Storage Management Subsystem (DFSMS) An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, Management Class, storage group, and automatic class selection routine definitions.

data recovery The restoration of computer files from backup media to restore programs and production data to the state that existed at the time of the last safe backup.

data replication The partial or full duplication of data from a source database to one or more destination databases. Replication may use any of a number of methodologies including mirroring or shadowing, and may be performed synchronous, asynchronous, or point-in-time depending on the technologies used, recovery point requirements, distance and connectivity to the source database, etc. Replication, if performed remotely, can function as a backup for disasters and other major outages. (

database management system (DBMS) A software system that controls the creation, organization, and modification of a database and access to the data stored within it.

DB2 thread The DB2 structure that describes an application's connection, traces its progress, processes resource functions, and delimits its accessibility to DB2 resources. and services.

DBCLOB A sequence of bytes representing double-byte characters where the size can be up to 2 gigabytes. Although the size of double-byte character large object values can be anywhere up to 2 gigabytes, in general, they are used whenever a double-byte character string might exceed the limits of the VARGRAPHIC type.

DFSMSdfp A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data management.

DFSMSdss A DFSMS functional component or base element of z/OS, used to copy, move dump, and restore data sets or volumes.

DFSMShsm A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

disaster A sudden, unplanned calamitous event causing great damage or loss. 1) Any event that creates an inability on an organizations part to provide critical business functions for some predetermined period of time. 2) In the business environment, any event that creates an inability on an organization's part to provide the critical business functions for some predetermined period of time. 3) The period when company management decides to divert from normal production responses and exercises its disaster recovery plan. Typically signifies the beginning of a move from a primary to an alternate location.

disaster recovery Activities and programs designed to return the entity to an acceptable condition. 1) The ability to respond to an interruption in services by implementing a disaster recovery plan to restore an organization's critical business functions.

disk mirroring Disk mirroring is the duplication of data on separate disks in real time to ensure its continuous availability, currency and accuracy. Disk mirroring can function as a disaster recovery solution by performing the mirroring remotely. True mirroring will enable a zero recovery point objective. Depending on the technologies used, mirroring can be performed synchronously, asynchronously, semi-synchronously, or point-in-time.

distinct type A user-defined data type that is internally represented as an existing type (its source type), but is considered to be a separate and incompatible type for semantic purposes.

distributed processing Processing that takes place across two or more linked systems.

distributed relational database architecture (DRDA) A connection protocol for distributed relational database processing that is used by IBM's relational database products. DRDA includes protocols for communication between an application and a remote relational database management system, and for communication between relational database management systems.

dynamic bind A process by which SQL statements are bound as they are entered.

dynamic SQL SQL statements that are prepared and executed within an application program while the program is executing. In dynamic SQL, the SQL source is contained in host language variables rather than being coded into the application program. The SQL statement can change several times during the application program's execution.

E.

EBCDIC Extended binary coded decimal interchange code. An encoding scheme used to represent character data in the MVS, VM, VSE, and OS/400Ñ environments. Contrast with ASCII.

electronic vaulting Electronically forwarding backup data to an offsite server or storage facility. Vaulting eliminates the need for tape shipment and therefore significantly shortens the time required to move the data offsite.

enclave In Language Environment® for MVS & VM, an independent collection of routines, one of which is designated as the main routine. An enclave is similar to a program or run unit.

extended long busy (ELB) Technology which assists with the consistency of application data capable of dependent writes. If any volume within a consistency group is unable to complete a write to its counterpart in the PPRC relationship, an ELB will be issued, preventing further writes to any of the volumes within the consistency group. This ELB period is the perfect time to issue a freeze to all volumes involved to maintain consistency.

Extended Remote Copy (XRC) A combined hardware and software business continuance solution for the zSeries and S/390® environments providing asynchronous mirroring between two ESSs at global distances. XRC is an optional feature on the ESS.

external function A function for which the body is written in a programming language that takes scalar argument values and produces a scalar result for each invocation. Contrast with sourced function and built-in function.

F.

FlashCopy An optional feature on the ESS which creates a physical point-in-time copy of the data, with minimal interruption to applications, and makes it possible to access both the source and target copies immediately.

foreign key A key that is specified in the definition of a referential constraint. Because of the foreign key, the table is a dependent table. The key must have the same number of columns, with the same descriptions, as the primary key of the parent table.

forward recovery The process of recovering a database to the point of failure by applying active journal or log data to the current backup files of the database.

function A specific purpose of an entity or its characteristic action such as a column function or scalar function. (See column function and scalar function.). Furthermore, functions can be user-defined, built-in, or generated by DB2. (See built-in function, cast function, user-defined function, external function, sourced function.)

G.

(IBM TotalStorage) Global Copy Same as PPRC-XD.

(IBM TotalStorage) Global Mirror Asynchronous PPRC

(IBM TotalStorage z/OS) Global Mirror Same as XRC.

Graphical User Interface (GUI) A type of computer interface consisting of a visual metaphor of a real-world scene, often of a desktop. Within that scene are icons, representing actual objects, that the user can access and manipulate with a pointing device.

H.

high availability Systems or applications requiring a very high level of reliability and availability. High availability systems typically operate 24x7 and usually require built in redundancy built-in redundancy to minimize the risk of downtime due to hardware and/or telecommunication failures.

hot site An alternate facility that already has in place the computer, telecommunications, and environmental infrastructure required to recover critical business functions or information systems.

Hypertext Markup Language (HTML) A file format, based on SGML, for hypertext documents on the Internet. Allows for the embedding of images, sounds, video streams, form fields and simple text formatting. References to other objects are embedded using URLs, enabling readers to jump directly to the referenced document.

I.

incremental bind A process by which SQL statements are bound during the execution of an application process, because they could not be bound during the bind process, and VALIDATE(RUN) was specified.

J.

JDBC (Java Database Connectivity) In the JDK, the specification that defines an API that enables programs to access databases that comply with this standard.

L.

large object (LOB) A sequence of bytes representing bit data, single-byte characters, double-byte characters, or a mixture of single and double-byte characters. A LOB can be up to 2GB -1 byte in length. See also BLOB, CLOB, and DBCLOB.

load module A program unit that is suitable for loading into main storage for execution. The output of a linkage editor.

M.

(IBM TotalStorage) Metro Mirror Synchronous PPRC

(IBM TotalStorage) Metro/Global Copy
Asynchronous Cascading PPRC

(IBM TotalStorage z/OS) Metro/Global Mirror
Three-site solution using Synchronous PPRC and XRC.

multithreading Multiple TCBs executing one copy of DB2 ODBC code concurrently (sharing a processor) or in parallel (on separate central processors).

N.

null A special value that indicates the absence of information.

O.

Open Database Connectivity (ODBC) A Microsoft database application programming interface (API) for C that allows access to database management systems by using callable SQL. ODBC does not require the use of an SQL preprocessor. In addition, ODBC provides an architecture that lets users add modules called database drivers that link the application to their choice of database management systems at run time. This means that applications no longer need to be directly linked to the modules of all the database management systems that are supported.

outsourcing The transfer of data processing functions to an independent third party

P.

parallel sysplex A sysplex with one or more coupling facilities, and defined by the COUPLExx members of SYS1.PARMLIB as being a parallel sysplex.

Peer-to-Peer Remote Copy (PPRC) A hardware-based disaster recovery solution designed to provide real-time mirroring of logical volumes within an ESS or between two distant ESSs. PPRC has two basic modes of operation: synchronous and non-synchronous.

Peer-to-Peer Remote Copy Extended Distance (PPRC-XD) This extension to PPRC offers a non-synchronous long-distance copy option whereby write operations to the primary ESS are considered complete before they are transmitted to the secondary ESS.

plan name The name of an application plan.

primary application site/system The site and systems where the production data and applications normally run are referred to as the primary site or primary systems. Application site and application systems have the same meaning.

primary ESS The ESS where the production data resides, which contains the primary volumes of the XRC volume pairs. It is also sometimes referred to as the primary subsystem, or primary storage control, or by referring to its primary logical control units (LSSs).

R.

reciprocal agreement Agreement between two organizations (or two internal business groups) with basically the same equipment/same environment that allows each one to recover at each other's site.

recovery Process of planning for and/or implementing expanded operations to address less time-sensitive business operations immediately following an interruption or disaster. 1) The start of the actual process or function that uses the restored technology and location.

recovery point objective (RPO) The point-in-time to which systems and data must be recovered after an outage. (e.g. end of previous day's processing). RPOs are often used as the basis for the development of backup strategies, and as a determinant of the amount of data that may need to be recreated after the systems or functions have been recovered.

recovery strategy An approach by an organization that will ensure its recovery and continuity in the face of a disaster or other major outage. Plans and methodologies are determined by the organizations strategy. There may be more than one methodology or solution for an organizations strategy. Examples of methodologies and solutions include contracting for Hotsite or Coldsite, building an internal Hotsite or Coldsite, identifying an Alternate Work Area, a Consortium or Reciprocal Agreement, contracting for Mobile Recovery or Crate and Ship, and many others.

recovery time objective (RTO) The period of time within which systems, applications, or functions must be recovered after an outage (e.g. one business day). RTOs are often used as the basis for the development of recovery strategies, and as a determinant as to whether or not to implement the recovery strategies during a disaster situation. Other term: Maximum allowable downtime.

reentrant Executable code that can reside in storage as one shared copy for all threads. Reentrant code is not self-modifying and provides separate storage areas for each thread. Reentrancy is a compiler and operating system concept, and reentrancy alone is not enough to guarantee logically consistent results when multithreading. See threadsafe.

relational database management system (RDBMS) A relational database manager that operates consistently across supported IBM systems.

remote Refers to any object maintained by a remote DB2 subsystem; that is, by a DB2 subsystem other than the local one. A remote view, for instance, is a view maintained by a remote DB2 subsystem. Contrast with local.

S.

scalar function An SQL operation that produces a single value from another value and is expressed as a function name followed by a list of arguments enclosed in parentheses. See also column function.

secondary ESS. The ESS where copies of the primary volumes reside is referred to as the secondary ESS, or secondary subsystem, or secondary storage control, or by referring to its secondary logical control units (LSSs).

secondary site/system We normally refer to the site and systems where the recovery or test data and applications run as the secondary site and secondary system. Recovery site and recovery systems have the same meaning. However, we prefer the more generic terms secondary site and systems, as XRC can be used for data and workload migration, as well as in a disaster recovery solution.

SQL Structured Query Language. A language used by database engines and servers for data acquisition and definition.

SQL Communication Area (SQLCA) A structure used to provide an application program with information about the execution of its SQL statements.

SQL Descriptor Area (SQLDA) A structure that describes input variables, output variables, or the columns of a result table.

static bind A process by which SQL statements are bound after they have been precompiled. All static SQL statements are prepared for execution at the same time. Contrast with dynamic bind.

static SQL SQL statements, embedded within a program, that are prepared during the program preparation process (before the program is executed). After being prepared, the SQL statement does not change (although values of host variables specified by the statement might change).

Storage Management Subsystem (SMS) A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, Management Class, storage group, and ACS routine definitions.

stored procedure A user-written application program, that can be invoked through the use of the SQL CALL statement.

striping A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

T.

table A named data object consisting of a specific number of columns and some number of unordered rows. Synonymous with base table or temporary table.

task control block (TCB) A control block used to communicate information about tasks within an address space that are connected to DB2. An address space can support many task connections (as many as one per task), but only one address space connection. See address space connection.

temporary table A table created by the SQL CREATE GLOBAL TEMPORARY TABLE statement that is used to hold temporary data. Contrast with result table.

thread A separate flow of control within a program.

timestamp A seven-part value that consists of a date and time expressed in years, months, days, hours, minutes, seconds, and microseconds.

trace A DB2 facility that provides the ability to monitor and collect DB2 monitoring, auditing, performance, accounting, statistics, and serviceability (global) data.

U.

Unicode A 16-bit international character set defined by ISO 10646. See also ASCII.

user-defined function (UDF) A function defined to DB2 using the CREATE FUNCTION statement that can be referenced thereafter in SQL statements. A user-defined function can be either an external function or a sourced function. Contrast with built-in function.

V.

virtual machine A software or hardware implementation of a central processing unit (CPU) that manages the resources of a machine and can run compiled code. See *Java Virtual Machine*.

W.

warm site An alternate processing site which is equipped with some hardware, and communications interfaces, electrical and environmental conditioning which is only capable of providing backup after additional provisioning, software or customization is performed.

WebSphere WebSphere is the cornerstone of IBM's overall Web strategy, offering customers a comprehensive solution to build, deploy and manage e-business Web sites. The product line provides companies with an open, standards-based, Web server deployment platform and Web site development and management tools to help accelerate the process of moving to e-business.

World Wide Web A network of servers that contain programs and files. Many of the files contain hypertext links to other documents available through the network.

X.

XRC volume pairs XRC will copy primary volumes from the primary site to the secondary volumes at the secondary site. The primary volume and its corresponding secondary volume makes an XRC volume pair.

X/Open An independent, worldwide open systems organization that is supported by most of the world's largest information systems suppliers, user organizations, and software companies. X/Open's goal is to increase the portability of applications by combining existing and emerging standards.

Z.

(IBM TotalStorage) z/OS Global Mirror Same as XRC.

(IBM TotalStorage) z/OS Metro/Global Mirror Three-site solution using Synchronous PPRC and XRC.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 484. Note that some of the documents referenced here may be available in softcopy only.

- *DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There,* SG24-7088-00
- *DFSMShsm Fast Replication Technical Guide,* SG24-7069-00
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries,* SG24-5680-04
- *IBM TotalStorage Solutions for Disaster Recovery,* SG24-6547-01
- *DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ... and More,* SG24-6079
- *DB2 UDB for z/OS Version 8 Technical Preview*, SG24-6871

Other publications

These publications are also relevant as further information sources:

- *DB2 UDB for z/OS Version 8 Administration Guide,* SC18-7413
- *DB2 UDB for z/OS Version 8 Application Programming and SQL Guide,* SC18-7415
- *DB2 UDB for z/OS Version 8 Application Programming Guide and Reference for Java,* SC18-7414
- *DB2 UDB for z/OS Version 8 Command Reference,* SC18-7416
- *DB2 UDB for z/OS Version 8 Data Sharing: Planning and Administration,* SC18-7417
- *DB2 UDB for z/OS Version 8 Installation Guide,* GC18-7418
- *DB2 UDB for z/OS Version 8 Messages and Codes,* GC18-7422
- *DB2 UDB for z/OS Version 8 Program Directory,* GI10-8566
- *DB2 UDB for z/OS Version 8 RACF Access Control Module Guide,* SC18-7433
- *DB2 UDB for z/OS Version 8 Release Planning Guide,* SC18-7425
- *DB2 UDB for z/OS Version 8 SQL Reference,* SC18-7426
- *DB2 UDB for z/OS Version 8 Utility Guide and Reference,* SC18-7427
- *z/OS DFSMSdss Storage Administration Reference,* SC35-0424
- *z/OS V1R5.0 DFSMShsm Storage Administration Reference*, SC35-0422
- *z/OS DFSMS Advanced Copy Services,* SC35-0428
- *Device Support Facility Users Guide and Reference,* GC35-0033
- *z/OS DFSMSdfp Advanced Services,* SC26-7400

- *IBM TotalStorage Enterprise Storage Server Web Interface User's Guide,* SC26-7448
- *z/OS V1R3.0 MVS System Management Facilities,* SA22-7630
- *z/OS DFSMSdss Storage Administration Guide*, SC35-0423
- *IBM TotalStorage Enterprise Storage Server User's Guide*, SC26-7445
- *DB2 Log Analysis Tool for z/OS Version 2.1 User's Guide*, SC18-9132
- *DB2 Archive Log Accelerator for z/OS Version 2 User's Guide*, SC18-7405
- *DB2 Change Accumulation Tool for z/OS Version 1.3 User's Guide*, SC27-1191
- *IBM TotalStorage Enterprise Storage Server Model 800 DFSMSdss FlashCopy Version 2 Performance Evaluation and Analysis* - 11/11/2003 - Xavier Velasquez, white paper

Online resources

These Web sites and URLs are also relevant as further information sources:

- For BS7799

  http://www.bsi.org.uk or http://www.bsi-global.com/index.xalter

- For ISO/IEC 17799

  http://www.iso.org

- For US Securities and Exchange Commission

  http://sec.gov/news/studies/34-47638.htm

- For the latest maintenance information on z/OS

  http://www.ibm.com/servers/storage/support/solutions/bc.html

- For information on GDPS

  http://www.ibm.com/servers/eserver/zseries/library/whitepapers/gf225114.html
  http://www.ibm.com/services/us/index.wss/so/its/a1000189

  or e-mail

  gdps@us.ibm.com

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A
ABAP 372–373, 378
ALTER BUFFERPOOL 445
ALTER INDEX 441
ANTA8124I 358
ANTASnnn address space 117, 119, 214, 218–219
ANTE8106I 343
ANTR8102I 344, 349, 360
ANTRQST macro 70, 102, 220
ANTV8109I 357, 359
ANTX8120I 341, 357–359
APSM 194
ARC1000I 402
ARC2FRST 47, 255, 261, 420
Architecture 23, 121, 169–170, 211, 234
Assign 110, 168, 215
Asynchronous PPRC 18, 20, 97, 106–111, 165–176, 178–181, 183–190, 192–195, 197–198
  basics 109
Automatic LPL recovery 352
Automatic Restart Manager 413
automation xxix, 13, 25–26, 36, 102, 106–107, 142, 146, 166, 189, 193–194, 197, 220, 225, 228, 234, 236, 238, 240–241, 245, 262, 281, 334, 434–436
  GDPS 234

B
background copy 69–70, 73–76, 141, 194, 197, 405
BACKUP xxvii
backup 9–11, 13, 18, 20, 23, 26, 28, 44, 46–47, 62, 73–75, 79–83, 85, 90–91, 93–94, 102, 131, 134, 140–146, 153–155, 157, 204, 210, 220, 228, 232, 234, 252, 254–255, 257–258, 260–261, 271, 282, 284, 294–295, 301, 333, 355, 359, 363–367, 369–370, 374, 376–377, 379–382, 389–390, 392–393, 395–397, 399, 402–406, 408, 416–417, 439, 453–459, 464–466, 468
backup and recovery xxix, 9, 45, 59, 79, 83, 132, 142, 364, 390, 415
backup control data set 81, 84
BACKUP SYSTEM 26, 28–30, 59, 68, 83, 85–88, 90, 92–93, 252, 255, 268, 364–371, 379, 386, 390–391, 399–401, 404–406, 463, 468
  DATA ONLY 88–89, 365–366, 370–371, 404
  FULL 29, 83, 88, 90, 365–367, 371, 390–391, 399–400, 405–406
  Operation 406
BCDS 81, 84, 371, 400
BCRS xxviii, 6–7, 10, 13–14, 139
best practices xxviii, 409, 438
BLKSIZE 36, 222, 224–225, 427
BOTH 5, 9, 14, 19–20, 22, 26–27, 33, 35, 37, 58, 67–72, 75, 86, 92–93, 98–99, 104, 108, 111, 113, 120, 123, 131, 135–136, 156, 165, 167, 170–172, 174, 180, 185, 189, 211, 216, 227, 235–236, 239–241, 257, 293, 299, 312, 337, 343, 353, 359, 366–368, 372, 374, 391, 393, 399, 401, 404–406, 420, 426–427, 429, 432, 440–441, 443, 445–446, 454–455
BSDS 16, 24–25, 29–31, 35, 44–47, 49–51, 53–54, 58–63, 81–83, 86–87, 93, 100, 120, 133, 137, 143, 159, 161–162, 254–255, 262–264, 266–267, 270, 276, 285, 295, 300, 315, 323, 331, 336, 346, 350, 367, 370–371, 374–376, 382, 391–394, 400, 408, 413, 417, 420, 425–426, 439, 449, 453, 460–461, 466
  list 46, 263, 417, 420, 460
Buffer pools 345, 349, 351, 360, 373, 412, 414–415, 421, 423, 435, 437, 444–445
Business Continuity and Recovery Services *see* BCRS
business recovery service 10

C
CACHE 99, 117–118, 123, 195, 208–210, 215, 229, 336, 340, 412
cache
  for XRC 209
CASTOUT 428, 435
CCA 101–102, 172, 180, 191, 194, 312, 329–330
CF 27, 61–64, 241–242, 244–245, 255, 285, 366–367, 392, 412, 421
Change Log Inventory 36, 84, 95, 275, 391–392, 420, 426–427
Change Log Inventory utility 31, 60, 62, 254, 262, 285, 365, 367, 374, 382, 420–421
CHECK DATA 38, 298–299, 319
CHECK INDEX 49, 422
check pending 317, 319, 333
CHKP 38, 297–298, 317, 319, 324, 333
CI boundary 275
CI size 29, 300
CICS 18, 23, 44, 46, 56, 117, 120, 234, 294, 435, 437
CLUSTER 234–236, 240, 381
COEXIST 234
collision 23
COMMENT 47
common time reference 114, 116, 122, 204
compression 435, 438, 444, 446, 448
Compression dictionary 444, 447
Concurrent copy 18, 21, 438
conditional restart control record 31, 48, 53, 61–62, 84, 92, 266, 268, 374–375, 421
CONNECT 114, 210, 326, 467–468
Consistency Group 25–26, 30, 34, 77–78, 100, 105–110, 116, 120, 124–126, 131, 136, 139–140, 142–145, 147, 149, 151, 158–163, 167–168, 171, 174, 181–182, 184–187, 189–193, 197–199, 208, 217, 221, 223, 241, 246, 281–286, 288, 290–292, 300, 302, 304, 306, 310, 312–313, 316, 319–321, 325–328, 333–334, 346, 350–351, 355, 357, 359, 361
consistency group

with FlashCopy V2 281
Continuous Availability 235–236, 239, 432
control data set 81, 84, 120, 123, 217, 223–224, 338, 464
controlling
XRC 238
controlling system 234, 238
COPY xxvii, 1, 11, 17, 21, 24, 27–29, 32, 36, 38–39, 43–46, 48–52, 54, 59–65, 67–83, 86–93, 96, 101–104, 109–111, 114, 123, 131–132, 136–141, 144, 147–151, 154–160, 162–163, 166–167, 169–170, 173, 175–177, 179, 187–188, 191, 194, 197–198, 201–205, 209, 212, 214–215, 217, 220, 229, 232, 235, 245, 251, 254–255, 257–258, 260–261, 263–264, 267–268, 270–274, 281–282, 284, 286–288, 290–292, 294–295, 300, 302, 304, 308, 310, 312, 315, 318–320, 323, 325–326, 329, 331, 334, 340–341, 359, 364–367, 369–370, 374, 376–377, 379–382, 389–403, 405, 407–408, 416–417, 420, 422, 424, 427–428, 438–442, 446–448, 454–457, 460, 463–464, 466, 468–469
copy pending 318, 324
COPY status 180, 187
COPY2 47, 50, 255, 261, 263, 420
COPYPOOL 29, 84, 89, 92, 95, 364, 366–367, 370, 377, 379–380, 384, 386, 391, 397–401, 406
COPYTOCOPY 44, 47
COPYVOLID parameter 72, 74, 260, 294, 407
COUNT 53, 85, 117, 184, 259, 266, 268, 271, 296, 317, 324, 332, 375
Coupled Extended Remote Copy, *see* CXRC
Coupling Facility 93, 235–236, 241, 367, 392, 413–415, 421, 423, 427, 469
CRCR 53, 61–64, 266–268, 374
Customer Relationship Management 364
CWDM 234
CXRC 116, 120, 207, 219, 224, 231, 239
CYCLE 58, 60–64, 73, 75, 110, 127, 169, 196, 435

D

DATA CAPTURE 444, 446
data consistency 1, 12, 25, 45, 86, 99, 107, 121–122, 127, 166–167, 206, 214, 232, 239, 281–282, 319, 333, 416–418
data consistent field 342
data corruption xxix, 8, 301, 435
Data Currency 12, 17, 100, 107, 109, 163, 166–168, 172, 232, 282
data exposure field 342
data set creation 29, 87
data sharing xxviii, 16, 24, 29–30, 32–36, 45, 51, 58–65, 83–87, 92–94, 106, 120, 160, 242, 244–245, 253–257, 262–263, 265–267, 275, 284–285, 303, 335–336, 345, 349–350, 360–361, 363–366, 368, 372–374, 376, 383, 387, 390–392, 409, 411, 413–421, 423–428, 430, 437, 441–442, 445, 449, 460–461, 468–469
data space 207
database copy pool 30, 82–83, 86, 92–93, 465
Database name 54–55, 315–316, 331, 352
DATE 23, 44–46, 49–52, 58, 85, 89, 127, 146, 227, 263, 267, 273–274, 277, 371, 398–401, 403, 405, 419, 424, 457–458
DB2 Control Center 22
DB2 data sharing xxix, 25, 59, 270, 355, 359, 365–366, 390–391, 412, 437
DB2 Group Buffer Pools 241
DB2 Lock Structure 415, 421, 423
DB2 tools xxix
DB2 Tracker 18, 57–59, 61
DBD 56, 271, 297–298, 317, 324, 333, 353–354, 378
DBD01 29–30, 49, 60–61, 86–87, 92–93, 138, 257, 269–270, 353, 371, 376, 422
DBET 449
DBM1 414, 429
DBRM 47, 420
DDF 285, 296, 313, 316, 323, 332, 352–353, 391–392, 428
DDL 22, 64, 132, 260, 468
delay time 216, 342
Dense Wavelength Division Multiplexing 211
dependent writes 21, 24, 77, 102, 105, 124, 127, 140, 166, 282
DFSMS xxvii, 68, 79–80, 139, 143, 156, 165, 251, 281, 301, 304, 393, 435, 444, 448, 454, 464
DFSMSdss
COPYVOLID 72
DUMPCONDITIONING 257
FCNOCOPY 74
FCWITHDRAW 74
invoking FlashCopy 71
RESTORE 146, 254, 284, 391–392
DFSMShsm 29, 79–84, 86–89, 91–93, 96, 363, 370–371, 376, 379–380, 382, 389–393, 397–406, 465–466
disaster xxvii, 9–13, 18, 24–25, 46, 57–58, 63, 100, 115, 155, 160, 167–168, 192, 206, 221, 228, 237, 241–243, 247, 253, 260, 298, 348, 350, 361, 416, 429
disaster recovery xxvii–xxviii, 1, 3–6, 8–9, 11, 13–16, 18, 20–21, 24, 28, 33–34, 36, 38–39, 43–45, 47, 57–60, 65, 70, 74, 86, 97, 101–103, 127–129, 131–132, 136–138, 142, 146–147, 153, 155–158, 163, 165–166, 179, 188, 198, 202, 210, 221, 226, 228, 232, 236, 238, 240, 246, 249, 254, 256, 258, 261, 279, 281, 294, 300–301, 339, 347, 383, 390, 392, 407, 411, 414–417, 419–420, 427–428, 430–433, 436, 448, 453–454
disaster recovery solution xxvii–xxviii, 5, 10, 16, 19, 25, 28, 30, 41, 97, 106, 108, 139–140, 157, 170, 181, 201, 203–204, 214, 228, 233, 235, 238–239, 428
DISPLAY THREAD 259
DISTINCT 14
DONOTBLOCK parameter 117, 215
downtime 5, 131
DSMAX 430
DSN1213I 37, 278–279, 424–425
DSN1COPY 49, 60–64, 422
DSN1LOGP 31, 36–37, 48, 52, 86, 162, 276–277, 279, 383, 421, 424–425, 447
DSN1PRNT 86, 269, 371
DSN6SPRM 47, 49, 59, 420, 422
DSNB250E 54–55, 315–316, 331, 352
DSNB260I 434
DSNB357I 352

DSNI006I 354
DSNI021I 354
DSNJ031I 434
DSNJ139I 46–47, 262, 417, 420
DSNJCNVB 51, 263
DSNJU003 37, 47–51, 53, 60, 62–63, 84, 92–93, 95, 254–255, 262–263, 265, 285, 365, 367, 374, 382, 417, 420, 424, 469
DSNJU004 46–49, 51, 53, 60, 62, 85, 255, 262–263, 371, 374–375, 417, 419–421, 443, 460
DSNR031I 353
DSNR035I 434
DSNT397I 56, 271, 297–298, 317, 324, 333, 353–354, 378
DSNTIJIN 47, 420
DSNTIPE 435
DSNTIPH 444
DSNTIPL 34, 414, 429, 433–434, 443
DSNU1602I 90, 370, 400
DSNUTILB 377, 385
DSNZPARM 33, 36, 54, 59, 255, 261, 266, 270, 295, 315, 322, 330, 385, 391–392, 427, 429, 433–435, 443
  CTHREAD 435
DSSIZE 269, 371
DUMP FULL 73, 75, 258, 294, 407, 458
DUMPCONDITIONING 73, 75, 152, 257–258, 293, 406, 448
DUMPCONDITIONING parameter 72, 261, 295, 407
DWDM 166, 211–212, 234
DWDMs 211
DWQT 445

E

ECSA 208
EDM pool 435
ELB 77, 101, 140, 145, 154, 160, 281–282, 284, 291
ENDING AT 91–92, 149, 289
ENDLRSN 53, 60–64, 85, 161, 253, 255, 262, 266, 268, 277–278, 374–375, 415, 421, 423–424, 426–427
ENDRBA 35, 48, 51, 53–54, 60–63, 85, 161–162, 253, 255, 262–263, 266–268, 270, 277–278, 295, 315, 323, 331, 375, 408, 421, 427
Enterprise Resource Planning 14, 364
ERRORLEVEL parameter 127, 206, 214
ESS FlashCopy 59, 80, 139, 364
establish time 19, 141
EXISTS 58, 68, 74, 83, 160, 174, 180, 195, 220, 224, 380, 423
expanded storage 208
Explain 131, 451
Extended Long Busy 30, 77, 101–102, 140, 160, 281–282, 291, 312–313, 328, 334
extended remote copy 16, 19–20, 113, 116, 166, 203, 234, 239

F

FACILITY class 226
Fallback 13
Fast Log Apply 92, 270, 273–274, 383–384, 414, 429, 438, 442, 449
fast replication 79–82, 86, 88–91, 96, 363–364, 376–377, 379, 381, 389–393, 395, 397–398, 401–406, 463–464, 468
fast replication services 86, 380
FCNOCOPY parameter 74–75
FCP
  PPRC links 107, 111
FCWITHDRAW parameter 74–75
FLA 92, 269–270, 383–385, 429
FlashCopy 18–19, 26–28, 30, 32, 34, 59, 67–72, 74–75, 77–81, 87, 105, 107–108, 110–111, 128, 131–134, 136–142, 144–149, 151–152, 154, 157, 160–163, 165–166, 168–169, 171–172, 174, 178–179, 182, 185, 188, 190, 192–193, 197–198, 203–204, 210, 220, 226, 228, 232, 239, 251–254, 256–260, 268, 271, 273–274, 279, 281–287, 290–297, 300–304, 310, 312–316, 319–323, 325–327, 330, 332–333, 364, 368, 386, 389, 392, 396, 405–406
  background copy 68–69, 110, 140, 168, 406
  DFSMSdss 68, 71–72, 139, 232, 254, 284, 302
  persistent relationship 70, 76
Forward Log Recovery 442
forward log recovery 260, 375
FRBACKUP 80, 82, 90–91, 391, 397–398, 401–403, 405–406
FRDELETE 80
freeze
  FlashCopy consistency group 77, 148, 287
FRRECOV 80, 92, 366–367, 379–380
full system backup 86–87, 92, 399
fuzzy copy 105

G

GDPS 21, 26, 31, 102, 173, 229, 233–234, 240–247, 334
  controlling system 234
  GDPS/PPRC 237
  Hyperswap 236–237
Geographically Dispersed Parallel Sysplex 26, 101, 221, 233, 240
Global Mirror for z/OS 201, 203
granularity 19, 439, 447
GRECP 27, 244–245, 285, 351–355, 361, 367, 386, 392, 428–429
group restart 244, 345–347, 351, 414–415, 421–423
group_name 206

H

health check 432
hole 161
how it works
  Asynchronous PPRC 109
HTTP 4, 10, 12, 14, 131, 139, 208, 221, 230, 233
Hyperswap 27, 236–237, 242–243, 245–247

I

IBM DB2 Archive Log Accelerator for z/OS 448
IBM DB2 Change Accumulation Tool for z/OS 448

IBM DB2 Tools xxix
ICF catalog 17, 46–47, 133, 156, 314, 322, 330, 336, 343, 348, 359, 369, 374, 379–380, 390–391, 393–394, 420
IDCAMS 47, 50, 262, 264, 382, 420
IEA491E 100–102, 106
IEA494I message 100–101
IFCID 0225 435
IFCID 0313 434
II10817 435
IMS xxix, 18, 44, 46, 56, 117, 120, 234, 294
inabort 294
incommit 294–295, 315, 330
index changes 271, 434
indoubt units of recovery 422, 428
indoubt URs 48, 93, 283, 428
in-flight 407
inflight 92, 294, 315, 322, 330, 365
insert 21, 24, 38, 260, 372–373, 444–446, 468–469
installation panel
  DSNTIPE 434
  DSNTIPI 413
ISO 4, 14
ITERATE 149, 289

J

Java xxix, 194
journal data set 120, 126, 221, 223, 342

L

LIMIT BACKOUT 16, 34, 352
LIST COPYPOOL 382, 398, 400, 403–405
List prefetch 429
LOB 372
location name 83, 85, 371
lock contention 433
locking 33, 432, 434
Log Analysis Tool 447
log data sets 24, 29, 31, 46, 255, 414–418, 420, 424–425, 439–440, 442–444, 446, 460
LOG NO events 92, 260, 268, 271, 273, 377
log record sequence number 413
Log truncation point 85, 92–94, 255, 265, 365–367, 372–374, 376, 382, 449, 469
LOGAPSTG 383, 385, 429
log-based recovery 445–447
logical partition 240, 256, 285, 303, 368, 392
loop 237, 449
lost data 5, 282
LPAR 116–117, 120, 134, 163, 207, 232, 237, 240, 256, 285, 292, 303, 336, 368, 392, 427
LPL 54–56, 100, 316–317, 331, 333, 351–352, 354–355, 366–367, 377, 379, 386, 429, 433, 449
LPL recovery 352–355, 361, 367, 392, 428
LRDRTHLD 434
LRSN 32, 36, 54–55, 61–62, 64, 84–85, 87–88, 138, 252, 255, 257, 265–266, 272, 275–276, 282, 315–316, 320, 331, 352, 355, 366–367, 371–375, 377, 384, 401, 413–414, 423–425, 429, 441, 443, 446, 469
  minimum 374, 424

M

managing
  XRC 221, 239
master 108, 110, 116, 120, 167, 169–171, 174–175, 179–180, 183, 187, 189–190, 192, 197, 205–206, 213, 217–220
master data set 224–225
MAXDBAT 435
messages
  IEA494I 102, 291
Migration 20, 136, 202, 206, 221, 228, 442, 464
MODIFY 29, 32, 49, 60, 173–174, 187, 190, 195, 231, 260, 294, 351, 374, 407, 422, 441
Multiple Extended Remote Copy, *see* MXRC
MXRC 116

N

new function mode 255, 336, 368, 448
NFM 84, 87, 255
NOCACHE 195
NOCOPYVOLID 257
NOVTOCENQ 84, 391, 401–402
NUMPARTS 269, 371

O

OA04877 195
OA06196 71
Object 30–31, 39, 62–63, 100, 246, 259, 271, 296–297, 317, 324, 332, 352, 364, 366, 380, 386, 391, 428, 430, 433, 438–439, 441–442, 445, 467
Online REORG 39, 87, 372, 378, 386
ORDER 5–6, 12, 15, 20, 25, 29, 33, 36, 41, 44, 46, 49, 62, 68, 71, 75, 77, 82, 105–106, 123, 134, 144–145, 159, 161, 166, 174, 195, 209, 213, 219, 242, 256, 259, 270, 292, 298, 307, 346, 350, 359, 361, 380, 384, 391–392, 401, 416–417, 419, 422, 426, 430, 432, 434, 438, 441–442, 446, 456
OS/390 V2R10 444
outages
  unplanned 118
OW48234 257, 407, 448
OW57347 71

P

packaging 154
PAGEFIX parameter 208
PARTITION 33, 145, 240, 256, 285, 303, 351, 368, 392, 438–439, 441, 443–444
PARTITIONED 441
partitioning index 441
peer-to-peer remote copy 16, 19, 97, 196, 234, 301
performance xxix, 11, 19–21, 23, 33, 58, 83, 99, 102, 109, 114, 121, 123, 139, 141–143, 157, 168, 195–196, 209, 222, 224, 230, 237–238, 304, 337, 364, 386, 405, 428–431, 438, 440–442, 444–446, 449, 464
PERMIT 148, 226–227, 309

persistent
FlashCopy relationship 19, 68, 189
PITR 29, 36, 86, 96, 255, 363–364, 368–369, 378, 387, 463–464, 468
scenarios 96, 363
Point-In-Time xxvii, 18–21, 67–68, 75, 77, 83, 102–103, 106, 108, 116, 119, 124, 132, 136, 141, 157–158, 160, 166, 205, 232, 239, 252–253, 265, 268, 276, 282, 334, 355, 359, 363–366, 369–371, 373–374, 379–380, 389, 393, 433, 447, 449
PORT 296, 304, 316, 323, 332
POSITION 419
postponed abort 294
PPRC 16, 18–21, 25, 28, 59, 67–68, 71, 97–112, 131–138, 140, 157–161, 163, 165–176, 178–191, 193–195, 197–198, 220, 234, 240–241, 245–247, 301–304, 306–308, 310, 312–314, 319–322, 325–330, 333–334, 396
consistency grouping 102
GDPS/PPRC 27, 235–237, 239
how it works 109
operation 19–20, 98, 158, 236, 326–327
PPRC-XD
characteristics 103
data consistency 108, 166
fuzzy copy 20, 103, 106, 158
PQ69741 430, 449
PQ73038 31
PQ86382 449
PQ86492 449
PQ87542 449
PQ88307 449
PQ88728 449
PQ89297 270, 382, 384, 449
PQ89742 449
PQ90764 449
PQ90795 449
PQ91099 382, 449
PQ91102 449
PQ92187 382, 449
PQ93548 383, 449
PQ94793 449
PQ95164 386, 449
PRECISION 448
PREFETCH 429
Print Log Map 35–36, 85, 138, 161–162, 276, 419, 424–426
Print Log Map utility 45, 48, 51, 60, 62, 255, 262, 371, 374, 421
prior point in time 86, 93, 230, 433
PSRBD 271, 364, 377
pull 23
push 23

Q

Q Replication 23–24
QUALIFIER 31, 36, 217, 220, 225, 337, 339, 426
QUICKCOPY option 212, 216
QUIESCE 32, 36, 46, 86, 106, 136, 236, 252, 282, 342–343, 348, 359, 391, 401, 416–419

R

RACF 91–92, 205–206, 226
XRC TSO commands 226
RBA 29, 31–32, 35–37, 47–55, 61, 84–85, 87–88, 138, 159, 161–162, 252, 255, 257, 262–268, 270, 272, 275–279, 281–282, 295–296, 300, 315–316, 320, 323, 331, 346, 350–353, 365, 371–373, 375–376, 381, 383, 401, 408, 414, 417–420, 424–425, 441, 443, 469
RBDP 39, 61, 92, 256, 270–274, 297–298, 317, 324, 333, 364, 366–367, 377–378, 380, 386, 469
RBLP 29–30, 85–88, 92–93, 138, 257, 265, 269–270, 273–274, 279, 281, 300, 371, 374, 376–377, 386
RCMF 212, 221, 240
real memory 256, 285, 303, 336, 368, 392
real storage 123, 125–127, 208, 435
REBP 318
REBUILD INDEX 32, 95, 273–274, 298, 318, 366–367, 378, 449
Rebuild pending 61, 256, 317–318, 333
RECOVER 1, 14, 21, 24, 28–29, 31, 34, 38, 43–45, 47–48, 55–56, 58–64, 80–81, 83–84, 86, 92, 94, 100, 115, 128, 131–132, 136, 153, 167, 213, 219–220, 226, 235, 240, 244, 252–255, 268, 270–272, 274, 279, 285, 294, 296, 298, 316–318, 323–324, 332–333, 343–344, 346–349, 351–353, 359–360, 363–367, 369, 371–373, 377–380, 383–384, 408, 414, 416–418, 420, 422, 428, 433, 436–437, 439, 441–443, 447, 449, 469
RECOVER INDOUBT 48, 422
recover the BSDS 48, 262, 420
recovery 28
XRC sessions 116, 224
recovery base log point 29–30, 86, 138, 257, 269–270
recovery pending 244, 256, 318, 351
Recovery Point Objective 5, 107, 235, 432
Recovery Time Objective 5, 235, 432
RECP 38, 61, 92, 256, 270–274, 317–318, 324, 333, 364, 366–367, 377–378, 380, 386, 469
REGION xxix, 152–153, 231, 293, 382, 454, 458, 460
RELOAD phase 38, 323, 332
remote copy 9, 11, 16, 18–21, 26, 97–98, 100, 106–107, 113, 116, 125, 136, 165–166, 191–192, 196, 203–204, 212, 221, 224, 229, 234–235, 237–240, 301
REORG 38–39, 60, 62–63, 87, 259–260, 273, 279, 296, 312, 316–318, 320, 323–324, 326, 332, 372, 378, 386, 447
reload phase 39, 316
REORG TABLESPACE 372
REPAIR 38, 60, 62–63, 231, 271, 298–299, 318
REPEAT 149, 151, 162, 288, 291, 307
Replication Center 23
report 22, 31, 37, 46–48, 52, 64, 127, 189, 195, 210, 216, 219–220, 230, 259, 262, 277, 292, 340–342, 344, 347–349, 356–360, 372, 417, 420, 443, 454, 456–457, 460–461
REPRO 35–36, 50, 60, 62, 254, 262, 264, 420, 425, 427
RESET 38, 76, 85, 93, 95, 118, 154, 192, 255, 266, 270, 282, 291, 312–313, 318–319, 324, 333, 365, 367, 378, 469
residual records 127
RESPORT 296, 316, 323, 332

RESTART xxix
restart 1, 8, 11, 17–18, 20–21, 25–26, 28, 30–34, 36, 38, 46, 48–49, 53–55, 59, 61–64, 85–86, 92, 95, 109, 115, 128, 137–138, 140, 143, 160–162, 166, 168, 198, 204, 210, 214, 219, 231–232, 234, 240, 244–247, 253, 255, 259, 265–268, 270–272, 274–276, 278, 281, 283–285, 294–296, 301–302, 310, 315–316, 319–320, 322–323, 325, 330–333, 335, 342, 345–347, 349–353, 356–357, 360–361, 364–367, 374–375, 378–379, 382, 386, 391–392, 407, 413–415, 417, 419, 421, 423, 426–429, 437, 440, 442, 449, 459, 469
RESTART WITH xxvii, 27, 375
RESTORE xxvii, 10, 13, 21, 24, 29, 34, 36, 44–45, 47–48, 50, 60–63, 73–74, 86, 95, 138–139, 142–147, 152–154, 232, 247, 252, 258, 261–262, 283–284, 294–295, 365–366, 373, 379–380, 383, 391–392, 405, 407, 416–417, 420, 427, 438–440, 442, 448, 454–456, 459, 469
RESTORE SYSTEM 27, 29–30, 34, 36, 58–61, 63–65, 68, 83–86, 92–95, 138, 161, 163, 251, 253–255, 260, 265–266, 268–271, 273–274, 279, 300, 364–365, 367–369, 371, 374–382, 384, 387, 432, 449, 463, 468–469
Example 95, 161, 382, 468
Execution 95
LOGONLY 27, 30, 36, 58–61, 63–65, 93–94, 138, 161, 163, 251–255, 260, 268–271, 279, 300, 449, 469
Operation 86, 271
RESTRICT 56, 226, 256, 271, 297–299, 317, 324, 353, 365–367, 378, 391–392, 404, 408, 429, 469
RESYNC 104, 135, 137–138, 198, 304, 358
Retained locks 346–347, 350–351, 361, 413, 427–428
RID Pool 435
RMF 195, 435
ROLLBACK 156
rolling disaster 1, 12, 19, 24, 26, 101, 197, 228, 240–242, 245, 334
RPO 5, 13, 16, 27, 34, 41, 106–107, 138, 140, 160–161, 163, 181, 199, 235, 238, 245, 432, 436
RTO 5, 12–13, 16, 27, 34, 41, 110, 138, 140, 144–145, 163, 168, 235, 238, 245, 432, 436
RUNSTATS 38, 372

S

SCA 86, 244, 345–346, 349, 360, 373, 413, 415, 421, 423
Scalability 239
SCHEMA 312, 320, 326
SCOPE ALL 299, 319
SDM 27, 114–115, 117–121, 123–128, 163, 202, 207–209, 212, 214–215, 217, 219–221, 223, 225–228, 230–232, 239, 246, 336–337, 340–341, 343–344, 347–349, 356–357, 359–360
requirements 207–208
SDSF 291, 312–313, 321, 326, 328
SDSNEXIT 460
Session 20, 107–110, 116–121, 123, 125–128, 165–174, 176, 179–181, 183–191, 193–195, 197–198, 205–207, 212–216, 218–220, 222–225, 227, 229–231, 247, 304, 326, 337, 339–344, 347–349, 356–360
session management 111, 174
session_id 222, 224–225, 337, 339
sessions
suspending 228
XCOUPLE command 220
XEND command 218
SET xxix, 10, 18–19, 21–22, 24, 29, 31–32, 34–36, 38–39, 46, 48–52, 54–55, 59–60, 62, 68, 71, 73, 76–77, 79–81, 84, 87, 91, 99, 102, 105, 107–108, 116, 118, 120, 123, 126, 133–134, 137, 139–140, 142–146, 151–152, 154, 161, 163, 166–167, 172–174, 179, 181, 188, 192, 195, 197, 204–206, 212, 216, 219, 221, 223–226, 228, 239–240, 245–246, 255, 258, 261, 263, 267, 270–274, 277–278, 284, 293–295, 298–299, 301–302, 306, 310, 312, 315–316, 318, 323, 327, 331, 338–340, 342, 351–352, 355, 366–367, 369, 377, 380–382, 384–385, 390, 397–398, 408, 412–414, 416–417, 419–421, 424–426, 428–432, 434–443, 445–446, 448–449, 453–456, 458, 460–461, 464, 468
SET LOG RESUME 29, 59, 86, 93, 136, 160, 163, 252, 254, 257–259, 268, 281, 319–321, 372, 391, 401, 403, 468–469
SET LOG SUSPEND 26–27, 29–30, 59, 86, 93, 103, 105–106, 135, 138, 158–159, 162–163, 252, 254, 257, 259, 265, 268–269, 281–282, 300, 302, 319–321, 333, 364, 371–373, 386, 391, 401–402, 468–469
SIGNAL 99, 175, 211
SMF 17, 195, 205, 210, 230, 434
SMF type 42 230
SMS-managed data sets 59
SNA 21
sort pool 435
SORTDEVT 299, 319, 372
SORTNUM 299, 319, 372
SQLJ xxix
SSID 84, 101, 189, 194, 237, 312, 329–330, 454–455, 457
START DATABASE 85, 244, 354, 367, 386, 392, 422, 428–429, 433
START WITH 10, 17, 36, 119
STARTLRSN 60, 62, 255, 262, 277–278, 421, 427
STARTRBA 36–37, 48, 51–54, 60, 62, 85, 255, 262–263, 266–268, 270, 276–278, 295, 315, 323, 331, 375, 408, 421, 427
state data set 120, 209, 222, 224, 229, 338
Statistics Class 3 434
striping 341, 347–348, 355, 361, 438, 444, 448
structures 13, 17, 61–64, 75, 93, 190–191, 244, 255, 285, 345, 349–350, 360–361, 366–367, 373, 392, 412–415, 421, 423, 464, 469
subject 22, 171, 208
subordinate 108, 110, 167, 169–171, 174–175, 179, 188–190, 193
suspend 20, 26–27, 29–30, 59, 86–87, 93, 102–106, 117, 127–128, 131, 133, 135–136, 138, 158–160, 162–163, 172, 198–199, 205, 207, 217–219, 225, 228–229, 251–252, 254–255, 257, 259, 265, 268–269, 279, 281–282, 300, 302, 304, 319–321, 333–334, 355–357, 359, 364, 371–373, 386, 391, 401–402, 468–469

SYSADM 48, 269, 354, 376–377, 384–385, 422, 429
SYSCOPY 21, 28, 32, 36, 274, 449
SYSIN 37–38, 50–53, 72–75, 88, 90, 94–95, 152–153, 162, 257–258, 261–265, 269, 272–274, 277, 279, 293–295, 298–299, 318–319, 370, 374, 376–377, 380, 382, 384, 399–400
SYSLGRNX 29, 31–32, 64, 354, 428, 438–439, 441, 443
SYSPITR 31, 35–37, 84, 94, 161, 253, 265–266, 268, 275–277, 279, 365, 367, 373–374
SYSPITR CRCR 84–85, 92–93, 255, 265–266, 374–375, 382, 469
SYSPRINT 37, 50–53, 72–75, 152–153, 162, 257–258, 261–265, 269, 277, 279, 293–294, 370, 374, 376, 382, 399, 407, 458–459
System checkpoints 33, 87, 382, 386, 433–434, 443
System Data Mover 26, 114–115, 119, 123, 202, 204, 218, 238–239
system level point in time recovery 84, 93, 252, 267, 276, 363
    BACKUP SYSTEM 363
    RESTORE SYSTEM 93, 252, 363
system level point-in-time recovery 275
System recover pending mode 255, 266–267, 269–270, 375–376

T

TCP/IP 195, 296, 316, 323, 332
TEXT 383, 448
TIME 5, 8–13, 16–21, 23–26, 28–29, 31, 33, 35, 37–38, 44–46, 49–54, 58–61, 64, 67–71, 75, 77, 79, 81, 83–86, 88–89, 92–95, 97–98, 100, 102–106, 108–111, 114–120, 122–128, 132–134, 136, 138, 140–146, 152–153, 155, 157–163, 166–169, 173–174, 180–181, 187, 189–190, 192, 196, 198, 204–207, 210, 216–217, 219–221, 229–232, 234–235, 239, 241–242, 244, 247, 251–255, 257–258, 263–264, 266–268, 270–271, 273–277, 282, 291, 293, 296, 298, 300, 312, 315, 319, 322–323, 328, 330–331, 334, 339–352, 355–361, 363–367, 369–371, 373–380, 382, 384–386, 389, 392–393, 397–401, 403–405, 408, 412–414, 416–419, 423–434, 436, 438–443, 445, 447–449, 453–454, 457–458, 460–461, 463, 468
TIMESTAMP 20, 31, 51, 54, 125, 127, 216–217, 219, 263, 266–267, 270, 295, 315, 323, 331, 341–344, 346, 348–350, 357, 359–360, 408, 413
TOKEN 82, 84, 89, 95, 259, 292, 366–367, 370–371, 377, 380, 384, 391, 398–403, 405, 457
tracker site xxvii, 16, 27, 57–65
two-phase commit 11

U

UNIQUE 81, 83, 140, 146, 232, 372, 402
UQ74865 449
UQ74866 449
UQ86901 449
UQ87589 449
UQ89106 449
UQ89350 449
UQ91265 449
UQ91315 449
UQ91525 383, 449
UQ91590 449
UQ91718 449
UQ92020 449
UQ92067 384, 449
UQ92442 449
UQ92875 449
UQ93407 449
URCHKTH 433–434
URLGWTH 434
USAGE xxix
Utility
    RESTART 31, 34, 38, 61, 84, 162, 266, 297, 374–375, 408, 424
utility device 121, 212
UW79364 448
UW79365 448
UW79366 448

V

VDWQT 445
Version xxvii, xxix, 17, 19, 21, 26, 28, 34, 38, 43, 49, 67–68, 79–80, 89, 93–94, 106, 111, 114, 117, 139–140, 146, 162, 195–196, 203, 237, 270, 335, 363, 369, 371, 376, 379, 381, 386, 397–405, 412–414, 418, 423, 427–428, 433, 435, 442, 444, 448, 457, 461
version information 86, 88
VOLATILE 99, 105
volume dump 45, 73–74, 93, 416, 448
volume level backups 83, 363, 390, 399
VTAM 21

W

WARM 8
WebSphere 23, 437
WebSphere Application Server 194
Wizard 23, 150–151
WLM 442

X

XADDPAIR command
    processing 212
XADVANCE command 219
XCOUPLE command 220, 225
XD option 104–105
XDELPAIR command 217–218
XEND command 218–219, 224, 343, 347
XQUERY command 216, 227, 340
XQUERY MASTER command 217
XRC 16, 19–20, 26–27, 59, 67, 71, 113–123, 125–128, 162–163, 166–167, 195, 201–210, 212, 214, 216–217, 219–220, 222–226, 228–231, 234, 240, 245–247, 335–344, 346–351, 355–361, 396
    components 114, 230
    consistency groups 123
    control data set 338
    diagnostic aids 230

GDPS/XRC 221, 235, 237–239
hardware bitmap 114
journal data set 120, 222, 337
master data set 120, 224
state data set 120, 224, 338
status CPY 340
status DUP 341
status PND 340
testing 113, 227, 232
utility device 121, 215
XRECOVER 119, 128, 214, 224, 226, 232, 349, 360
XRECOVER command 126, 128, 205, 219, 224, 343–344, 347–349, 355, 359–361
XSET command 121, 208, 219
XSTART command 117, 119, 206–207, 214–215, 219, 339, 343
XSUSPEND command 218–219, 357, 359

Z

z/OS xxviii–xxix, 17, 19, 21, 23–24, 26–29, 34, 38, 43, 49, 59–61, 65, 68, 79–80, 83, 86, 93, 95, 107–108, 110, 113–115, 118–119, 121, 123, 132–134, 137, 139–140, 142–146, 151–152, 154–155, 162–163, 165–167, 169, 172–173, 189, 195, 198, 201–204, 207, 212, 216, 220, 230, 233–237, 240–244, 246, 251, 255–256, 281, 285, 301, 303–304, 335–336, 363–365, 368, 383, 389–390, 392, 407, 412–414, 428, 433, 435, 437, 439, 441–442, 444, 447, 449, 464, 468–469
z/OS V1R3 93, 230, 268
z900 33
z990 33
zSeries xxix, 19–21, 67, 71, 80, 97, 110, 113, 116, 159–160, 165, 173, 201, 221, 233–234, 240, 256–257, 281, 285, 301, 303–304, 336, 368, 392

Redbooks

Disaster Recovery with DB2 UDB for z/OS

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Disaster Recovery with DB2 UDB for z/OS

Examine your choices for local or remote site recovery

Recover your DB2 system to a point in time

Adopt best practices for recovery execution

DB2 for z/OS is the database of choice for critical data for many enterprises. It is becoming more and more important to protect this data in case of disaster and to be able to restart with a consistent copy of the DB2 data as quick as possible and with minimal losses.

A broad range of functions can be used for the disaster recovery of DB2 subsystems. The traditional DB2 based solution consists of safe keeping and restoring image copies and logs. More general functions, applicable not only to DB2 data, but to the whole system, are hardware related, such as tape vaulting or disk volumes mirroring. Other functions are specific to DB2 such as the Tracker Site. There are also products providing replication capabilities which can be used for specific propagation requirements.

DB2 UDB for z/OS Version 8 has introduced two new subsystem wide utilities, BACKUP and RESTORE, which, by interfacing the copy pools functions of DFSMS 1.5, are able to provide Point-In-Time recovery capabilities.

The disaster recovery solution consists of the combination of coherent options that best fit in with the requirements, the current environment, and the investment.

In this IBM Redbook we first introduce the main concepts, and the primary components for possible solutions. We then describe the most common solutions, and implement several recovery scenarios. All our tests were implemented with DB2 UDB for z/OS Version 8. We also include criteria for choosing a solution, and recommendations based on recovery best practices.

We focus on requirements and functions available for a disaster recovery strategy for data stored and managed by DB2 for z/OS. It is worth remembering that the non-DB2 data, logically or physically related to the DB2 applications, should be treated with equivalent and congruent solutions.

SG24-6370-00 ISBN 073849092X

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks