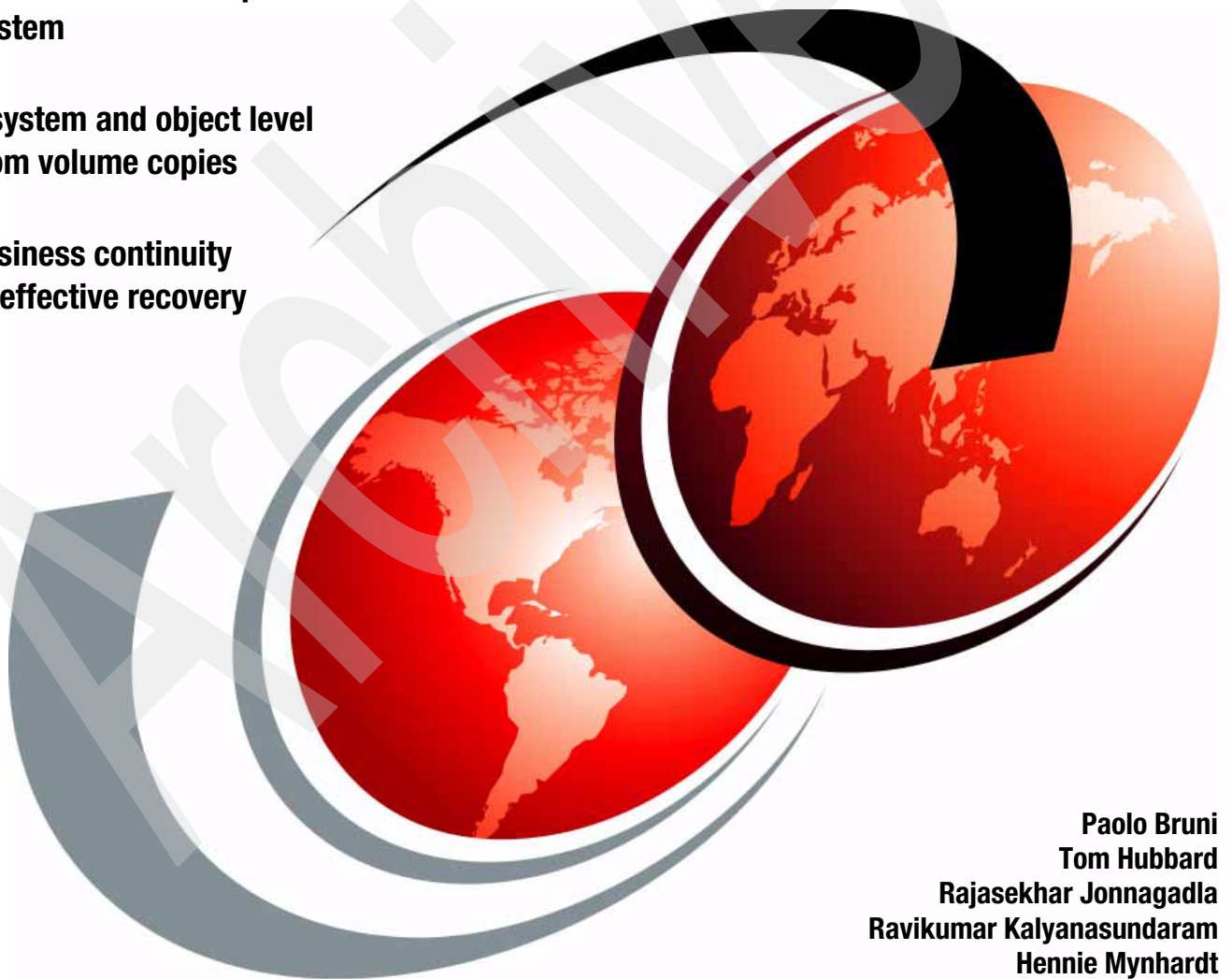


Optimizing Restore and Recovery Solutions with DB2 Recovery Expert for z/OS V2.1

Manage and validate backup and restore system

Optimize system and object level restore from volume copies

Ensure business continuity with cost-effective recovery



Paolo Bruni
Tom Hubbard
Rajasekhar Jonnagadla
Ravikumar Kalyanasundaram
Hennie Mynhardt

Redbooks



International Technical Support Organization

**Optimizing Restore and Recovery Solutions with DB2
Recovery Expert for z/OS V2.1**

June 2008

Archived

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

Archived

First Edition (June 2008)

This edition applies to Version 2, Release 1 of IBM DB2 Recovery Expert for z/OS (program number 5697-N92) for use with IBM DB2 for z/OS and OS/390 Version 7 (program number 5675-DB2), DB2 for z/OS Version 8 (program number 5625-DB2), and DB2 Version 9.1 for z/OS (program number 5635-DB2).

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables	vii
Examples	ix
Figures	xi
Notices	xv
Trademarks	xvi
Preface	xvii
The team that wrote this book	xvii
Become a published author	xix
Comments welcome	xix
Summary of changes	xxi
June 2008, First Edition	xxi
October 2008, First Update	xxi
February 2009, Second Update	xxi
Part 1. Architecture and installation	1
Chapter 1. Introduction to DB2 Recovery Expert V2.1	3
1.1 Introduction to DB2 Recovery Expert for z/OS	4
1.2 What is new in Recovery Expert V2.1	5
1.2.1 System Backup and Restore Services	5
1.2.2 Remote site recovery	7
1.3 Recovery Expert architecture	7
1.4 GUI and ISPF user interfaces	11
1.4.1 DB2 object recovery support	13
1.4.2 Recover groups of objects	14
1.4.3 Recovering an entire DB2 subsystem	14
1.4.4 DB2 log analysis	15
Chapter 2. Installation	17
2.1 Installation requirements	18
2.2 Installation	20
2.3 Migration from Version 1.1	32
Chapter 3. Our test environment	39
3.1 GUI environment configuration	40
3.2 ISPF environment configuration	44
Part 2. Backup and restore services	51
Chapter 4. Configuring DB2 for system backup and restore	53
4.1 Suggested data set naming conventions	54
4.1.1 BSDS and logs	54
4.1.2 DB2 catalog, directory, and user data	54

4.2	Planning for subsystem configuration	55
4.3	Using the subsystem setup utility	56
4.3.1	Defining the new ICF user catalogs	63
4.3.2	Creating the DFSMS copy pools.	66
4.3.3	Renaming and moving the BSDS and log data sets.	66
4.3.4	Moving the DB2 table space data sets to new volumes	74
4.3.5	Merging the DB2 table space alias to its new ICF user catalog	76
4.3.6	Optimal subsystem configuration	78
4.4	Profiles	80
4.4.1	Backup profiles	80
4.4.2	Object recovery profiles	81
4.4.3	Disaster recovery profiles	81
	Chapter 5. Local system level backup and restore	83
5.1	System level backup using DB2 and Recovery Expert GUI to perform the restore . . .	84
5.2	System level backup and restore using Recover Expert ISPF-based System Backup and Restore Services	99
	Chapter 6. Object recovery functions from system level backup.	125
6.1	Object recovery with Recovery Expert under DB2 V8.	126
6.1.1	Managing object profiles for recovery	126
6.1.2	Recover object to current using system level backup.	135
6.1.3	Recover object to point in time using system level backup.	138
6.1.4	Recovery object to copy (full or incremental copy or quiesce)	138
6.2	Object Recovery using DB2 9 for z/OS system level backup	139
6.3	Object recovery with Recovery Expert using system level backup	143
6.4	PPRC/XRC restriction for using data set level recover from FlashCopy.	147
6.5	Optimal recovery action with DB2 Recovery Expert	147
	Part 3. Managing disaster recovery	149
	Chapter 7. Traditional image copy recovery.	151
7.1	Overview	152
7.2	Planning for disaster recovery.	153
7.2.1	Dump and restore of DB2 environment.	153
7.2.2	Consistent point-in-time recovery	154
7.2.3	Continuous archive log transfer	154
7.3	Preparing for recovery at the local site	155
7.3.1	Create the disaster recovery profiles	155
7.3.2	Create the application recovery jobs.	158
7.3.3	Build the disaster recovery profiles.	158
7.4	Recovery procedures	162
7.4.1	Establish the environment.	162
7.4.2	Contents of the recovery PDS	163
7.4.3	Run the job in member ssid#JCL1	165
7.4.4	Restart the DB2 subsystems.	169
7.4.5	Run the job in member ssid#JC3	171
7.5	Considerations	172
	Chapter 8. System level backup based remote recovery	173
8.1	Offloading to tape at the local site.	174
8.1.1	Offloading to tape during a system level backup	174
8.1.2	Offloading to tape as a separate task	179

8.2 Building Disaster Recovery JCL at the local site	184
8.3 Restoring steps at the disaster recovery site	193
Part 4. Appendixes	197
Appendix A. Frequently asked questions	199
Appendix B. Recommended maintenance	207
Appendix C. Storage and DB2 functions	209
FlashCopy	210
Introducing FlashCopy	210
How FlashCopy works	211
DFSMSdss utility	214
Incremental FlashCopy	218
FlashCopy consistency group	219
DFSMS functions for DB2 point in time recovery	221
DFSMSHsm Fast Replication	221
Preparing for DFSMSHsm Fast Replication	224
DB2 BACKUP and RESTORE functions in V8	224
DB2 PITR using Fast Replication	227
DB2 Restore using Fast Replication backups	234
BACKUP and RESTORE enhancements with DB2 9 for z/OS	238
Object-level recovery	239
Tape support for BACKUP SYSTEM	241
Incremental FlashCopy	243
RECOVER utility enhancements for point-in-time recovery	245
Appendix D. Contents of schema level repository	247
Overview	248
SLR tables	250
SLR maintenance considerations	253
Appendix E. Sample jobs	263
Control statement listings for Recovery Expert Agents 63 and 67	264
Configuration files for Recovery Expert Agent 59	278
Control file job listing for Recovery Expert Agent 63 and 64	286
Sample job to create the repository VSAM data sets	293
Sample job to create DR objects	301
Sample CLISTS	303
Sample PARMLIB member	306
Related publications	311
IBM Redbooks	311
Other publications	311
Online resources	311
How to get Redbooks	312
Help from IBM	312
Index	313

Archived

Tables

1-1	Comparison of functions available through the GUI and the ISPF interface.	12
2-1	Steps for installing Recovery Expert V2.1 - complete install.	21
2-2	Mapping of data set names used in ARYSJ001 to DD name in ARYV210 CLISTs . . .	25
2-3	ARYV210 CLIST	30
2-4	RACF profile	31
2-5	Steps for migration from Recovery Expert V1.1 to V2.1	33
3-1	Repositories and corresponding VSAM data sets	46
3-2	SBRS DB2 DR objects	48
5-1	Comparing Recovery Expert GUI and ISPF system level backup and recovery	122
7-1	Summary of recovery steps to rebuild the subsystem at the remote site	163
8-1	Summary of steps built in the job to RESTORE the subsystem at the remote site. . .	191
A-1	Application development frequently asked questions.	200
B-1	Recovery Expert V2.1 current APARs	207
B-2	DB2 Backup and Restore current APARs	208
C-1	The DB2 token	225
C-2	DB2 token contents breakout of Example C-4 output listing	225
D-1	SLR tables in comparison with DB2 catalog tables	251

Archived

Examples

2-1	Sample SYSIN card for ARYSJ001	25
2-2	Sample configuration file for the IBM DB2 Recovery Expert for z/OS 2.1.0 Agent	27
2-3	Sample configuration file for the IBM DB2 Recovery Expert or z/OS 2.1.0 server	28
3-1	Snippets from server configuration file	41
3-2	Snippets of Recovery Expert Agent 53 agent configuration file on LPAR SC53	41
3-3	Snippets of Recovery Expert Agent 67 agent configuration file on LPAR SC67	41
3-4	Snippets of Recovery Expert Agent 59 agent configuration file	43
3-5	Snippets of the Recovery Expert Agent 63 configuration file	43
3-6	Snippets of the Recovery Expert Agent 64 configuration file	44
4-1	JCL and control cards to merge catalog alias	77
5-1	BACKUP SYSTEM JCL	85
5-2	FAST REPLICATION output in DFSMSHsm for the DB COPYPOOL	85
5-3	FAST REPLICATION output in DFSMSHsm for the LOG COPYPOOL	86
5-4	BSDS BACKUP SYSTEM UTILITY HISTORY output from DSNJU004	90
5-5	DSNJU003 - Recovery Expert JOB1	94
5-6	Output of "JOB1" - create a conditional restart control record in the BSDS	95
5-7	Displaying the XCF structures	95
5-8	Deleting the lock structure	95
5-9	Deleting the SCA structure	96
5-10	Display to verify delete	96
5-11	Acknowledgement of the restart record truncation	96
5-12	DB2 restarted with ACCESS(MAINT)	96
5-13	The SCA structure being rebuild	97
5-14	Closing the ICF catalogs	97
5-15	Unallocating the ICF catalogs	98
5-16	Example of the RESTORE SYSTEM job	98
5-17	Recovery messages in DFSMSHsm log	99
5-18	Output of backup JCL with SETUP	111
5-19	Output of Recovery Expert generated system level backup	112
5-20	DSNJU004 Output - BACKUP SYSTEM UTILITY HISTORY - showing the backup entry	116
6-1	SMS output on successful system level backup	140
6-2	Output from a DB2 RECOVER utility that selects FLASHCOPY as input	140
6-3	DFSMSsms output - RECOVER TS with 5 data sets selecting FlashCopy backup	141
6-4	Job steps created by Recovery Expert for object recovery	143
8-1	Example of an offload to tape	183
8-2	Output from the system backup offload job run at the local site	193
8-3	Output from the step to restore the volumes backups from tape	193
8-4	Output from the deleting the BSDS and log data sets	194
8-5	Output of defining the active logs and BSDS data sets	194
8-6	IDCAMS REPRO output for the restore of the BSDS	195
8-7	Conditional restart record creation output	195
8-8	Replying to CRCR outstanding message	196
C-1	DFSMSdss COPY FULL with COPYVOLID	215
C-2	Backup-restore cycle with DUMPCONDITIONING and background copy	216
C-3	Backup to tape with no-background copy	217
C-4	DFSMSHsm LIST command output	225
C-5	DSNJU004 output	226

C-6	Invoking DFSMSHsm replication with DB2 utility	230
C-7	Invoking DFSMSHsm Fast Replication with DB2 utility (DATA ONLY option)	231
C-8	DFSMSHsm LIST command and output on DSN\$DB8A\$DB copy pool	231
C-9	DFSMSHsm LIST command and output on DSN\$DB8A\$LG copy pool.	231
C-10	DB2 backup system full	232
C-11	FRBACKUP output in syslog	232
C-12	FRBACKUP output in DFSMSHsm backup log data set.	233
C-13	Sample JCL of RESTORE SYSTEM utility.	236
C-14	Sample JCL of RESTORE SYSTEM with LOGONLY option	236
C-15	RESTORE SYSTEM utility output	237
C-16	FRRECOVER messages in DFSMSHsm log	238
D-1	When delete authority is removed for objects in DB2 for z/OS	254
D-2	When image copies are deleted for objects in DB2 for z/OS	255
D-3	When objects are dropped in DB2 for z/OS	255
D-4	When quiet time details are removed.	257
D-5	When stogroup or volume changes occur in DB2 for z/OS	257
D-6	Deleting rows in SLR table SRYVRLOG	258
D-7	Deleting rows in ARYAUXRELS SLR table	258
D-8	Deleting rows in ARYDATATYPES, ARYPARMS, ARYROUTINES, ARYVIEWSR SLR tables	258
D-9	Deleting row in ARYFIELDS SLR table	259
D-10	Deleting DBRM, PLAN, and PACKAGES information in SLR tables	259
D-11	Deleting referential constraint information in SLR tables	260
D-12	Deleting specifications in SLR tables	261
E-1	Control statements for DB2 subsystem D8F1	264
E-2	Control statement listing for DB2 subsystem D8F2	267
E-3	Control statement listing for DB2 subsystem D8G1	271
E-4	Control statement listing for DB2 subsystem D8G2	275
E-5	Control statement listing for DB2 subsystem DB8A	279
E-6	Control statement listing for DB2 subsystem DB8W	282
E-7	Control statement listing for DB2 subsystem D9C1	286
E-8	Control statement listing for DB2 subsystem D9C2	290
E-9	JCL to create repository VSAM data sets.	294
E-10	JCL to create mover repository	299
E-11	JCL to create VSAM data set to capture RBA information	299
E-12	Sample proc for the RBA capture utility started task	300
E-13	Sample job to create GDG bases	300
E-14	Sample JCL to create DR objects	301
E-15	ARYV21 CLIST	303
E-16	ARYV210 CLIST	304
E-17	ARYVTSOC CLIST	306
E-18	ARYCAPS CLIST	306
E-19	ARY#PARM parmliib member.	306

Figures

1-1 IBM DB2 Recovery Expert for z/OS Client Server architecture	9
1-2 DB2 Recovery Expert Architecture for z/OS - single LPAR	10
1-3 IBM DB2 Recovery Expert for z/OS architecture - multiple LPARs	11
3-1 Sample Recovery Expert configuration describing GUI components	40
3-2 Snippets of the shared Recovery Expert control file setup job for Agents 53 and 67 ..	42
3-3 Recovery Expert product control file setup job for Agent 59.	43
3-4 Recovery Expert product control file setup job for Agents 63 and 64	44
3-5 Recovery Expert configuration used in the user scenarios for ISPF services.	46
4-1 Copy pool layout	55
4-2 Subsystem setup - user catalogs	57
4-3 Usercat Alias List Display	58
4-4 Subsystem setup - storage copy pools and boot strap data sets	59
4-5 Subsystem setup - active log data sets	60
4-6 Subsystem setup - aliases and volumes in use	61
4-7 Using Subsystem setup to create new ICF user catalogs.	63
4-8 Update User Catalog Information - logs catalog	64
4-9 IDCAMS interface module	65
4-10 Update User Catalog Information - data catalog.	66
4-11 Rename BSDS	67
4-12 Boot Strap Data Set Rename	68
4-13 IDCAMS Interface Module - BSDS	69
4-14 Rename active log data sets.	70
4-15 Active Log Data Set Rename	71
4-16 IDCAMS Interface Module - active logs	72
4-17 Subsystem information after BSDS and log renames.	73
4-18 Moving all data sets on a volume	74
4-19 DB2 data set move	75
4-20 Moving a catalog alias	76
4-21 Enter jobholder information.	76
4-22 Subsystem configuration completed	79
5-1 Recovery launchpad - recovery function.	87
5-2 Recovery selection hierarchy	88
5-3 Recovery selection - DB2 subsystem level.	89
5-4 Point-in-time selection - using explicit function	90
5-5 Time stamp selection for RESTORE SYSTEM.	91
5-6 Recovery plans generation panel	92
5-7 View generated JCL before exporting to a z/OS file	93
5-8 Export window to send JCL to z/OS PDS	93
5-9 Recovery Expert main menu.	101
5-10 Creating a backup and restore profile for a DB2 system level backup and restore .	102
5-11 Updating the backup and restore profile values	103
5-12 Subsystem setup - screen 1 of 4	104
5-13 Subsystem setup - screen 2 of 4	105
5-14 Subsystem setup - screen 3 of 4	105
5-15 Subsystem setup - screen 4 of 4	106
5-16 Creating a backup profile for DB2 D9C1.	107
5-17 Creating a backup profile - enter your values	107
5-18 Backup profile detail window.	108

5-19	Results from saving your newly created profile.	109
5-20	Using backup profile to build JCL	109
5-21	Building backup job - specify JOB card and data set name that will contain JCL. . .	110
5-22	Generated backup JCL - with SETUP	111
5-23	Selecting the system restore option to build RESTORE SYSTEM JCL	114
5-24	High-level view of the system level backup	114
5-25	Viewing the detailed backup system report	115
5-26	Selecting the system level backup for recovery	116
5-27	Selection choice for the recovery	117
5-28	JCL generation window with data set name and members for restore system jobs .	117
5-29	Conditional restart control record create for each DB2 member.	118
5-30	Recovery Expert system level restore JCL	119
5-31	Recovery Expert RESTORE SYSTEM LOGONLY JCL	120
6-1	Objects profile name search	126
6-2	Objects Profile Display window.	127
6-3	Objects profile name	127
6-4	Add Objects to the Object Profile	128
6-5	Enter recovery object (database or table space) name	128
6-6	Enter recovery object (Indexes)	128
6-7	Recovery object (database or table space) name for DB2 9 for z/OS	129
6-8	Tablespace selection window	129
6-9	Object Queue message	130
6-10	Update Object Profile Display	130
6-11	Profile update confirmation	130
6-12	Update Objects Profile Display	132
6-13	Update recovery options through Update Object Profile Display	132
6-14	Build recovery job selection from Objects Profile Display	133
6-15	Build object recovery job.	133
6-16	Object Recovery Options for recover to current	135
6-17	Partial recovery message	136
6-18	Object Recovery Options panel DB2 9 for z/OS	137
6-19	Object Recovery Options for recover to RBA/LRSN	138
6-20	Object recovery options	139
6-21	Object recovery - scenario 1	147
6-22	Object recovery - scenario 2	148
7-1	Dump and restore of DB2 environment	152
7-2	A daily business approach	153
7-3	Primary DR profile detailed definition window.	157
7-4	Secondary DR profile detailed definition window	158
7-5	Disaster recovery job that builds DR JCL	160
7-6	Syntax of the step that builds the DR JCL	161
7-7	The first step of DR job, ssid#JC1, and example of ssidDELC input	165
7-8	The second step of DR job, ssid#JC1, to allocate the DB2 data sets.	166
7-9	The third step of the ssid#JC1 DR job - recover.	167
7-10	Fourth step of ssid#JC1 job - rebuild the BSDS	167
7-11	The fifth step uses REPRO to update the BSDS with the information from the previous step.	168
7-12	The sixth step creates the CRCR record for DB2 to restart properly for recovery . .	168
7-13	The seventh step prints the BSDS information.	169
7-14	The last step of job ssid#JC1 copies the archives logs from tape to DASD	169
7-15	The second DR job, xxxx\$JC1, is optional	170
7-16	DR Job xxx#JC3 recovers DB2 data sets in the proper order	171
8-1	Tape dump	174

8-2	Selecting a backup profile for update	175
8-3	Selecting the offload options to update the backup profile	175
8-4	Offload copy selection.	176
8-5	Offload options specification for the LP and RP copies	177
8-6	Offload options specifications for the local copy.	177
8-7	Updating the output naming convention for the offload	178
8-8	Additional OFFLOAD step as part of system level backup.	179
8-9	Selection options for an offload - R.	180
8-10	Selecting a valid recovery point for an offload - view details.	181
8-11	Example of the view report output	182
8-12	Window to specify JOBCARD information and output data set name for offload JCL	183
8-13	Offload JCL example	183
8-14	Recovery Expert main menu - using option D to create a disaster recovery profile .	185
8-15	Disaster Recovery Profile detailed definition window	186
8-16	Build disaster recovery job window - options for JOBCARD and JCL placement. . .	187
8-17	Building the disaster recovery JCL - actual DR JCL JOBCARD and placement options	187
8-18	First step of batch JOB that will build the disaster recovery JCL	188
8-19	The second step of the batch job that builds disaster recovery JCL.	189
C-1	FlashCopy with background copy	212
C-2	DFSMSdss COPY with COPYVOLID.	215
C-3	DFSMSdss COPY with DUMPCONDITIONING.	216
C-4	Incremental FlashCopy	219
C-5	FlashCopy consistency group	220
C-6	Copy pool structure	223
C-7	BACKUP SYSTEM utility execution for DSNDBOG	229
C-8	Two BACKUP SYSTEM FULL, and one DATA ONLY are taken.	230
C-9	Recovering to an arbitrary point in time using the RESTORE SYSTEM utility.	236
C-10	Panel: DSNTIP6.	240
C-11	BACKUP SYSTEM syntax	241
C-12	RESTORE SYSTEM syntax.	243
C-13	Incremental FlashCopy with single version	244
C-14	Incremental FlashCopy with two versions	244
D-1	Object definition level entries in the SLR	249
D-2	SLR table details for quiet times.	250
D-3	Quiet times	250

Archived

Notices

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about 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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

CICS®	FlashCopy®	Redbooks (logo)  ®
DB2®	IBM®	RETAIN®
DFSMS™	IMS™	TDMF™
DFSMSdfp™	Language Environment®	TotalStorage®
DFSMSdss™	MVS™	z/OS®
DFSMSHsm™	OS/390®	zSeries®
Enterprise Storage Server®	RACF®	
eServer™	Redbooks®	

The following terms are trademarks of other companies:

SAP R/3, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

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

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

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

Preface

IBM® DB2® Recovery Expert manages the recovery of DB2 for z/OS® V8 and V9 objects in a data sharing and non-data sharing environment. The major focus of IBM DB2 Recovery Expert V2.1 for z/OS (5697-N92) is on DB2 system level backup and restore. The tool provides the ability to create a consistent full-system backup with no impact on DB2 availability, comprehensive backup and recovery reporting capabilities, and robust database and storage validity checking to ensure successful creation and usage of database system backups. By integrating with disk storage functions, it supports DB2 subsystem recovery from system backup, DB2 subsystem disaster recovery, and DB2 object data recovery from system backups. It also helps in selecting the most convenient recovery solution and in managing backups.

This IBM Redbooks® publication documents how to use Recovery Expert V2.1 for all restore and recovery functions related to system level backup. For scenarios on using Recovery Expert functions when recovering DB2 objects from traditional image copies, see the previous companion IBM Redbooks publication *DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226.

The team that wrote this book

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

Paolo Bruni is a DB2 Information Management Project Leader at the International Technical Support Organization, based in the Silicon Valley Lab. 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.

Tom Hubbard is a Product Specialists Manager for Rocket Software Inc. based in Houston, Texas. He has 27 years of experience in the information technology field. His areas of expertise include both DB2 and IMS™ backup and recovery planning. He also has extensive experience with system administration and performance management.

Rajasekhar Jonnagadla is a DB2 for z/OS Specialist in ISL (Lab Services) India. He has 10 years of experience in the IT field. He has worked at IBM for one year. His area of expertise is Database Administration of DB2 for z/OS.

Ravikumar Kalyanasundaram is a certified IT Specialist. He works as a Managing Consultant with IBM SWG. Ravi has over 16 years of experience with database technology. He provides DB2 performance consulting services for large customers on z/OS and LUW systems. He holds a Bachelor of Engineering degree in Electrical and Electronics and a master's degree in business administration. Ravi is also a certified Database Administrator on DB2 V7, V8, and V9. His interests are DB2 performance tuning and Disaster Recovery.

Hennie Mynhardt is a Consulting IT Specialist with IBM Software Group. He has lead and worked on different technical projects for database customers in the USA and overseas. His special interests are systems performance tuning and backup/recovery. He currently provides technical consulting and pre-sales and post-sales support for DB2 for z/OS Engine and Tools. Before joining IBM, Hennie worked for an independent software vendor.



The authors from left to right: Tom, Paolo, Hennie, Ravi, and Raj

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

Rich Conway
Bob Haimowitz
Emma Jacobs
Deanna Polm
Sangam Racherla
International Technical Support Organization

Johannes Schuetzner
IBM Germany

Bill Franklin
June Kin
Mary Petras
Haakon Roberts
Judy Ruby-Brown
Dave Schwartz
Jack Shedden
Jim Teng
IBM Silicon Valley Lab

Pierluigi Buratti
IBM Italy

Paul Barthoff
Bob Bersano
Mary Jo Haile
John Pittman
Mark Wallace
Shwan Wikle
Tim Willging
Barry Davis
Rocket Software

Thanks to the authors of the previous book on this topic, *DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226:

Nagraj Alur
Hennie Mynhardt
Gregg Palmer
Alicia Wangelien

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will 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 books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

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

ibm.com/redbooks

- Send your comments in an e-mail to:

redbooks@us.ibm.com

- Mail your comments to:

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

Archived

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7606-00
for Optimizing Restore and Recovery Solutions with DB2 Recovery Expert for z/OS V2.1
as created or updated on February 20, 2009.

June 2008, First Edition

This revision of the First Edition, first published on June 22, 2008, reflect the changes and additions described below.

October 2008, First Update

This revision reflects the addition, deletion, or modification of new and changed information described below.

Changed information

- ▶ Deleted one incorrect row from Table 1-1 on page 12.

February 2009, Second Update

This revision reflects the addition, deletion, or modification of new and changed information described below.

Changed information

- ▶ Corrected Figure 4-1 on page 55.

Archived



Part 1

Architecture and installation

In this part we introduce DB2 Recovery Expert V2.1. We provide an overview of the main features and new functions introduced in V2.1, the differences between the GUI and ISPF interfaces, the process for installing a new release or migrating from a prior release, and the description of our test environment.

This part contains the following chapters:

- ▶ Chapter 1, “Introduction to DB2 Recovery Expert V2.1” on page 3
- ▶ Chapter 2, “Installation” on page 17
- ▶ Chapter 3, “Our test environment” on page 39

Archived

Introduction to DB2 Recovery Expert V2.1

In this chapter we describe the main features of IBM DB2 Recovery Expert for z/OS and its architecture. We also highlight the main new functions made available with DB2 Recovery Expert for z/OS V2.1.

This chapter discusses the following topics:

- ▶ Introduction to DB2 Recovery Expert for z/OS
- ▶ What is new in Recovery Expert V2.1
- ▶ Recovery Expert architecture
- ▶ GUI and ISPF user interfaces

1.1 Introduction to DB2 Recovery Expert for z/OS

The DB2 for z/OS environment can experience different kinds of failures such as application errors, DB2 subsystem failures, Internal Resource Lock Manager (IRLM) failures, disk failures, z/OS failure, power failures, site failures, and so on. When these failures occur, appropriate recovery procedures have to be executed swiftly and precisely to ensure minimal loss of data and minimal loss of system availability. This is where IBM Recovery Expert for z/OS will help.

IBM Recovery Expert for z/OS also supports recovery of the DB2 for z/OS environment accidentally dropped objects. The IBM DB2 Recovery Expert for z/OS tool provides an easy-to-use, automated recovery solution that enables database recovery operations with minimal disruption, and enables you to maintain high availability for database users. Its easy-to-use graphical user interface (GUI) provides powerful reporting and automated recovery capabilities for productive database maintenance and high availability.

Menus make recovery to a specific point in time (PIT) simple and easy to implement. The tool creates recovery options that include rolling changes forward or backward, whichever is the most efficient in a given situation. Not only does the tool provide options for recovery scenarios, it also makes recommendations as to which option is relatively the least expensive in any given situation. This expert functionality saves you time and money by helping you make better decisions.

IBM DB2 Recovery Expert for z/OS also makes versioning easy, because by using the tool you can track object versions and data dependencies. It also supports recovery through the RESTORE SYSTEM utility of DB2 for z/OS Version 8, simplifies complex and laborious operations (such as reversing undesired data changes, including those that have cascaded to related tables), and rebuilding database assets that have been accidentally dropped and therefore do not exist in the DB2 system catalog.

You can also use object profiles developed with the IBM DB2 Automation Tool to recover a set of objects using IBM DB2 Recovery Expert for z/OS. In addition, the tool generates all the necessary job control language (JCL), submits it, and helps you to track progress as recovery proceeds. You can also use the log analysis function to determine *quiet times*, thereby ensuring that the objects that you recover have no activity occurring against them.

IBM DB2 Recovery Expert for z/OS provides a method to discover related sets of tables. This ensures that all relevant objects are recovered together, thereby maintaining integrity of your data. IBM DB2 Recovery Expert for z/OS restores all missing objects related to the objects selected for recovery. By default, Recovery Expert for z/OS always recovers missing objects related through DB2 object dependencies.

The System Backup and Restore Services component of DB2 Recovery Expert provides facilities for using storage-based backup techniques to back up and subsequently recover DB2 data. Data can be processed at the subsystem or table space level of granularity. An ISPF-based interface is provided to assist with DB2 subsystem preparation and management of the backup processes.

DB2 Recovery Expert for z/OS also includes support for recovering objects or entire subsystems at a remote location. You can perform recoveries at a remote site using either traditional DB2 image copies or the equivalent of a system level backup created by DB2 Recovery Expert for z/OS.¹

¹ The functions in *italic* are two new major enhancements with DB2 Recovery Expert V2.1 and are described in detail in this book. For the other functions, see *DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226.

To summarize, the main features of IBM DB2 Recovery Expert for z/OS include:

- ▶ Windows®-based simple, easy-to-use GUI with menus that make recoveries to a point in time current, quick, and precise.
- ▶ ISPF-based interface to assist with DB2 subsystem preparation and management of the backup processes.
- ▶ Multiple options for recovery and recommendations about the least expensive option for a given situation (but leaves the decision to you to choose the most appropriate option for your situation).
- ▶ Schema level repository (SLR) that inspects version levels available for restoration. This includes related dependent objects even if they no longer exist in the DB2 system catalog.
- ▶ Backup and restore services component for creating and managing system level backups.
- ▶ Support for the RESTORE SYSTEM utility of DB2 for z/OS Version 8 and later.
- ▶ Support for restoring individual objects from system level backups in DB2 Version 7 and later.
- ▶ Object profiles developed with any version of IBM DB2 Automation Tool that you can use to recover a set of objects from the GUI.
- ▶ Log analysis function that helps you to determine quiet times to ensure that recovered objects to that time do not have any activity occurring against them.
- ▶ Supports non-data sharing and data sharing environments.
- ▶ Supports DB2 for z/OS Version 7, DB2 for z/OS Version 8 (CM, ENFM, and NFM), and DB2 Version 9 (CM, ENFM, NFM).
- ▶ Support for remote site recovery.

For further information about the functions and constraints, see *DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1*, SC19-1222, and the Web site:

<http://www.ibm.com/software/data/db2imstools/db2tools/db2re-zos/>

1.2 What is new in Recovery Expert V2.1

DB2 Recovery Expert for z/OS V1.1 was made available in July 2006 and its functions are described in *DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226.

DB2 Recovery Expert for z/OS V2.1 (program number 5697-N92) was made available in October 2007 and introduced several new features, which are discussed in this book. The new features include:

- ▶ System Backup and Restore Services
- ▶ Remote site recovery

1.2.1 System Backup and Restore Services

The System Backup and Restore Services component of DB2 Recovery Expert for z/OS supports the creation of backups that are functionally equivalent to system level backups created by the DB2 BACKUP SYSTEM utility. In addition to the backups, *System Backup and Restore Services* add extensive validation checking that makes the backup and restore more thorough and less error-prone. System Backup and Restore Services support DB2 V7, DB2 V8, and DB2 V9, and do not require SMS-managed DB2 subsystems. An inventory of the

backups created by System Backup and Restore Services is maintained by DB2 Recovery Expert for z/OS.

System Backup and Restore Services also offers functionality equivalent to DB2 V8 SYSTEM BACKUP and RESTORE. System Backup and Restore Services can create system level backups using a variety of techniques enabled by specific hardware and software. Supported options are:

- ▶ DB2 BACKUP SYSTEM utility
- ▶ Native FlashCopy® support
- ▶ EMC TimeFinder/Snap
- ▶ EMC TimeFinder/Mirror

The backups created by System Backup and Restore Services can be used to restore a complete subsystem, an individual object, or a group of related objects. The restore of individual objects emulates in DB2 V7 and later the functionality provided by the DB2 RESTORE SYSTEM utility in DB2 9 for z/OS. The restore process can be managed using the DB2 Recovery Expert for z/OS GUI interface or the new ISPF user interface.

System Backup and Restore Services is designed to be simple to use, offering the following features:

- ▶ Employs backup profiles that define and maintain specific settings for backups.
- ▶ Easy setup and management of backup profiles using an ISPF interface.
- ▶ Finds all volumes in use by the DB2 subsystem by entering the VOLUME primary command.
- ▶ Can easily assign a range of target units to source volumes using the TARGET primary command.
- ▶ If log backup is desired, ensures all log volumes are included.
- ▶ Allows optional recovery of the log volumes.
- ▶ When restoring a subsystem, you can see all the valid backups available using the ISPF interface.
- ▶ Automates offloading backups and recalling the backups during restoration.
- ▶ For data sharing subsystems, you can enter a subsystem to restore and a time stamp. System Backup and Restore Services will select the appropriate backup to use.
- ▶ An optional RBA capture utility can track RBAs and their associated time stamps. If you need to restore the subsystem, you can use this information to identify the RBA/LRSN to which to restore.
- ▶ All restore subsystem JCL is generated and contains instructions on what needs to be done to restore the subsystem.
- ▶ Provides both detail and summary reporting with every backup and restore. These reports are also viewable via the ISPF interface.
- ▶ If you specify to enable object recovery when the backup profile is built, you can later restore individual objects from the backup.

Comprehensive validity checking:

- ▶ Ensures that user catalogs are included in the backup, thereby ensuring integrity of the restoration.
- ▶ Optionally checks to ensure that all volumes containing DB2 table space or index space data sets are included in the backup.

- ▶ Optionally, checks to ensure that all volumes containing DB2 active log or BSDS data sets are included in the backup.
- ▶ Makes sure that all source and target volumes are valid.
- ▶ Makes sure that the source volumes are online and available for backup.
- ▶ Ensures that the target volumes for a backup are not in use by another backup profile.
- ▶ Validates that the target volumes have been properly configured and are available for the backup.
- ▶ Validates that the paired source and target volumes are the same device type and reside on appropriate fast replication hardware.
- ▶ Validates that log and object data are on separate volumes. If they are not, a backup is still allowed, but System Backup and Restore Services will issue warnings, and you can only back up and restore the full subsystem (both logs and data).
- ▶ Validates that DB2 log data sets and object data sets are cataloged in separate user catalogs. If they are not, a backup is still allowed, but System Backup and Restore Services will issue warnings, and you can only back up and restore the full subsystem (both logs and data).
- ▶ Ensures that the System Backup and Restore Services component control information resides on a separate device from volumes being backed up. The control information is backed up in its own process.
- ▶ When a volume is used in a restore process, the contents of the volume are validated to insure that there have not been any changes to the backup volume after it was created.

Subsystem Setup facility

System Backup and Restore Services's Subsystem Setup facility collects and displays information about the DB2 subsystem's user catalogs, boot strap data sets, active logs, and related DB2 object data sets. The facility assists you with preparing your DB2 subsystem to use the DB2 BACKUP and RESTORE utilities to create system level backups. The tasks necessary to prepare a DB2 subsystem to use the BACKUP SYSTEM utility instead of or in addition to traditional image copies include:

- ▶ Creation of necessary ICF user catalogs
- ▶ Verifying the existence of correctly named SMS copy pools
- ▶ Moving or renaming DB2 logs and BSDS data sets
- ▶ Moving or renaming the DB2 catalog and directory data sets
- ▶ Moving or renaming the DB2 application data sets

1.2.2 Remote site recovery

Remote site recovery is also known as disaster recovery. Support for remote site recovery has been added to DB2 Recovery Expert for z/OS V2.1. The remote site recovery services support restores and recoveries at a remote site during either system level backups created by DB2 Recovery Expert or standard DB2 image copies.

1.3 Recovery Expert architecture

The main components of IBM DB2 Recovery for z/OS are:

- ▶ Recovery Expert Server

The DB2 Recovery Expert for z/OS server centrally manages and controls all DB2 Recovery Expert for z/OS functions that are performed on behalf of user requests. You

must run at least one instance of the server to manage all of your DB2 subsystems and data sharing groups and to support all of your DB2 Recovery Expert for z/OS user clients. Using TCP/IP connections, the DB2 Recovery Expert for z/OS server, clients, and agents communicate with each other to perform the recovery functions.

- Recovery Expert for z/OS agent

The DB2 Recovery Expert for z/OS agent provides access to database and system services, in support of the DB2 Recovery Expert for z/OS server and remote clients. You must run one instance of the agent on every system or logical partition (LPAR) that hosts DB2 subsystems or data sharing groups that you want to access with DB2 Recovery Expert for z/OS. Each agent communicates with the DB2 Recovery Expert for z/OS server to provide services.

- Recovery Expert for z/OS GUI client

This is a Windows-based simple, easy-to-use GUI with menus that make recoveries to a point in time current, quick, and precise.

- DB2 Recovery Expert for z/OS ISPF Interface

An easy-to-use ISPF interface is provided to assist you with setting the options used when creating and using system level backups managed by DB2 Recovery Expert for z/OS.

- Schema-level repository

IBM DB2 Recovery Expert for z/OS captures DB2 system catalog information and stores it in a set of DB2 tables referred to as the schema level repository (SLR). The schema level repository is an archive to hold object definitions and alterations to object definitions. The sample library member ARYSJ002 contains JCL to update the schema level repository. The initial run of job ARYSJ002 might take multiple hours to copy the contents of the DB2 system catalog. Run times vary depending on the DB2 system catalog size. You must run the schema level repository update job at least daily. If important application object definition updates are performed, then run the schema level repository update after the object definition updates.

A detailed description of the schema level repository is available in Appendix D, "Contents of schema level repository" on page 247.

- System Backup and Restore Services VSAM repository

The System Backup and Restore Services component of DB2 Recovery Expert for z/OS uses a set of VSAM files to store information necessary for tracking information related to system level backups and DB2 subsystem definitions. This repository can optionally be automatically backed-up any time that a system level backup is created by DB2 Recovery Expert for z/OS.

Figure 1-1 shows the client/server nature of the architecture of IBM DB2 Recovery Expert for z/OS. A single instance of IBM DB2 Recovery Expert for z/OS can manage one or more DB2 subsystems located on one or more machines or sysplexes that might be geographically distributed.

Note: If you change the name of the product libraries from the default names, you must substitute the appropriate names that are used in your installation.

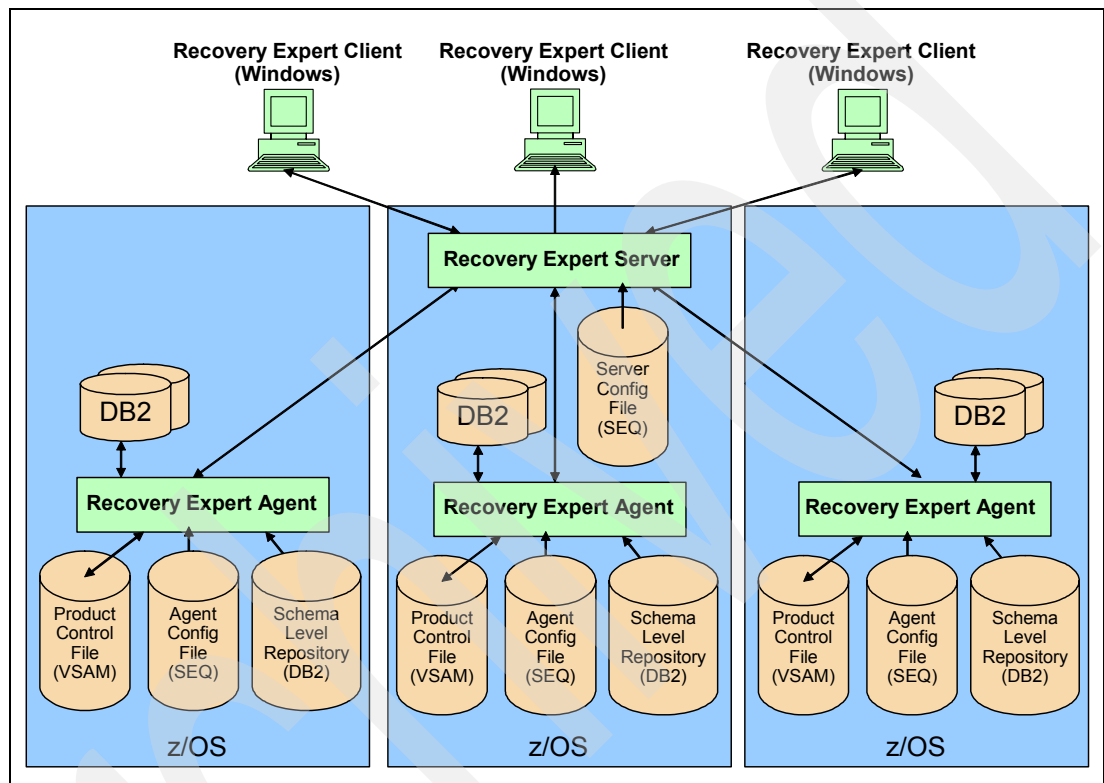


Figure 1-1 IBM DB2 Recovery Expert for z/OS Client Server architecture

Figure 1-2 shows the architecture in a single LPAR implementation. This diagram is included as a base to clarify the components. The new components introduced in DB2 Recovery Expert for z/OS V2.1 are included in this diagram. Those components are:

- ▶ System Backup and Recovery Services (SBRs) CLIST
- ▶ SBRs repository
- ▶ SBRs disaster recovery (DR) objects

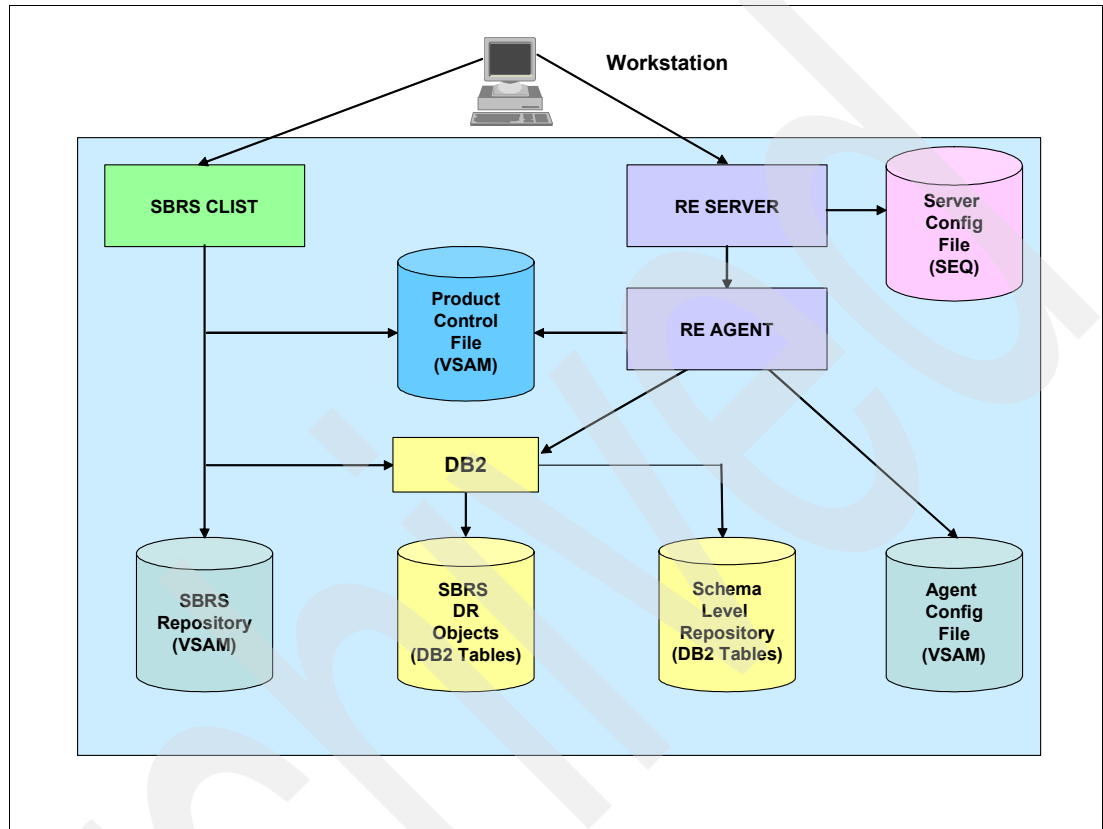


Figure 1-2 DB2 Recovery Expert Architecture for z/OS - single LPAR

The diagram in Figure 1-3 represents the architecture of a multi-LPAR implementation of DB2 Recovery Expert for z/OS. The implementation in Figure 1-3 is the same as previously shown in Figure 1-1 on page 9 with the addition of the components new to Version 2.1. The main new components introduced in Version 2.1 are the ISPF interface and additional repository components.

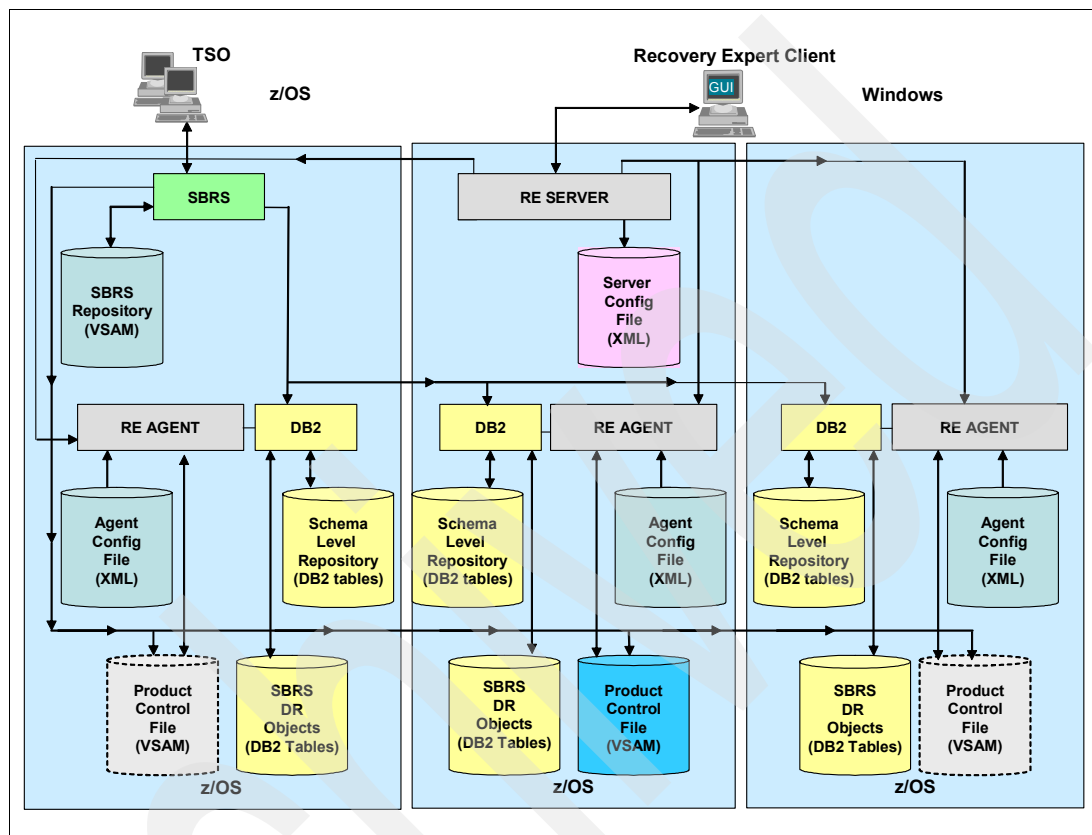


Figure 1-3 IBM DB2 Recovery Expert for z/OS architecture - multiple LPARs

Note: The product control file actually should be shared by all agents and the System Backup and Restore Services (SBRS) component. The entries in the file are common to the agent and SBRS. Therefore, a common file will eliminate the need to try and keep multiple files synchronized. Setup and initialization of the control file are described in detail in “Step 7 - Create the Recovery Expert control file” on page 24, and “Step 8 - Configure the Recovery Expert product control file” on page 24.

1.4 GUI and ISPF user interfaces

DB2 Recovery Expert V1.1 was made available with a Windows-based GUI with menus that make recoveries to a point in time current, quick, and precise. With DB2 Recovery Expert V2.1, a new ISPF interface has been added mostly for managing the System Backup and Restore Services component. This means that, depending on the function needed by the user, there is a choice on the user interface to be used.

In this section we describe in some detail the two interfaces and provide information to help you decide which one is to be used for the task on hand.

The possible configurations at installation time are:

- ▶ GUI only
- ▶ ISPF only
- ▶ GUI and ISPF

Since no one user interface supports all of the functionality provided by DB2 Recovery Expert for z/OS, you should configure both GUI and ISPF interfaces, and use the more appropriate user interface, as shown in Table 1-1.

Table 1-1 Comparison of functions available through the GUI and the ISPF interface

Function	Option	GUI	ISPF
Database/table space object recovery.	To current.	X	X
	To LRSN/RBA or Quiesce point.	X	X
	To Copy (last full, last incremental, specific copy).	X	X
	From Recovery Expert created system level backup.	X ^a	X
	To prior version with DDL.	X	
Table recovery.	To current.	X	
	To LRSN/RBA or Quiesce point.	X	
	To Copy (last full, last incremental, specific copy).	X	
	To prior version with DDL.	X	
Dropped object recovery.	To current.	X	
	To LRSN/RBA or Quiesce point.	X	
	To Copy (last full, last incremental, specific copy).	X	
	To prior version with DDL.	X	
Recover groups of objects.	Based on referential integrity groups.	X	
	Based on DB2 Automation Tool object profiles.	X	
	Based on Recovery Expert object recovery profiles.		X
Recover entire DB2 subsystem.	From Recovery Expert created system level backup.		X
	From DB2 created system level backup.	X ^b	
Identify quiet times in the DB2 log.		X	
Manage DB2 disaster recovery.	Create JCL for jobs necessary to recover a DB2 subsystem at a remote location.		X
Create system level backups.	Generate JCL needed to generate Recovery Expert Managed system level backups.		X

a. Must be using Recovery Expert System Backup and Restore Services - UPDATE IPC IPC_RBR must be set to Y.

b. Must be DB2 V8 and not using Recovery Expert system backup and restore service.

Table 1-1 shows the major functions supported by DB2 Recovery Expert for z/OS and indicates the user interfaces that allow you to use the function.

The GUI user interface was implemented in DB2 Recovery Expert for z/OS V1.1. This interface remains the most function-rich interface available with DB2 Recovery Expert for z/OS. We recommend that both the GUI and ISPF be installed, but in general, the GUI user interface should always be installed and used as the primary user interface since a more limited number of functions require the ISPF user interface.

In order to better explain the contents of Table 1-1 on page 12 we now examine four groups of tools' basic functions and explain which user interface should be used for each function:

- ▶ DB2 object recovery support
- ▶ Recover groups of objects
- ▶ Recovering an entire DB2 subsystem
- ▶ DB2 log analysis

1.4.1 DB2 object recovery support

DB2 Recovery Expert for z/OS supports recovery of individual objects or groups of objects using either the ISPF or the GUI interfaces. Different options are available based on the interface that you are using. These options change based on the type of object being recovered and the interface being used to generate the recovery JCL:

- ▶ Database/table space object recovery
- ▶ Table recovery
- ▶ Dropped object recovery

Object recovery using the GUI

Using the GUI interface, you can recover a single object or a group of objects. An object can be a DB2 database, table space, table space partition, or an individual table. Recovery plans are generated using available recovery resources and available DB2 utilities. Additional recovery options are available using a log analysis component accessed using the GUI interface. Examples of possible recovery plans that the GUI may generate are:

- ▶ Using system level backup (SLB) and RECOVER LOGONLY (cost = 2)
- ▶ Using DSN1COPY and RECOVER LOGONLY (cost = 4)
- ▶ Using DSN1COPY of IC and redo SQL (cost = 4)
- ▶ Using RECOVER (cost = 4)
- ▶ Using RECOVER to IC and redo SQL (cost = 4)

Note: Possible recovery options are presented to the user of the GUI along with an estimated cost of the recovery associated with each plan.

Notes: The user of the GUI can use the recovery option recommended by DB2 Recovery Expert for z/OS or choose any other plan from the available list of recovery plans.

Recovery of individual tables is only available using the GUI. Recovery of dropped objects is only available using the GUI.

The GUI interface introduced in DB2 Recovery Expert for z/OS v1.1 is described in detail in *DB2 Recovery Expert for z/OS User Scenarios, SC24-7226*.

Object recovery using the ISPF user interface

Using the ISPF or the GUI interface, you can also recover a single object or group of objects. An object can be a database, table space, table space partition, or index.

The ISPF interface selects the appropriate recovery based on your input and generates the appropriate JCL to execute the recovery using either image copies or a system level backup for the recovery.

Note: The user of the ISPF interface is not presented with any additional recovery options that may be available based on the recovery assets in the system.

Details of object recovery using the ISPF user interface provided with DB2 Recovery Expert for z/OS are discussed in 6.1, “Object recovery with Recovery Expert under DB2 V8” on page 126.

1.4.2 Recover groups of objects

In this section we discuss recovering groups of objects.

Grouping objects using the GUI

For additional information see the *DB2 Recovery Expert for z/OS User's Guide*, SC19-1222.

Grouping objects using DB2 Automation Tool object profiles

If the DB2 Automation Tool is installed in your installation, the object profiles that have been created can be used by DB2 Recovery Expert for z/OS as a basis for recovery processing. You will have to use the GUI client in order to use DB2 Automation Tool object profiles.

Using DB2 Automation Tool object profiles as a basis for recovery is especially useful if the DB2 Automation Tool is also being used to create image copy backups. This feature then allows you to use the same object profile for the recovery as was used to create the image copy backups.

Grouping objects using the ISPF interface

In this section we discuss grouping objects using the ISPF interface.

Grouping objects using DB2 Recovery Expert for z/OS object recovery profiles

Object recovery profiles can be created using the System Backup and Restore Services component of DB2 Recovery Expert for z/OS. These profiles are used to group objects together for a single recovery. Common recovery options are also specified in the object recovery profile so that the same recovery options are used for all objects included in the object recovery profile.

1.4.3 Recovering an entire DB2 subsystem

In this section we discuss recovering an entire DB2 subsystem.

Subsystem recovery using the GUI

The GUI client can be used to recover an entire DB2 subsystem from a system level backup created by the DB2 BACKUP SYSTEM utility. The GUI client uses the DB2 RESTORE SYSTEM utility for the system restore.

Note: The GUI client does not use non DB2 system level backups created by DB2 Recovery Expert for z/OS.

Subsystem recovery using the ISPF interface

The ISPF client is used for native Recovery Expert functions. It can be used to recover an entire DB2 subsystem using a system level backup created by DB2 Recovery Expert for z/OS. DB2 Recovery Expert for z/OS can be used to invoke the DB2 BACKUP SYSTEM utility.

Note: The ISPF interface does not have visibility to system level backups created outside the control of DB2 Recovery Expert for z/OS. This can include backups created by the DB2 BACKUP SYSTEM utility or backups created using other techniques.

Note: Either the GUI or the ISPF interface can use backups created by the DB2 BACKUP SYSTEM utility when the utility is invoked using DB2 Recovery Expert for z/OS.

1.4.4 DB2 log analysis

DB2 Recovery Expert for z/OS has the ability to analyze activity in the DB2 log for objects to determine times when the objects are not being updated. These quiet points can then be registered in the schema level repository and used as recovery points for the objects.

Archived

Installation

In this chapter we describe the required steps to install Recovery Expert. We describe what to do for a *new installation* of DB2 Recovery Expert for z/OS and for a *migration* of DB2 Recovery Expert for z/OS from V1.1 to V2.1.

This chapter contains the following sections:

- ▶ Installation requirements
- ▶ Installation
- ▶ Migration from Version 1.1

2.1 Installation requirements

When you are ready to install the product, access the IBM Software Support Web site at:

<http://www.ibm.com/software/support/>

Check the PSP Bucket identified by UPGRADE 5697N92 and SUBSET H30R210 and H30RKN0.

For additional service-related information visit:

<http://www.ibm.com/software/data/db2imstools/support.html>

Have the manual *DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1*, SC19-1222, and the *Program Directory for IBM DB2 Recovery Expert for z/OS V2.1*, GI10-8764, on hand.

For the list of APARs/PTFs installed in our environment or recommended, refer to Appendix B, "Recommended maintenance" on page 207.

Hardware and software prerequisites

The hardware and software requirements are:

- ▶ Server and agent components on the mainframe
 - z/OS Version 1 Release 7 or later
 - DB2 - any of the following versions:
 - DB2 UDB for OS/390® and z/OS Version 7 (5675-DB2) and DB2 Utilities Suite for z/OS V7.1 (5697-E98)
 - DB2 for z/OS Version 8 (5625-DB2) and DB2 Utilities Suite for z/OS V8.1 (5655-K61)
 - DB2 Version 9.1 for z/OS (5635-DB2) and DB2 Utilities Suite for z/OS V9.1 (5655-N97)
 - DB2 utilities

The DB2 utilities that are used by DB2 Recovery Expert require execution under TSO/E. Refer to the DB2 Version 7, DB2 Version 8, and DB2 Version 9 product support information for information regarding TSO/E support levels.
 - FlashCopy requirements for backup and restore

Use of FlashCopy for full system backup and restore requires the storage subsystem to be at least FlashCopy V1 capable.
 - Object-level recovery from system level backup requires both:
 - Storage subsystem FlashCopy V2
 - Operating system z/OS V1.8 or later
- ▶ The GUI component
 - Windows

One of the following Windows operating systems:

 - Windows 2000 Professional
 - Windows XP Professional (32 bit)
 - Windows Server® 2003 (32 bit)

- The DB2 Recovery Expert GUI component supports object profiles created with any version of DB2 Automation Tool.
- Hardware requirements
 - For GUI component: any hardware environment that supports the required software
 - For mainframe component: minimum 150 MB free disk space, minimum 128 MB memory, and any hardware environment that supports the required software

Security requirements on the mainframe

In this section we discuss the security requirements on the mainframe.

Server security

The DB2 Recovery Expert Server uses RACF® security. The server requires UNIX® System Services access. The user ID under which the server job runs must have an OMVS segment in its RACF profile.

To check whether the ID has an OMVS segment in its profile, use the following command:

```
LU userid OMVS
```

To add an OMVS segment to an ID's RACF profile, use the following command:

```
ADDUSER ddfluid OMVS(UID(nnn))
```

Agent security

The DB2 Recovery Expert Agent has the following levels of security:

▶ APF authorization

The DB2 Recovery Expert Agent must run APF-authorized. Modify the IEAAPFxx or PROGxx PARMLIB members to define the DB2 Recovery Expert load library ('ARY.IBMTAPE.SARYLOAD') as an APF-authorized library.

▶ BPX.SERVER access

If the BPX.SERVER FACILITY class profile is not defined within RACF, the user ID under which the agent job runs must be assigned a UID=0.

If BPX.SERVER is defined, the user ID must be permitted to it:

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(userid) ACCESS(READ)
```

where *userid* is the user ID under which the agent job runs. If BPX.SERVER is defined, the agent must also run RACF program controlled.

The following RACF commands establish the required definitions for the DB2 Recovery Expert load library and other required load libraries.

For the DB2 Recovery Expert load library:

```
RALTER PROGRAM * ADDMEM('ARY.IBMTAPE.SARYLOAD'//NOPADCHK)
```

For the Language Environment® runtime libraries:

```
RALTER PROGRAM * ADDMEM('xxx.SCEERUN'//NOPADCHK)
RALTER PROGRAM * ADDMEM('xxx.SCEERUN2'//NOPADCHK)
```

where *xxx* is the high-level qualifier of the Language Environment data sets, typically CEE.

For each DB2 subsystem that is accessed by DB2 Recovery Expert:

```
RALTER PROGRAM * ADDMEM('DSN.Vxxx.SDSNEXIT'//NOPADCHK)
RALTER PROGRAM * ADDMEM('DSN.Vxxx.SDSNLOAD'//NOPADCHK)
```

where DSN.Vxxx.SDSNEXIT and DSN.Vxxx.SDSNLOAD name the DB2 load libraries required for accessing that subsystem.

After issuing any of the above commands, the storage program control tables must be refreshed by issuing the following command:

```
SETOPTS REFRESH WHEN(PROGRAM)
```

Note that some of these definitions may already be in place at your site. All of them are required.

► Package and plan access

Each DB2 Recovery Expert user must be granted EXECUTE privileges on the DB2 Recovery Expert packages and plans. DB2 Recovery Expert privileges are granted when you run SARYSAMP member ARYGRT1. See “Step 6 - Grant Recovery Expert privileges” on page 23. You may edit this member to grant appropriate access privileges to user IDs as per your shop standards.

User ID authorities required for installation

This information describes the DB2 Recovery Expert user ID authority required for installation.

If the DB2 DSNZPARM DBACRVW (DBADM CREATE AUTHORITY) is set to YES in the installation panel DNTIPP, then DB2 Recovery Expert can be installed with a user ID that has SYSCTRL authority.

If the DBACRVW (DBADM CREATE AUTHORITY) is set to NO (the default), then the installation user ID must have SYSADM authority to install DB2 Recovery Expert.

- A user ID has SYSADM authority if they are running as the SYSADM user ID.
- A user ID can use DB2 secondary authorization IDs. For more information about DB2 secondary authorization IDs, see the *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840. Additional user ID authorities required for installation are:
 - Server user ID - requires SYSCTRL as the primary auth ID
 - Agent user ID - requires SYSCTRL as the primary auth ID

2.2 Installation

The installation is based on directly customizing and executing the jobs provided in the product library SARYSAMP. In this section you will find the detailed steps required for a complete successful installation of Recovery Expert V2.1.

The following three types of new installation are possible with DB2 Recovery Expert for z/OS V2.1:

► Client/server (GUI) component only

With the GUI interface you can perform:

- Dropped object recovery
- Log analysis through which you can determine quiet points
- Relative cost evaluation for the generated recovery plans
- Recovery using object profiles created using DB2 Automation Tool

► ISPF component only

With the ISPF component only you can perform:

- System level backup and restore
- Object restore using object profiles
- Disaster recovery

► Complete installation

With both GUI and ISPF components you can perform all functions.

In this section we describe the steps for a complete install. Table 2-1 lists the sequence of steps for a new installation. We mention the tasks performed, the members of the SARYSAMP library that perform it, and indicate whether the step needs repeating in case of multiple DB2 subsystems to control.

All the required JCL members are supplied in the SARYSAMP library.

Table 2-1 Steps for installing Recovery Expert V2.1 - complete install

Step and description ^a	Comment	SARYSAMP members	Multiple system install
Step 1 - APF authorize the DB2 Recovery Expert LOAD library data set.		-	
Step 2 - Copy and edit the ARYEMAC1 CLIST.		ARYEMAC1	X
Step 3 - Create Recovery Expert DB2 objects.		ARYDDLA ARYDDL7, ARYDDL71, ARYDDL72 or ARYDDL8, ARYDDL81, ARYDDL82 or ARYDDL9, ARYDDL91, ARYDDL92 ARYDDL0 (optional) ARYDDLG (optional)	X
Step 4 - Create indexes on the DB2 catalog.	Optional	ARYDSCIX	X
Step 5 - Bind Recovery Expert DB2 packages and plans.		ARYBND71 ARYBND72 ARYBND73 ARYBND74 ARYBND81 ARYBND82 ARYBND83 ARYBND84 ARYBND91 ARYBND92 ARYBND93 ARYBND94	X
Step 6 - Grant Recovery Expert privileges.		ARYGRT1	X
Step 7 - Create the Recovery Expert control file.		ARYSJ000	Optional
Step 8 - Configure the Recovery Expert product control file.		ARYSJ001	
Step 9 - Update the schema level repository.		ARYSJ002	X
Step 10 - Install the GUI client.		-	
Step 11 - Configure the server and agent.		ARYCFGs, ARYSJSRV, ARYSPSRV, ARYCFGa, ARYSJAGT, ARYSPAGT	X ^b
Step 12 - Verify System Backup and Restore Services parameter settings ^c .		-	

Step 13 - Create the System Backup and Restore Services VSAM repository		ARYREPO	
Step 14 - Update the System Backup and Restore Services CLISTs.		ARYV21, ARYV210	
Step 15 - Customize the CLIST members for SBRS.		ARYV21, ARYV210, ARYTSOC, ARYCAPS	
Step 16 - Set up the job for repository backup data sets.		ARYGDG	
Step 17 - Change defaults for backup restore.	Optional	ARY#PARM	
Step 18 - Update the authorized programs list.		SYS1.PARMLIB	
Step 19 - Configure RACF profiles for SBRS.	Optional	-	
Step 20 - Set the default value for backup profiles.	Optional	-	
Step 21 - Configure the RBA capture utility.	Optional	ARY#RBAR, ARY#RBA	X
Step 22 - Create the SBRS subsystem setup repository.	Optional	ARYMOVER	
Step 23 - Activate Recovery Expert started tasks.			

a. Click the step to go to the corresponding paragraph.

b. If the agent config file is on shared disk, then multiple agents are not required. If it is not, then you need to define multiple agents.

c. DB2 9 for z/OS and later only.

Note: If you are configuring DB2 Recovery Expert for z/OS to work with more than one DB2 subsystem then you have to repeat the steps that are checked under the Multiple system install column in Table 2-1.

Step 1 - APF authorize the DB2 Recovery Expert LOAD library data set

IBM DB2 Recovery Expert for z/OS V2.1 requires that the product LOAD library is APF authorized. You may need to APF authorize libraries for each LPAR involved. Refer to *z/OS V1R9.0 MVS System Commands, SA22-7627-16*, for more information about how to use SETPROG to APF authorize libraries.

Step 2 - Copy and edit the ARYEMAC1 CLIST

The edit macro "ARYEMAC1" can be used to edit and customize the JCL and DDL members in SARYSAMP to suit your environment during the installation process. The variables in this macro can be specified with values for each DB2 subsystem or each member of a data sharing group. For example, the #SSID variable in the macro should be assigned a value representing the DB2 subsystem for which DB2 Recovery Expert for z/OS is configured. Make necessary changes to this member after copying this macro to the SYSPROC library. If you plan to perform the configuration in more than one DB2 subsystem, then copy this macro multiple times with unique names in the SYSPROC library (that is, the appropriate user CLIST library).

In data sharing, an edit macro would need to be created for each member of the data sharing group.

Note for the following steps: When creating the Recovery Expert objects with the help of DDL members supplied, you have a choice to create a PDS for each subsystem or to keep all the DDLs in one PDS with different names. Depending on your naming conventions, you can make the selection. Some jobs, like creating the ISPF VSAM Repository data sets, are common to multiple subsystems in an LPAR (that is, they need only to be run once in an LPAR). If you can place them on a shared disk device across all the LPARs then you need to create the data sets once.

Step 3 - Create Recovery Expert DB2 objects

Edit and run the DDL in member ARYDDLA. This creates the SYSTOOLS database where the DB2 Recovery Expert for z/OS objects are created and used for all members of a data sharing group.

Based on the version of DB2 that you have, edit and run ARYDDL7 through ARYDDL72, ARYDDL8 through ARYDDL82, or ARYDDL9 through ARYDDL92. For details see the comments section in each member.

If you have an automation tool installed then edit and run the JCL in member ARYDDL0.

If you have grouper installed then edit and run the JCL in member in ARYDDLG.

Step 4 - Create indexes on the DB2 catalog

Optionally, run this step to improve product performance. This step creates additional indexes on DB2 system catalog tables SYSSYNONYMS, SYSTABLES, SYSVIEWDEP, and SYSRELS.

Step 5 - Bind Recovery Expert DB2 packages and plans

Based on the version of DB2, edit and run ARYBND71, ARYBND81, or ARYBND91 for DB2 Recovery Expert packages.

Based on the version of DB2, edit and run ARYBND72, ARYBND82, or ARYBND92 for DB2 Recovery Expert plans.

Based on the version of DB2, edit and run ARYBND73, ARYBND83, or ARYBND93 for DB2 grouper support.

Based on the version of DB2, edit and run ARYBND74, ARYBND84, or ARYBND94 for DB2 Automation Tool Support.

Note: Further details on what needs to be changed on each of these sample library members is clearly documented in the comments section of the corresponding members.

Step 6 - Grant Recovery Expert privileges

Edit and run the member ARYGRT1 to grant privileges required to access DB2 Recovery Expert. Read the instructions in member ARYGRT1 to edit the member to suit your environment. The default DCL code in this member grants all the required access to PUBLIC. You should edit this member to allow appropriate access to your users.

Note: Change the GRANT statements from PUBLIC access to a specific user ID as per your shop requirements.

Step 7 - Create the Recovery Expert control file

The member ARYSJ000 has the IDCAMS definitions to create the product control file. The product control file is a VSAM KSDS data set. This VSAM data set is used in the following members: ARYSJ001 through ARYSJ004, ARYCFGA, ARYSJAGT, ARYSPAGT, ARYV210, ARYMIG91, and ARYMIG94.

You may create just one product control file to serve the entire system by placing it on a shared disk. If you prefer to have different product control files for test and development environments then you may create two VSAM files by running ARYSJ001 twice with different data set names. You must update the agent started task procedure to reflect the corresponding product control file name.

The product control file is used by both GUI and ISPF components.

Step 8 - Configure the Recovery Expert product control file

The member ARYSJ001 updates the product control file with DB2 Recovery Expert configuration information. This file should include control statements for all DB2 subsystems. You should copy the SYSIN card in this member for each of the DB2 subsystems that you are trying to manage using this product control file. You should also make sure to add an entry with the group attach name of each data sharing group.

You may have to pay attention to the control statements in the SYSIN card of ARYSJ001 member discussed below.

if you want the GUI to generate recovery plans to recover objects from system level backups that were created in the System Backup and Restore Services (ISPF), often abbreviated as SBRS, then you must have the following control statement in your SYSIN card:

```
UPDATE IPC IPC_RBR = Y
```

Set the RBR DEVICE control statement to a valid disk work device (that is, SYSDA, SYSALLDA, and so on, to match your shop standard). For instance, you may set it to SYSALLDA using the following control statement in your SYSIN card:

```
UPDATE RBR DEVICE = SYSALLDA
```

If you want to configure DB2 Recovery Expert for z/OS to create system level backups using EMC-based arrays for fast replication then you should set the following control statements with appropriate load library names for EMC TimeFinder, EMC Snap, and the EMC SCF load libraries (in any order):

```
UPDATE RBR EMC LOAD1 = YOUR.EMC.LOADLIB1
UPDATE RBR EMC LOAD2 = YOUR.EMC.LOADLIB2
UPDATE RBR EMC LOAD3 = YOUR.EMC.LOADLIB3
UPDATE RBR EMC LOAD4 = YOUR.EMC.LOADLIB4
```

If you wish to have Recovery Expert offload your system backups using the Innovations FDR product, you need to set the following control statements to the load library names that contain the FDR product:

```
UPDATE RBR FDR LOAD1 =YOUR.FDR.LOADLIB1
UPDATE RBR FDR LOAD2 =YOUR.EMC.LOADLIB2
```


Important: You must match the data set names used in ARYV210 CLIST with the corresponding ones in ARYSJ001, as shown in Table 2-2.

Table 2-2 Mapping of data set names used in ARYSJ001 to DD name in ARYV210 CLISTs

ARYSJ001 (GUI component)	ARYV210 (ISPF component)
UPDATE RBR PROF REPO = ARY.PROFILES	RBRBPROF(ARY.PROFILES)
UPDATE RBR MAPS REPO = ARY.PROFILE.MAPS	RBRBPMAP(ARY.PROFILE.MAPS)
UPDATE RBR CATS REPO = ARY.PROFILE.CATS	RBRBPCAT(ARY.PROFILE.CATS)
UPDATE RBR SYSBK REPO = ARY.SYSBACK	RBRBACK(ARY.SYSBACK)
UPDATE RBR BKUP VOLS = ARY.SYSBACK.VOLS	RBRBVOL(ARY.SYSBACK.VOLS)
UPDATE RBR BKUP SSID = ARY.SYSBACK.SSIDS	RBRBSSD(ARY.SYSBACK.SSIDS)
UPDATE RBR BKUP OBJS = ARY.SYSBACK.OBJS	RBRBBOBJ(ARY.SYSBACK.OBJS)
UPDATE RBR REPORT REPO = ARY.BREPORT	RBRBREPT(ARY.BREPORT)
UPDATE RBR OFFLD REPO = ARY.OFFOPTS	RBRBOFFL(ARY.OFFOPTS)
UPDATE RBR PARMLIB = ARY.V2R1M0.SARYSAMP	PARMLDSN(ARY.V2R1M0.SARYSAMP)
UPDATE RBR PARMLIB MBR= ARY#PARM	PARMLMBR(ARY#PARM)
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU	RBRLVL(ARY.V2R1M0)
	MLIB(SARYMENU)

Example 2-1 shows a sample input for one DB2 V8 subsystem to the SYSIN card for ARYSJ001.

Example 2-1 Sample SYSIN card for ARYSJ001

```
//SYSIN DD *
SET DB2 SSID = DB8W
UPDATE DB2 ZPARMS = DSNZPARM
UPDATE DB2 BOOTSTRAP1 = DB8WU.DB2.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB8WU.DB2.BSDS02
UPDATE DB2 LOADLIB1 = DB8W8.SDSNEXIT
UPDATE DB2 LOADLIB2 = DB8W8.SDSNLOAD
SET PRODUCT CFG = NULL
SET PRODUCT VER = NULL
UPDATE ARY PLAN1 = ARYPLAN1 DISPLAY DATA EXTRACT
UPDATE ARY PLAN2 = ARYPLAN2 SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3 = ARYPLAN3 RECOVERY PLAN GENERATION
UPDATE ARY PLAN4 = ARYPLAN4 JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5 = ARYPLAN5 LOG ANALYSIS SERVICES
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU
UPDATE ARY ARCHLOG1 = Y USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2 = N USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI = Y ACTIVE LOG PRIORITY
UPDATE ARY DSN PREFIX = SG7606.&USERID
UPDATE ARY RCVR AUTHS = Y RECOVER DB2 OBJECT AUTHORIZATIONS
UPDATE SLR LOAD AUTHS = Y N = DB2 authorizations are
UPDATE IPC IPC_GROUPE = N Y = Enable Grouper-related
```

```

UPDATE QT GRP TOWNER = SYSTOOLS
UPDATE QT GRP TNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
UPDATE LAS DSN PREFIX = SG7606
UPDATE LAS VOLSERS = SBOX49,SBOX3Q,SBOX3R
UPDATE LAS DATA AUNIT = C
UPDATE LAS DATA PQT = 00005
UPDATE LAS DATA SQT = 00005
UPDATE LAS INDEX AUNIT = C
UPDATE LAS INDEX PQT = 00005
UPDATE LAS INDEX SQT = 00005
UPDATE CCS SLR TECHN = ER
UPDATE CCS SLR SBCS = 00037
UPDATE CCS SLR DBCS = 01200
UPDATE CCS SLR MIXED = 01208
UPDATE CCS ARY TECHN = ER
UPDATE CCS ARY SBCS = 00037
UPDATE CCS ARY DBCS = 01200
UPDATE CCS ARY MIXED = 01208
UPDATE FTW DEVICE = SYSALLDA
UPDATE FTW ALCUNIT = C
UPDATE FTW PQT = 00001
UPDATE FTW SQT = 00001
UPDATE ICF DEVICE = SYSALLDA
UPDATE ICF ALCUNIT = C
UPDATE ICF PQT = 00001
UPDATE ICF SQT = 00001
UPDATE RDA DEVICE = SYSALLDA
UPDATE RDA ALCUNIT = C
UPDATE RDA PQT = 00001
UPDATE RDA SQT = 00001
UPDATE RBR PROF REPO = ARY.PROFILES
UPDATE RBR MAPS REPO = ARY.PROFILE.MAPS
UPDATE RBR CATS REPO = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO = ARY.SYSBACK
UPDATE RBR BKUP VOLS = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID = ARY.SYSBACK.SSIDS
UPDATE RBR BKUP OBJS = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO = ARY.BREPORT
UPDATE RBR OFFLD REPO = ARY.BREPORT
UPDATE RBR PARMLIB = ARY.V2R1M0.SARYSAMP
UPDATE RBR PARMLIB MBR = ARY#PARM
/*

```

CHARACTER CONVERSION TECHNIQUE
EBCDIC
UNICODE UT-16
UNICODE UT-8
CHARACTER CONVERSION TECHNIQUE
EBCDIC
UNICODE UT-16
UNICODE UT-8
DEVICE TYPE
C=CYLS, T=TRACKS
PRIMARY QTY
SECONDARY QTY
DEVICE TYPE
C=CYLS, T=TRACKS
PRIMARY QTY
SECONDARY QTY
DEVICE TYPE
C=CYLS, T=TRACKS
PRIMARY QTY
SECONDARY QTY

You do not need multiple control data sets for multiple DB2 subsystems. All the information that Recovery Expert needs for each DB2 subsystem is described within the setup panel. After saving your input, the corresponding DB2PARMS control file is updated. Only one DB2PARMS control file is necessary for managing multiple DB2 subsystems on a single LPAR. If you are monitoring members of a data sharing group on a single LPAR with one Recovery Expert started task, then identify all of these members within the same DB2PARMS control file. You can share the same DB2PARMS control file among multiple subsystems.

The only reasons for having multiple DB2PARMS control files is if you have multiple subsystems on different LPARS without shared disk or if you want to support multiple versions of the products in the same DB2 subsystem.

Define the proper RACF controls on the DB2PARMS product control file to allow read and update access to appropriate IDs. In this way, the DB2PARMS control file has proper security to prevent access by unauthorized users.

Step 9 - Update the schema level repository

Edit and run the member ARYSJ002 to update the schema level repository with current information from the DB2 system catalog. Read the instructions completely in the commented section of this member. Contents of the SLR tables are described in Appendix D, "Contents of schema level repository" on page 247.

Step 10 - Install the GUI client

Follow these steps to install the GUI client. The GUI client installation executable is a member of a data set from the z/OS installation.

1. Locate the ARY.IBMTAPE.SARYGUIW(ARYGUIW) data set member. In our environment it is ARY.V2R1M0.SARYGUIW(ARYGUIW).
2. Use FTP to transfer member ARYGUIW (in binary) to your workstation, renaming the member with a local file name of setupwin32.exe.
3. Run setupwin32.exe.

Installation information:

- ▶ During installation, all of the DB2 Recovery Expert client components are copied to the location specified in the installation program. The default location is C:\Program Files\IBM\DB2TOOLS\DB2RecoveryExpert. Follow the instructions in the installation program if you want to change the location of the files. It is possible to install and run Recovery Expert from a network drive if you prefer.
- ▶ No registry keys are created during the installation, but application preferences are saved under HKEY_CURRENT_USER\Software\JavaSoft\Prefs\com\rocketsoftware\ary when you run DB2 Recovery Expert.
- ▶ The installation does not create desktop icons, but it does create Start menu items that you use to start DB2 Recovery Expert.

Step 11 - Configure the server and agent

Edit the following members:

- ▶ ARYCFGA

Agent configuration file. The default values for all parameters may be used, except for <server-address> and <server-port>, which must be configured properly to connect to the DB2 Recovery Expert Server

Example 2-2 shows a sample configuration file for the IBM DB2 Recovery Expert for z/OS 2.1.0 Agent.

Example 2-2 Sample configuration file for the IBM DB2 Recovery Expert for z/OS 2.1.0 Agent

```
<agent-config>
  <server-address>wtsc59.itso.ibm.com</server-address>
  <server-port>9876</server-port>
  <community-string>wtsc53</community-string>
  <multicast-address>236.1.2.3</multicast-address>
  <multicast-port>19875</multicast-port>
```

```

<control-file-dd>DB2PARMS</control-file-dd>
<log-level>I</log-level>
<server-connect-retry-max>10</server-connect-retry-max>
<server-connect-retry-delay>10</server-connect-retry-delay>
<request-thread-timeout>300</request-thread-timeout>
<uppercase-passwords>true</uppercase-passwords>
<job-poll-rate>5</job-poll-rate>
<job-cancel-timeout>5</job-cancel-timeout>
<check-ownership-external>true</check-ownership-external>
<topology-discovery-restriction>configured</topology-discovery-restriction>
<trace-csi>false</trace-csi>
<trace-db2-attachment>false</trace-db2-attachment>
<trace-sql>false</trace-sql>
<trace-ifi>false</trace-ifi>
<trace-network>false</trace-network>
<trace-xml>false</trace-xml>
<trace-events>false</trace-events>
<trace-config>true</trace-config>
</agent-config>

```

► ARYCFG

Server configuration file. The default values for all parameters may be used.

Example 2-3 shows a sample configuration file for the IBM DB2 Recovery Expert for z/OS 2.1.0 server.

Example 2-3 Sample configuration file for the IBM DB2 Recovery Expert for z/OS 2.1.0 server

```

<server-config>
  <client-listener-port>9875</client-listener-port>
  <agent-listener-port>9876</agent-listener-port>
  <description></description>
  <community-string>wtsc53</community-string>
  <multicast-address>236.1.2.3</multicast-address>
  <multicast-port>19875</multicast-port>
  <multicast-interface></multicast-interface>
  <multicast-ttl>5</multicast-ttl>
  <multicast-delay>5</multicast-delay>
  <log-level>I</log-level>
  <bind-retry-max>30</bind-retry-max>
  <bind-retry-delay>10</bind-retry-delay>
  <trace-network>false</trace-network>
  <trace-xml>false</trace-xml>
  <trace-events>false</trace-events>
  <trace-config>true</trace-config>
</server-config>

```

► ARYSJSRV

Sample batch JCL to invoke the server. Customize this JCL to your environment and make use of it when you need to invoke the server in batch mode.

► ARYSJAGT

Sample batch JCL to invoke the agent. Customize this JCL to your environment and make use of it when you need to invoke the agent in batch mode.

► ARYSPSRV

The procedure provided in this member can be used to run DB2 Recovery Expert Server as a started task.

► ARYSPAGT

The procedure provided in this member can be used to run DB2 Recovery Expert Agent as a started task.

Step 12 - Verify System Backup and Restore Services parameter settings

This step is applicable to DB2 9 for z/OS and later. The following four DB2 parameters must be updated with the values shown:

```
SYSTEM_LEVEL_BACKUPS = NO
```

The DB2 RECOVER utility should not use system level backups as a recovery base for object-level recoveries:

```
RESTORE_RECOVER_FROMDUMP = NO
```

Use the disk copy of system level backup for the RESTORE SYSTEM and the RECOVER utilities:

```
UTILS_DUMP_CLASS_NAME = <blank>  
RESTORE_TAPEUNITS = NOLIMIT
```

Note that the above values are the default in DB2 9.

Even though the DSNZPARM restricts the DB2 RECOVERY utility from using SLB as a recovery base for object-level restore, DB2 Recovery Expert System Backup and Restore Services can perform object-level restore using the SLB taken from Recovery Expert.

Step 13 - Create the System Backup and Restore Services VSAM repository

Edit the JCL in member ARYREPO (as per the instructions provided in the member) to create VSAM data sets that would hold backup and restore information. The following data sets are created when you run the JCL in this member:

```
ARY.PROFILES  
ARY.PROFILE.MAPS  
ARY.PROFILE.CATS  
ARY.SYSBACK  
ARY.SYSBACK.VOLS  
ARY.SYSBACK.SSIDS  
ARY.SYSBACK.OBJS  
ARY.BREPORT  
ARY.OFFOPTS  
ARY.OBJECTS
```

Step 14 - Update the System Backup and Restore Services CLISTS

Edit the CLIST member ARYV210. This CLIST is used by ARYV21 to execute the product and is used by the ISPF component only. Note that some of the file names used in this member are being created in other steps (like ARYSJ000, ARYREPO, ARYMOVER, ARY#RBAR, and ARYGDG). You must change the default names used in this member as outlined in the initial comments section of the member. Table 2-3 lists the key data sets used in this member.

Table 2-3 ARYV210 CLIST

ARYV210 (ISPF component)	Description
DB2CNTFL(ARY.DB2PARMS.CONTROL)	Product control file name.
RBRBPROF(ARY.PROFILES)	VSAM data set that contains backup profiles that are created via ISPF interface.
RBRBPMAP(ARY.PROFILE.MAPS)	VSAM data set that contains volume mappings entered in each backup profile.
RBRBPCAT(ARY.PROFILE.CATS)	VSAM data set that contains catalogs for each volume entered in a backup profile.
RBRBACK(ARY.SYSBACK)	VSAM data set that contains system backup information.
RBRBVOL(ARY.SYSBACK.VOLS)	VSAM data set that contains system backup volume information.
RBRBSSD(ARY.SYSBACK.SSID)	VSAM data set that contains system backup SSID information.
RBRBBOBJ(ARY.SYSBACK.OBJS)	VSAM data set that contains one record for all the DB2 object data sets at the time of backup.
RBRBREPT(ARY.BREPORT)	VSAM data set that contains the backup report.
RBRBOFFL(ARY.OFFOPTS)	VSAM data set that contains the offload options for a profile.
RBRPOBJS(ARY.OBJECTS)	VSAM data set that contains objects in an object profile.
RBRMOVER((ARY.SG7606.MOVDATA)	VSAM data set that contains catalog, alias, and volume information of a DB2 subsystem.
RBRSSRBA(SG7606.SC53.ALLSSIDS.RBA)	VSAM data set that contains the collected RBA to time stamp correlation values.
RBRBUPRE(ARY)	Repository files are backed up into a GDG data set, whose prefix is defined here.
RBRBUSUF(ARYGDGSF(+1))	Repository files are backed up into a GDG data set, whose suffix is defined here.
PARMLDSN(ARY.V2R1M0.SARYSAMP)	PDS data set that contains the member specified in PARMLMBR.
PARMLMBR(ARY#PARM)	Member in the PDS data set specified in PARMLDSN. It contains various parameters that affect the functioning of utilities.
RBRLOAD1(ARY.V2R1M0.SARYLOAD)	Recovery Expert Loadlib.
RBRLVL(ARY.V2R1M0)	High-level prefix for ARY load libraries.
MLIB(SARYMENU)	Suffix of the message library.
PLIB(SARYPENU)	Suffix of the panel library.
SLIB(SARYSLIB)	Suffix of the skeleton library.
LLIB(SARYLOAD)	Suffix of the load library.

Important: You must match the VSAM control file names used in the ARYV210 member with the corresponding ones in ARYSJ001, as shown in Table 2-2 on page 25.

Step 15 - Customize the CLIST members for SBRS

Edit and make the necessary changes to the following members and then move them to the SYSPROC library:

```
ARYV21
ARYV210
ARYTSOC
ARYCAPS
```

Step 16 - Set up the job for repository backup data sets

Edit and run the member ARYGDG to create GDG bases that would be used to optionally back up the repository files. The prefix and suffix of the GDG data sets are referenced in “Step 14 - Update the System Backup and Restore Services CLISTs” on page 30.

Step 17 - Change defaults for backup restore

This step is optional, but recommended. The ARY#PARM member provides settings that control various aspects of the backup and restore utilities. If you require changing a parameter from the defaults provided, then update this member with the appropriate settings. The member ARY#PARM and the PDS data set that contains this member are referenced in “Step 14 - Update the System Backup and Restore Services CLISTs” on page 30.

You might want to customize parameters that control functions such as the resetting of COPYP status for table spaces and indexspaces via the options RESET_COPY_PENDING_TS and RESET_COPY_PENDING_IX, detailed in Example E-19 on page 306.

Step 18 - Update the authorized programs list

Add ARY\$TSOC to the list of authorized programs in SYS1.PARMLIB(IKTJT000):

```
AUTHPGM NAMES(          /* AUTHORIZED PROGRAMS      */      +
  ARY$TSOC              /* DB2 RECOVERY EXPERT      */      +
  ...)
```

Step 19 - Configure RACF profiles for SBRS

We recommend regulating user access to System Backup and Restore Services functions using security procedures implemented in your environment.

You can optionally restrict use of certain System Backup and Restore Services functions by using RACF profiles. To regulate user access to System Backup and Restore Services functions, create RACF security profiles. To secure System Backup and Restore Services's functions, you can create the profile shown in Table 2-4.

Table 2-4 RACF profile

Function	Description of authority	Profile	Access
Access	Grants a user the authority to use System Backup and Restore Services	ARY.ACCESS.ssid, where ssid is a four-character DB2 subsystem name or ‘*’	READ

A user is not allowed to execute a System Backup and Restore Services backup or restore utility if he is not granted READ access to the corresponding profile or if the profile does not exist. If the specific RACF Facility Class Profile does not exist, then the most granular generic RACF Facility Class Profile will be applied in its place. For example, if ARY.ACCESS.ssid does not exist for a given System Backup and Restore Services subsystem, but a generic Facility Class Profile name ARY.ACCESS.* exists, then the generic profile will be used. Only authorization IDs with READ access to the profile will be cleared by RACF.

Step 20 - Set the default value for backup profiles

Optionally, set the default values for creating backup profiles by invoking the DB2 Recovery Expert through the ISPF interface. Select option **S (Product setup)** and follow the panels that open.

Step 21 - Configure the RBA capture utility

Optionally, edit and run ARY#RBA to create the VSAM data set used by the RBA capture utility. This data set can be used to capture RBA information for all the DB2 subsystems involved in your installation.

Then update the member ARY#RBA with the newly created VSAM data set name and specify the time interval to capture the RBA information. Copy this member to SYS1.PROCLIB, and you may start this started task.

The VSAM data set created in this step is referenced by the ARYV210 CLIST in “Step 14 - Update the System Backup and Restore Services CLISTs” on page 30.

Step 22 - Create the SBRs subsystem setup repository

Optionally, edit and run the member ARYMOVER to create the VSAM data set to hold DB2 subsystem information. The VSAM data set created in this step is referenced by the ARYV210 CLIST in “Step 14 - Update the System Backup and Restore Services CLISTs” on page 30.

Step 23 - Activate Recovery Expert started tasks

Issue the start command to start the DB2 Recovery Expert Server started task and the agent started tasks. This step is required for GUI.

Important: Start the DB2 Recovery Expert for z/OS server before starting any of the associated agents.

2.3 Migration from Version 1.1

In this section we list the steps for a successful migration of DB2 Recovery Expert for z/OS from DB2 Recovery Expert V1.1 to 2.1, which has new features via ISPF interface.

When you migrate from DB2 Recovery Expert V1.1 to V2.1, you need to retain the DB2 tables of SLR. The authorization and privileges on the DB2 objects of SLR must be preserved. Appropriate privileges must be granted to the new DB2 tables that correspond to the ISPF interface of DB2 Recovery Expert. Also, note that the backup control statements and recovery control statements are new to the product control file when migrated from V1.1 to V2.1. So, before updating the product control file by running the job ARYSJ001, you need to stop the DB2 Recovery Expert Agent.

Table 2-5 lists the steps necessary to migrate.

Table 2-5 Steps for migration from Recovery Expert V1.1 to V2.1

Step and description	Comment	SAMPLIB member	Multiple system Install
Step 1 - Copy the ARYEMAC1 CLIST.		ARYEMAC1	X
Step 2 - Create DB2 Recovery Expert objects for V2.1.		ARYDDLRL	X
Step 3 - Create image copies in the schema level repository.		ARYCPY1	X
Step 4 - Bind Recovery Expert DB2 packages and plans.		ARYBND71 ARYBND72 ARYBND73 ARYBND74 ARYBND81 ARYBND82 ARYBND83 ARYBND84 ARYBND91 ARYBND92 ARYBND93 ARYBND94	X
Step 5 - Bind Recovery Expert DB2 packages and plans.		ARYGRT1	X
Step 6 - Stop the DB2 Recovery Expert Agent and server tasks.		-	X
Step 7 - Configure the Recovery Expert control file.		ARYSJ001	-
Step 8 - Update the schema level repository.		ARYSJ002	X
Step 9 - Install the GUI client.		-	-
Step 10 - Configure the server and agent.		ARYCFGS ARYSJSRV ARYSPSRV ARYCFGA ARYSJAGT ARYSPAGT	X
Step 11 - Update system parameters.		For DB2 9 for z/OS and later only	X
Step 12 - Create the SBRS VSAM repository.		ARYREPO	-
Step 13 - Update the SBRS CLIST.		ARYV21, ARYV210	-
Step 14 - Copy members to the CLIST library for SBRS.		ARYV21, ARYV210, ARYTSOC, ARYCAPS	-
Step 15 - Create the GDG base for the VSAM repository.		ARYGDG	-
Step 16 - Configure the SBRS PARMLIB member.	Optional	ARY#PARM	-

Step and description	Comment	SAMPLIB member	Multiple system Install
Step 17 - Configure the RBA capture utility.	Optional	ARY#RBAR ARY#RBA	X
Step 18 - Create SBRS subsystem setup repository.	Optional	ARYMOVER	-
Step 19 - Start Recovery Expert Server and agent tasks.		-	-

Step 1 - Copy the ARYEMAC1 CLIST

You will have to overlay the old macro ARYEMAC1 V1.1 member with the new version of the ARYEMAC1 V2.1 member. The edit macro "ARYEMAC1" can be used to edit and customize the JCL and DDL members in SARYSAMP to suit your environment during the installation process. The variables in this macro can be specified with values for each DB2 subsystem. For example, the #SSID variable in the macro should be assigned a value representing the DB2 subsystem for which DB2 Recovery Expert for z/OS is configured. After making the necessary changes, copy this macro to the SYSPROC library. If you plan to configure in more than one DB2 subsystem, copy this macro multiple times with unique names in the SYSPROC library.

Step 2 - Create DB2 Recovery Expert objects for V2.1

Edit and run the DDL in member ARYDDLRL. This creates the required DB2 objects in the SYSTOOLS database for System Backup and Restore Services.

Step 3 - Create image copies in the schema level repository

Edit and run the JCL in member ARYCPY1 to create image copies of DB2 Recovery Expert for z/OS table spaces. You may be required to run this JCL for each DB2 subsystem.

Step 4 - Bind Recovery Expert DB2 packages and plans

Based on the version of DB2, edit and run ARYBND71, ARYBND81, or ARYBND91 for DB2 Recovery Expert packages.

Based on the version of DB2, edit and run ARYBND72, ARYBND82, or ARYBND92 for DB2 Recovery Expert plans.

Based on the version of DB2, edit and run ARYBND73, ARYBND83, or ARYBND93 for DB2 grouper support.

Based on the version of DB2, edit and run ARYBND74, ARYBND84, or ARYBND94 for DB2 Automation Tool Support.

Note: For further details, read the comments section in each member.

Step 5 - Grant the DB2 Recovery Expert privilege

Edit and run the member ARYGRT1 to grant privileges required to access DB2 Recovery Expert. Read the instructions in member ARYGRT1 to edit the member to suit your environment. Note that you may be required to remove the GRANT statements.

Step 6 - Stop the DB2 Recovery Expert Agent and server tasks

Issue the following commands in the order listed to stop the agent and server address spaces:

```
/F STOP <agent-address space>  
/F STOP <server-address space>
```

Step 7 - Configure the Recovery Expert control file

Edit and run the member ARYSJ001 to update the product control file with DB2 Recovery Expert configuration information for each DB2 subsystem. Replicate the SYSIN card in this member for each DB2 subsystem and make the appropriate changes. A sample SYSIN card for one DB2 subsystem is provided in Example 2-1 on page 25.

Step 8 - Update the schema level repository

Edit and run the member ARYSJ002 to update the schema level repository with current information from the DB2 system catalog. Read the instructions completely in the commented section of this member.

Step 9 - Install the GUI client

Follow these steps to install the GUI client. The GUI client installation executable is a member of a data set from the z/OS installation.

1. Locate the ARY.IBMTAPE.SARYGUIW(ARYGUIW) data set member. In our environment it is ARY.V2R1M0.SARYGUIW(ARYGUIW).
2. Use FTP to transfer member ARYGUIW (in binary) to your workstation, renaming the member with a local file name of setupwin32.exe.
3. Run setupwin32.exe. Follow the instructions in the installation program if you want to change the location of the files. It is possible to install and run Recovery Expert from a network drive if you prefer.

Installation information:

- ▶ During installation, all of the DB2 Recovery Expert client components are copied to the location specified in the installation program. The default location is C:\Program Files\IBM\DB2TOOLS\DB2RecoveryExpert.
- ▶ No registry keys are created during the installation, but application preferences are saved under HKEY_CURRENT_USER\Software\JavaSoft\Prefs\com\rocketsoftware\ary when you run DB2 Recovery Expert.
- ▶ The installation does not create desktop icons, but it does create Start menu items that you use to start DB2 Recovery Expert.

Step 10 - Configure the server and agent

Edit the following members:

- ▶ ARYCFGA

Agent configuration file. The default values for all parameters may be used, except for <server-address> and <server-port>, which must be configured properly to connect to the DB2 Recovery Expert Server. For a sample configuration file, refer to Example 2-2 on page 27.

- ▶ ARYCFGS

Server configuration file. The default values for all parameters may be used.

Example 2-3 on page 28 shows a sample configuration file for the IBM DB2 Recovery Expert for z/OS 2.1.0 server.

► ARYSJSRV

Sample batch JCL to invoke the server. Customize this JCL to your environment and make use of it when you need to invoke the server in batch mode.

► ARYSJAGT

Sample batch JCL to invoke the agent. Customize this JCL to your environment and make use of it when you need to invoke the agent in batch mode.

► ARYSPSRV

The procedure provided in this member can be used to run DB2 Recovery Expert Server as a started task.

► ARYSPAGT

The procedure provided in this member can be used to run DB2 Recovery Expert Agent as a started task.

Step 11 - Update system parameters

This step is required if the ISPF component is installed during migration. This step is applicable to DB2 9 for z/OS and later. The following four parameters must be updated with the values shown:

```
SYSTEM_LEVEL_BACKUPS = NO
RESTORE_RECOVER_FROMDUMP = NO
UTILS_DUMP_CLASS_NAME = <blank>
RESTORE_TAPEUNITS = NOLIMIT
```

Step 12 - Create the SBRS VSAM repository

This step is required if the ISPF component is installed during migration. Edit and run the JCL in member ARYREPO to create VSAM data sets that hold backup and restore information. The data sets that the JCL in this member creates when run are:

```
ARY.PROFILES
ARY.PROFILE.MAPS
ARY.PROFILE.CATS
ARY.SYSBACK
ARY.SYSBACK.VOLS
ARY.SYSBACK.SSIDS
ARY.SYSBACK.OBJS
ARY.BREPORT
ARY.OFFOPTS
ARY.OBJECTS
```

Step 13 - Update the SBRS CLIST

This step is required if the ISPF component is installed during migration. Edit the CLIST member ARYV210. This CLIST is used by ARYV21 to execute the product and is used by the ISPF component only. Note that some of the file names used in this member are being created in other steps (like ARYSJ000, ARYREPO, ARYMOVER, ARY#RBAR, ARYGDG). You must change the default names used in this member as outlined in the initial comments section of the member. For a listing of the data sets used by the CLIST, refer to Table 2-3 on page 30.

Step 14 - Copy members to the CLIST library for SBRS

This step is required if the ISPF component is installed during migration. Edit and make the necessary changes to the following members and then move them to the SYSPROC library:

```
ARYV21
ARYV210
ARYTSOC
```

Step 15 - Create the GDG base for the VSAM repository

This is required if the ISPF component is installed during migration. Edit and run the member ARYGDG to create GDG bases that would be used to back up the repository files. The prefix and suffix of the GDG data sets are referenced in “Step 13 - Update the SBRS CLIST” on page 36.

Step 16 - Configure the SBRS PARMLIB member

This step is optional when the ISPF component is installed during migration. The ARY#PARM member provides settings that control various aspects of the backup and restore utilities. If you require changing a parameter from the defaults provided then update this member with the appropriate settings. The member ARY#PARM and the PDS data set that contains this member are referenced in “Step 13 - Update the SBRS CLIST” on page 36.

Step 17 - Configure the RBA capture utility

This step is optional when the ISPF component is installed during migration. Edit and run ARY#RBA to create the VSAM data set used by the RBA capture utility. This data set can be used to capture RBA information for all the DB2 subsystems involved in your installation.

Then update the member ARY#RBA with the newly created VSAM data set name and specify the time interval to capture the RBA information. Copy this member to SYS1.PROCLIB and you may start this started task.

The VSAM data set created in this step is referenced by the ARYV210 CLIST in “Step 13 - Update the SBRS CLIST” on page 36.

Step 18 - Create SBRS subsystem setup repository

This step is optional when the ISPF component is installed during migration. Edit and run the member ARYMOVER to create the VSAM data set to hold DB2 subsystem information. The VSAM data set created in this step is referenced by the ARYV210 CLIST in “Step 13 - Update the SBRS CLIST” on page 36.

Step 19 - Start Recovery Expert Server and agent tasks

Issue the following start commands to start the DB2 Recovery Expert Server and agent:

```
/S <server-address space>
/S <agent-address space>
```

Upon successful installation of the product, you can invoke the Recovery Expert for z/OS GUI client on the Microsoft® Windows platform by clicking Select **Start** → **Programs** → **IBM DB2 Recovery Expert** → **IBM DB2 Recovery Expert**.

You can invoke the ISPF interface by invoking the ARYV21 CLIST. You can also add the CLIST ARYV21 to the ISPF menu selection customized for your environment and invoke it from the ISPF primary option menu.

Archived

Our test environment

Different functionalities are provided by DB2 depending on its version and whether data sharing is implemented. Recovery Expert adapts to the DB2 functions present in the environment and complements them. We have considered DB2 V8 and DB2 9, and simple data sharing group and non-data sharing subsystems in our test environment with particular attention to the backup and restore capabilities.

In this chapter we describe the test environment that we used during development of this book and the corresponding setup parameters.

This chapter discusses:

- ▶ GUI environment configuration
- ▶ ISPF environment configuration

These configurations demonstrate the following tasks:

- ▶ How to configure Recovery Expert in a data sharing environment and a non-data sharing environment
- ▶ How to configure Recovery Expert with multiple managing environments to enable separation of recovery responsibilities between DBAs
- ▶ How various VSAM repositories are used in the ISPF component of Recovery Expert

3.1 GUI environment configuration

In a GUI environment, Recovery Expert supports the recovery of partitioned and non-partitioned table spaces, table space partitions, tables, and aggregate objects such as databases, storage groups, plans, packages, DB2 referential integrity groups, and DB2 Automation Tool object profiles. It currently does not support indexes as objects of recovery in the Recovery Expert GUI client.

Before we can show the step-by-step approach to perform recovery of various DB2 objects using the IBM DB2 Recovery Expert for z/OS, we need to set up a representative test environment. In this environment you will see two stand-alone (that is, non-data sharing) DB2 sub-systems, two 2-way data sharing groups running DB2 Version 8, and another two-way data sharing group running DB2 9.

Figure 3-1 shows the Recovery Expert configuration used for our recovery solutions with the following components:

- There are five logical partitions (LPARs), SC53, SC67, SC59, SC63, and SC64, grouped in three distinct environments: data sharing group D9CG, D8GG, LPAR SC59.

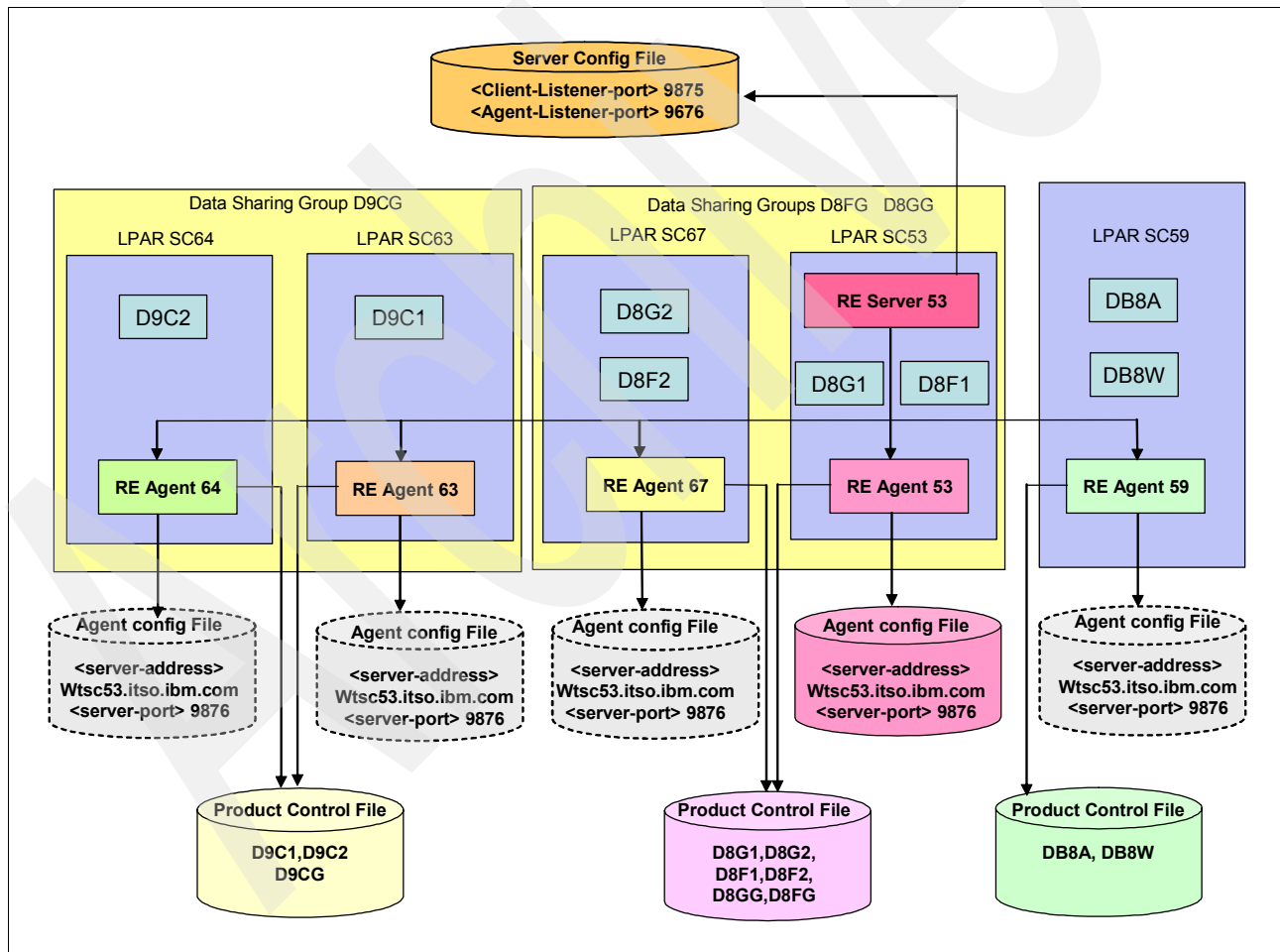


Figure 3-1 Sample Recovery Expert configuration describing GUI components

- LPAR SC59 is a single, non-data sharing environment with two DB2 subsystems, DB8A and DB8W.

- LPARs SC53 and SC67 form a sysplex that has two data sharing groups, D8GG and D8FG.
 - Data sharing group D8GG contains two member subsystems, namely, D8G1 and D8G2.
 - Data sharing group D8FG contain two member subsystems, namely, D8F1 and D8F2.
- LPAR SC63 and SC64 form a sysplex that has one data sharing group, D9CG. Data sharing group D9CG contains members D9C1 and D9C2.
- ▶ The DB2 subsystems on LPARs SC53, SC59, and SC67 are DB2 for z/OS Version 8. The DB2 subsystems in SC63 and SC64 are DB2 9 for z/OS.
- ▶ There is one *Recovery Expert Server* established in this configuration. Recovery Expert Server 53 is on one member of the sysplex LPAR SC53. The corresponding server configuration file is shown in Example 3-1.

Example 3-1 Snippets from server configuration file

```
<server-config>
<client-listener-port>9875</client-listener-port>
<agent-listener-port>9876</agent-listener-port>
<community-string>wtsc53</community-string>
.....
</server-config>
```

Note: If you would like to set up an environment where one DBA is responsible to manage all the production DB2 subsystems and another DBA to manage development DB2 subsystems, then you can set up a second server started task.

- ▶ There is a *Recovery Expert Agent* started task on each of the five LPARs. The agent on each LPAR and the corresponding DB2 subsystem information setup in the product control file are given as follows:
 - Recovery Expert Agent 53 on LPAR SC53 references Recovery Expert Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default). Snippets of this file are shown in Example 3-2.

Example 3-2 Snippets of Recovery Expert Agent 53 agent configuration file on LPAR SC53

```
<agent-config>
<server-address>wtsc53.itso.ibm.com</server-address>
<server-port>9876</server-port>
.....
</agent-config>
```

- Recovery Expert Agent 67 on LPAR SC67 references Recovery Expert Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default). The snippets of this file are shown in Example 3-3. Note that the contents of both the agent configuration files are identical in our case. If you decide to manage your environment with multiple servers then the <server-address> tag will contain different addresses.

Example 3-3 Snippets of Recovery Expert Agent 67 agent configuration file on LPAR SC67

```
<agent-config>
<server-address>wtsc53.itso.ibm.com</server-address>
<server-port>9876</server-port>
```

```
.....
</agent-config>
```

Recovery Expert Agent 53 on LPAR SC53 and Recovery Expert Agent 67 on LPAR SC67 share a common Recovery Expert product control file that identifies subsystems D8G1, D8G2, D8F1, and D8F2 pertaining to the data sharing groups D8GG and D8FG. To give you an idea on how we accomplished this, we listed job ARYSJ001 in Figure 3-2.

```
//PCFUPDT EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//SYSUDUMP DD SYSOUT=*
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
/*
/* REPORTS
/*
//SYSOUT DD SYSOUT=*,RECFM=FBA,LRECL=133 SYSIN REPORT
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133 PCF REPORT
/*
/* CONTROLS
/*
//SYSIN DD DSN=SG7606.ARY.V2R1M0.SARYSAMP(D8F1),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP(D8F2),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP(D8FG),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP(D8G1),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP(D8G2),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP(D8GG),DISP=SHR
.....
```

Figure 3-2 Snippets of the shared Recovery Expert control file setup job for Agents 53 and 67

We used the procedure described in “Step 8 - Configure the Recovery Expert product control file” on page 24 to create the control statements for one DB2 subsystem. We then copied that information to a PDS member, which has a name that represents the corresponding DB2 subsystem. For each member of the two data sharing groups (that is, D8F1, D8F2, D8G1, D8G2), the number of control statements is equal but the DB2 subsystem specific information like DSNZPARM name and BSDS names is provided with appropriate values. For a complete listing of the control statements, refer to Example E-1 on page 264 through Example E-4 on page 275. The PDS member pertaining to the group name should have just one control statement with group attach name assigned to DB2 SSID. For example, in the case of data sharing group D8FG, we used only one control statement:

```
SET DB2 SSID = D8FG
```

- Recovery Expert Agent 59 on LPAR SC59 references Recovery Expert Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default). The snippets of this file are shown in Example 3-4, and the complete file is shown in Example 3-4. Again, in this environment, since we used only one server, the contents of this agent configuration file are identical to the other two described above.

Example 3-4 Snippets of Recovery Expert Agent 59 agent configuration file

```
<agent-config>
<server-address>wtsc53.itso.ibm.com</server-address>
<server-port>9876</server-port>
.....
</agent-config>
```

Recovery Expert Agent 59 on LPAR SC59 has a separate Recovery Expert product control file that identifies the DB2 subsystems DB8W and DB8A. To give you an idea on how we accomplished this we listed job ARYSJ001 in Figure 3-3. For a detailed listing of the control statements refer to Example E-5 on page 279 through Example E-6 on page 282.

```
//PCFUPDT EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//SYSUDUMP DD SYSOUT=*
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//*
//* REPORTS
//*
//SYSOUT DD SYSOUT=*,RECFM=FBA,LRECL=133 SYSIN REPORT
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133 PCF REPORT
//*
//* CONTROLS
//*
//SYSIN DD DSN=SG7606.ARY.V2R1M0.SARYSAMP.DB8W(DB8A),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP.DB8W(DB8W),DISP=SHR
//*
//
```

Figure 3-3 Recovery Expert product control file setup job for Agent 59

Note: Each non-data sharing DB2 subsystem (or data sharing group) has its own SLR (DB2 tables) and each set should be updated separately as described in “Step 9 - Update the schema level repository” on page 27. Though it is essential for the GUI functioning, the SLR is not shown in Figure 4-1 to reduce the clutter.

- Recovery Expert Agent 63 references the Recovery Expert Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default). The snippets of this file are shown in Example 3-5.

Example 3-5 Snippets of the Recovery Expert Agent 63 configuration file

```
<agent-config>
<server-address>wtsc53.itso.ibm.com</server-address>
<server-port>9876</server-port>
.....
```

</agent-config>

- Recovery Expert Agent 64 references the Recovery Expert Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default). The snippets of this file are shown in Example 3-6.

Example 3-6 Snippets of the Recovery Expert Agent 64 configuration file

```
<agent-config>
<server-address>wtsc53.itso.ibm.com</server-address>
<server-port>9876</server-port>
.....
</agent-config>
```

Recovery Expert Agent 63 on LPAR SC63 and Recovery Expert Agent 64 on LPAR SC64 share a common IBM Recovery Expert product control file that identifies subsystems D9C1 and D9C2 pertaining to the data sharing groups D9CG. See job ARYSJ001 in Figure 3-4.

Detailed listings are shown in Example E-7 on page 286 and Example E-8 on page 290.

```
//PCFUPDT EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//SYSUDUMP DD SYSOUT=*
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
/*
/* REPORTS
/*
//SYSOUT DD SYSOUT=*,RECFM=FBA,LRECL=133 SYSIN REPORT
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133 PCF REPORT
/*
/* CONTROLS
/*
//SYSIN DD DSN=SG7606.ARY.V2R1M0.SARYSAMP.D9C1(D9C1),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP.D9C1(D9C2),DISP=SHR
// DD DSN=SG7606.ARY.V2R1M0.SARYSAMP.D9C1(D9CG),DISP=SHR
```

Figure 3-4 Recovery Expert product control file setup job for Agents 63 and 64

Attention: From the perspective of managing the recovery of DB2 objects in a data sharing or non-data sharing environment, the location of the Recovery Expert Server (whether it is on a member of a sysplex or on an LPAR that is not part of a sysplex) does not alter the approach for recovering DB2 objects using the Recovery Expert tool.

3.2 ISPF environment configuration

In an ISPF environment, Recovery Expert supports system level backup (using backup profiles), recovery of the entire DB2 subsystem, disaster recovery (that is, remote-site recovery), recovery of partitioned and non-partitioned table spaces, recovery of single objects and groups of objects (including indexes) using recovery object profiles, and some miscellaneous functions like converting time stamps to RBA/LRSN.

Next is the setup of the test environment for ISPF services of Recovery Expert that we implemented to back up and recover DB2 subsystem and objects. The z/OS and DB2 environments are the same as what we described in 3.1, “GUI environment configuration” on page 40, and repeat here for the sake of completeness. This section also describes various repositories used in the ISPF component of Recovery Expert.

Figure 3-5 on page 46 shows the Recovery Expert configuration for ISPF services used in our backup and recovery solutions with the following components:

- ▶ Five logical partitions (LPARs): SC53, SC67, SC59, SC63 and SC64.
- ▶ The DB2 subsystems on LPARs SC53, SC59, and SC67 are DB2 for z/OS Version 8. The DB2 subsystems in SC63 and SC64 are DB2 9 for z/OS.
- ▶ LPAR SC59 is a single, non-data sharing environment with two DB2 subsystems, DB8A and DB8W.
- ▶ LPARs SC53 and SC67 form a sysplex that has two data sharing groups, D8GG and D8FG:
 - Data sharing group D8GG contains subsystems D8G1 and D8G2.
 - Data sharing group D8FG contain subsystems D8F1 and D8F2.
- ▶ LPAR SC63 and SC64 form a sysplex that has one data sharing group, D9CG. Data sharing group D9CG contains members D9C1 and D9C2.
- ▶ There is one SBRS installed per sysplex environment or single non-data sharing environment.

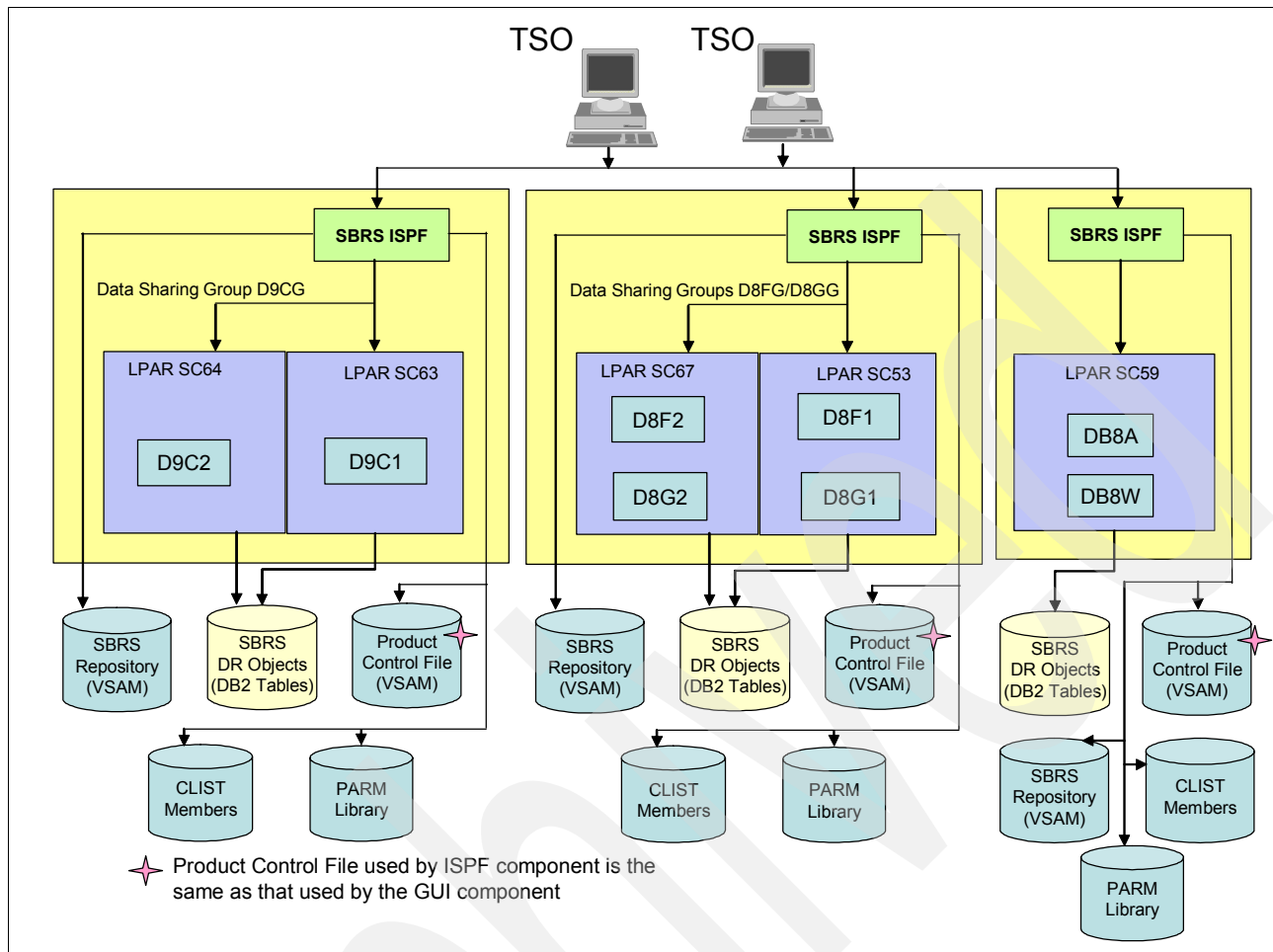


Figure 3-5 Recovery Expert configuration used in the user scenarios for ISPF services

Next is a description of the major components of the environment for ISPF services that we implemented in reference to the sysplex environment with LPARs SC53 and SC67. This implementation is the same for a sysplex environment with LPARs SC63 and SC64 and a single non-data sharing environment with LPAR SC59.

► The SBRS repository

The SBRS repository is a collection of VSAM data sets that are created during installation of Recovery Expert, as described in “Step 13 - Create the System Backup and Restore Services VSAM repository” on page 29. Table 3-1 lists *all* data sets of the SBRS. The *first 10* data sets must be added to the Recovery Expert product control file as described in Table 3-1.

Table 3-1 Repositories and corresponding VSAM data sets

Repository	VSAM data set
Profile repository	ARY.PROFILES
Source/target map repository	ARY.PROFILE.MAPS
Catalogs repository	ARY.PROFILE.CATS
System backup repository	ARY.SYSBACK
Backup volumes repository	ARY.SYSBACK.VOLS

Repository	VSAM data set
Backup SSID repository	ARY.SYSBACK.SSIDS
Backup objects repository	ARY.SYSBACK.OBJS
Backup report repository	ARY.BREPORT
Offload options repository	ARY.OFFOPTS
Profile objects repository	ARY.OBJECTS
Mover repository	ARY.SG7606.MOVDATA
RBA capture repository	SG7606.SC53.ALLSSIDS.RBA

► Sample JCL from ARYREPO member

A sample JCL from the ARYREPO member of the supplied SARYSAMP installation library is shown in Example E-9 on page 294. This JCL creates the *first 10* VSAM repository data sets.

► Sample JCL from ARYMOVER member to create *mover* repository

A sample JCL from the ARYMOVER member of the supplied SARYSAMP installation library is shown in Example E-10 on page 299. This JCL creates the “ARY.SG7606.MOVDATA” VSAM data set that contains information that is the result of analysis of a DB2 subsystem. You can conduct analysis of a DB2 subsystem from option M on the SBRS ISPF menu. For a detailed description refer to “Step 22 - Create the SBRS subsystem setup repository” on page 32. This data set must be added to ARYV210 CLIST.

► Sample JCLs from ARY#RBAR and ARY#RBA to create RBA capture repository

A sample JCL from the ARY#RBAR member of the supplied SARYSAMP installation library is shown in Example E-11 on page 299. This JCL creates the “SG7606.SC53.ALLSSIDS.RBA” VSAM data set to capture RBA information when the RBA capture utility is run. This VSAM data set must be added to the ARYV210 CLIST member.

The sample procedure from the ARY#RBA member of the supplied SARYSAMP installation library is shown in Example E-12 on page 300. This job, which runs as a started task, captures the DB2 time stamp with high used RBA to be able to perform point-in-time recovery. For a detailed description refer to “Step 21 - Configure the RBA capture utility” on page 32.

► Sample JCL from ARYGDG

A sample JCL to create GDG bases to back up the *first 10* repository data sets listed in Table 3-1 on page 46 is shown in Example E-13 on page 300. For a detailed description, refer to “Step 16 - Set up the job for repository backup data sets” on page 31. Ensure that the prefix of GDG bases is added to the variable RBRBUPRE and that the suffix of GDG bases is added to the variable RBRBUSUF in the CLIST ARYV210.

► SBRS DR objects

The disaster recovery DB2 objects that are required by SBRS are listed in Table 3-2. These objects are created by running the JCL provided in the member ARYDDLRL from the supplied SARYSAMP installation library and are used during disaster recovery.

Table 3-2 SBRS DB2 DR objects

DB2 object type	Name of the object
Table space	ARYDRCPY
Table space	ARYDRARC
Table	DR_IMAGE_COPY_CATL
Table	DR_ARCHIVE_LOGS00
Index	DR_ICOPY_CATL_IX
Index	DR_ARCHIVE_LOG_IX
View	DR_ICOPY_CATLV
View	DR_ARCHIVE_LOGSV
Alias	DR_ICOPY_CATL
Alias	DR_ARCHIVE_LOGS

► DDL from ARYDDLRL member

DDL from ARYDDLRL is shown in Example E-14 on page 301.

► Product control file

We created one product control file for each installed SBRS in our environment. If you can place the product control file in a shared disk, then you can have just one product control file for the entire environment.

The same product control file is shared between GUI and ISPF components of Recovery Expert. For a detailed description of how we set up the product control file, refer to “Step 8 - Configure the Recovery Expert product control file” on page 24.

There is one product control file for each installed SBRS in our environment. If we can place this file in a shared disk, then we can have one product control file for the entire environment. The same product control file is shared between GUI and ISPF components of Recovery Expert. For a detailed description of how we set up the product control file, refer to Figure 3-2 on page 42.

► CLIST members

There are four CLIST members:

ARYV21	ISPF Dialog start-up CLIST
ARYV210	ISPF Dialog start-up CLIST
ARYTSOC	Services CLIST
ARYCAPS	Uppercase conversion utility CLIST

These members are supplied in the SARYSAMP installation library. Sample CLISTs are shown in “Sample CLISTs” on page 303. For a detailed description of the CLIST, refer to “Step 14 - Update the System Backup and Restore Services CLISTs” on page 30 and “Step 15 - Customize the CLIST members for SBRS” on page 31.

- ▶ **PARM library**

The parmlib member has control statements that influence Backup and Recovery for DB2, the setup utility, the backup utility, snap global parameters, BCV split parameters, the restore utility, and the offload utility. Refer to “Step 17 - Change defaults for backup restore” on page 31 for description.

The parmlib data set and the parmlib member in that data set must be specified in the Recovery Expert product control file. For a description of the Recovery Expert product control file, refer to Figure 3-2 on page 42. For a detailed listing of the control statements in the parmlib member, refer to Example E-19 on page 306.

Considerations

Consider the following:

- ▶ Having different DB2 subsystems (DB8A and DB8W) on the same LPAR managed by two different Recovery Expert Servers

With Recovery Expert Server 53 (which manages DB8A) and Recovery Expert Server 59 (which manages DB8W), this effect is achieved by having the Recovery Expert Agent 53 and Recovery Expert Agent 59 on LPAR SC59 use different product control files. This simulates an environment where a more experienced DBA is responsible for managing a critical DB2 subsystem, and a less experienced DBA manages a less critical DB2 subsystem on the same LPAR.

- ▶ Having different DB2 subsystems (D8G1, D8G2, D8F1, and D8F2) on the same sysplex managed by two different IBM DB2 Recovery Expert Servers

With Recovery Expert Server 53 and Recovery Expert Server 59, this effect is achieved by having the agents Recovery Expert Agent 53 and Recovery Expert Agent 59 on LPARs SC53 and SC67 use the same product control files.

Archived

Backup and restore services

DB2 for z/OS V8 provides enhanced backup and restore 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.

DB2 9 for z/OS has added tape support, incremental FlashCopy support, and the ability to recover a DB2 object from a SYSTEM BACKUP.

Keep in mind that we show functions related to the implementation of a recovery schema for the DB2 subsystem. If you have a disaster recovery schema at the system level, you need to integrate the DB2 subsystem part depending on your overall solution.

In this part we describe the storage and DB2 functions used for BACKUP and RESTORE, then we show how the functions provided by Recovery Expert can help you in setting up and managing your DB2 environment so that you can satisfy the requirements for the backup and restore processes.

This part contains the following chapters:

- ▶ Chapter 4, “Configuring DB2 for system backup and restore” on page 53
- ▶ Chapter 5, “Local system level backup and restore” on page 83
- ▶ Chapter 6, “Object recovery functions from system level backup” on page 125

An overview of DB2 and storage functions is provided in Appendix C, “Storage and DB2 functions” on page 209.

Archived

Configuring DB2 for system backup and restore

The DB2 BACKUP and RESTORE SYSTEM utilities are the fastest and most effective backups available for DB2. In order for these utilities to be effective, there are strict guidelines on how DB2-related data sets are placed on your disk.

These utilities require the segregation of the logs and boot strap data sets from the rest of the DB2 subsystem. They also require additional ICF user catalogs that are backed up and restored with the DB2 subsystem. System Backup and Restore Services's subsystem setup facility will help us make our DB2 subsystem available for the DB2 BACKUP and RESTORE SYSTEM utilities.

In this chapter we discuss:

- ▶ Suggested data set naming conventions
- ▶ Planning for subsystem configuration
- ▶ Using the subsystem setup utility
- ▶ Profiles

4.1 Suggested data set naming conventions

Storage-based backups in DB2 Recovery Expert for z/OS are currently processed at the disk volume level. Because of this, naming conventions and data set separation are important considerations. The MVS™ user catalogs are backed up along with the DB2 subsystem. Therefore, strict rules need to be followed to ensure that the data is backed up and restored correctly. This section discusses the guidelines that ensure success in using the DB2 BACKUP SYSTEM and RESTORE SYSTEM utilities.

4.1.1 BSDS and logs

The BSDS and active log data sets should use the same high-level qualifier. This high-level qualifier should reside in its own individual MVS user catalog. Care must be taken that no other data sets either share the high-level qualifier or are cataloged in the same MVS user catalog.

4.1.2 DB2 catalog, directory, and user data

The data sets containing the DB2 catalog, directory, and user data must all be cataloged in the same MVS user catalog. There should not be any other data sets cataloged in this MVS user catalog. The catalog will be recovered along with the data when the RESTORE SYSTEM utility is run.

Note: Allowing data other than what is defined above to be cataloged in the ICF user catalogs or reside on the volumes containing the data sets associated with the DB2 subsystem or data sharing group can result in corruption of the unauthorized data.

Note: Load libraries, SMP/E files, and other data sets that may be used by a DB2 subsystem are not considered to be actual DB2 data sets for purposes of system level backup and system level restore.

4.2 Planning for subsystem configuration

Figure 4-1 shows the configuration that we implement for our two-member data sharing group. The proposed final configuration shown is what we consider to be the best possible configuration for a DB2 subsystem or data sharing group that is going to be backed up using the DB2 BACKUP SYSTEM utility.

In Figure 4-1 SMS storage classes are indicated below the volume images. Volume serial numbers are indicated in the upper section of the volume images. Data set masking and catalog locations are represented in the body of the volume images.

Note: It is extremely important that the DB2 subsystem be optimally configured before using system level backups. Using system level backups of an improperly configured subsystem or data sharing group can cause unpredictable results.

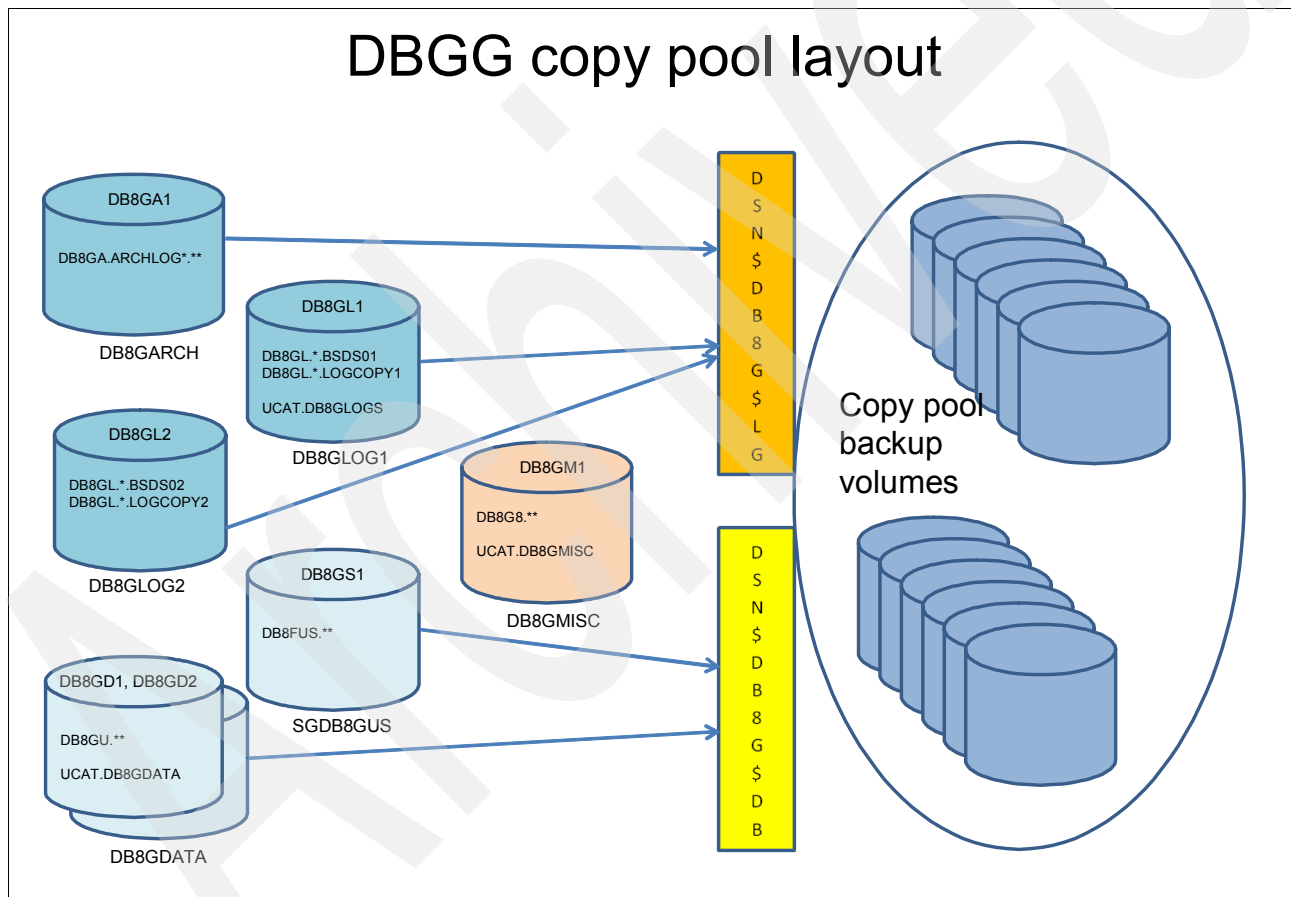


Figure 4-1 Copy pool layout

There are three classes of data shown in Figure 4-1. They are:

- ▶ DB2 BSDS, active logs, archive logs

The BSDS, active, and archive logs are shown in the upper left portion of Figure 4-1. The BSDS and active logs use a high-level qualifier of DB8GL. The archive logs use a high-level qualifier of DB8GA. Both of these high-level qualifiers are cataloged in the ICF user catalog named UCAT.DB8GLOGS. The catalog is allocated on volume DB8GL1.

We have also allocated two separate storage classes for the BSDS and logs. This allows us to use SMS to manage the placement of the data sets, and yet maintain separation of the various data sets (for example, BDSD01 and BSDS01). The storage classes are DB8LOG1 and DB8LOG2. Both of the storage classes are defined as part of the log copy pool. Copy pool definitions are discussed in 4.3.2, “Creating the DFSMS copy pools” on page 66.

Note: No data sets other than DB2 BSDS, active log, and in our case archive logs are defined in this ICF catalog or allocated on the volumes in the storage groups. This is what we call a *pristine* copy pool. Having the copy pools that will be backed up using the DB2 BACKUP SYSTEM utility setup in this way is highly recommended.

► DB2 catalog, directory, and application data

The DB2 data is shown in the lower left section of Figure 4-1 on page 55. The DB2 data is also split between two separate storage classes. Those storage classes are SGDB8GUS and DB8GDATA. Both storage classes are in the DB2 data copy pool.

Note: No data sets other than DB2 catalog, directory, and application data are defined in this ICF catalog, or allocated on the volumes in the storage groups. This is what we call a *pristine* copy pool. We highly recommend having the copy pools that will be backed up using the DB2 BACKUP SYSTEM utility setup in this way.

► Other miscellaneous data sets

This group is represented by the single volume in the center of Figure 4-1 on page 55, and it includes all other data sets that may in some way be associated with the DB2 data sharing group. These data sets include:

- DB2 System libraries (SDSNEXIT, DSDSNLOAD)
- SMP/E libraries and Consolidated Software Inventories (CSIs)
- Application load libraries

The data sets in this category have more dependencies on other system data sets than any data sets contained in the DB2 copy pools. Therefore, we recommend that data sets in this category be backed up and recovered along with the other system data sets.

Note: In an SMS-managed system, it is necessary for the SMS ACS routines to be modified to enforce the allocations in the storage groups associated with the copy pools discussed above.

4.3 Using the subsystem setup utility

System Backup and Restore Services’s subsystem setup facility is the Recovery Expert component that collects and displays information about the DB2 subsystem’s user catalogs, bootstrap data sets, active logs, and related DB2 object data sets. The subsystem setup utility also assists with the creation of new catalogs and aliases, moving and renaming log and BSDS data sets, and moving and renaming DB2 object data sets.

Once the information is collected, it is displayed on a scrollable ISPF panel divided into several sections. These sections are mapped in Figure 4-2 on page 57 through Figure 4-6 on page 61.

We can see from the subsystem setup utility data that this data sharing group is far from an optimal configuration for using the DB2 BACHKUP SYSTEM utility for system level backups shown in Figure 4-1 on page 55.

Figure 4-2 shows page 1 of the subsystem information setup screens. In this case we use a DB2 V8 subsystem, and we are set it up using the DB2 BACKUP SYSTEM utility. We recommend using the DB2 BACKUP SYSTEM utility for system level backups for DB2 V8 and later.

The top section of the window shows that the subsystem name is D8G1. The subsystem is currently active. It is a data sharing member, and the last analysis was run February 12th. Most importantly, there is a message on line 3 of the first section that reads Subsystem configuration prevents system level backup. Our goal it to change the configuration of the DB2 data sharing group to the configuration discussed in 4.2, “Planning for subsystem configuration” on page 55. Once we complete the reconfiguration, the message will read Subsystem configuration is optimal.

The middle section shows the new MVS user catalogs to be used by the data sharing group. This section can be used to provide JCL to create new ICF user catalogs as necessary. Since the DB2 table space and index space data sets must be cataloged in a separate ICF user catalog from the DB2 archive and active logs, it may be necessary to define new ICF user catalogs.

In the bottom section of the window a list of the ICF user catalogs currently in use is provided. In this case, there is only one ICF user catalog in use for all data sets associated with this particular DB2 data sharing group. The aliases associated with the single ICF user catalog in use by this DB2 data sharing group can be viewed using the “V” line command. Since we know that we will need at least two ICF user catalogs for data set separation, let us take a closer look at the aliases defined in the catalog UCAT.TOTDDL.

```

ARY$SSID      ----- Subsystem Setup Information ----- 2008/02/12  17:20:21
Option ==>                                         Scroll ==> CSR
-----
Subsystem: D8G1   Active: Yes   Datasharing: Yes
Date of Last Analysis: 02/12/2008   Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
                                         Row 1 of 49   +
New MVS User Catalogs to be used by this DB2 Subsystem
Log/BSDS Catl                                         Volume
DB2 Data Catl                                         Volume
Line Cmds: (C-Create, A-Add Alias, D-Dataset Disp, U-Update, V-View Alias)

Existing MVS User Catalogs used by this DB2 Subsystem
Data Log Other UCAT.TOTDDL                               Volume  TOTDDL
Line Cmds: (D-Dataset Display, V-View Aliases)

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 4-2 Subsystem setup - user catalogs

Figure 4-3 shows the results from a “View Alias” line command.

```

ARY$SSAL          ----- Usercat Alias List Display ----- 2008/02/12  15:57:52
Option  ==>                               Scroll ==> CSR
-----

Subsystem: D8G1   Usercat: UCAT.TOTDDL

The following aliases are in this usercat.  If the "DB2" column is Yes, the
alias is being used by this DB2 for either Logs or Data. The count is the
number of DB2 datasets using the alias name.

Row 1 of 3
-----
Cmd  Alias                DB2  Logs  Data  Count
    DB8GU                Yes  Yes   Yes   00000624
    DB8GUS               Yes  No    Yes   00000027
    DB8G8                No   No    No    N/A
***** Bottom of Data*****

```

Figure 4-3 Usercat Alias List Display

We see that there are three aliases associated with this ICF user catalog:

- ▶ DB8GU - This alias is associated with DB2 table space and index space data sets, DB2 log data sets, and non-DB2 data. These data sets need to be separated before we can use the DB2 BACKUP SYSTEM utility to create system level backups.
- ▶ DB8GUS - This alias is associated with only DB2 table space data sets. From an Recovery Expert perspective, this is a good setup.
- ▶ DB8G8 - This alias is associated with our ICF user catalog. However, none of the data sets using this alias contain DB2 table space data or DB2 log data.

Figure 4-4 shows the storage copy pools constructs and the boot strap data set section of the subsystem setup information.

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/12 18:15:41
Option  ==>                               Scroll ==> CSR
-----
Subsystem: D8G1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/12/2008  Analysis Recommended: Y
Message: Subsystem configuration prevents system-level backup.
-----
Row 10 of 49  -+

Storage Copy Pools
  Data Copy Pool  DSN$DB8G$DB
  Log Copy Pool   DSN$DB8G$LG
  Line Cmds: (V-Volume List)

Boot Strap Datasets
D8G1 - BSDS 1  DB8GU.D8G1.BSDS01          Volume  TOTDDL
D8G1 - BSDS 2  DB8GU.D8G1.BSDS02          Volume  TOTDDM
D8G2 - BSDS 1  DB8GU.D8G2.BSDS01          Volume  TOTDDL
D8G2 - BSDS 2  DB8GU.D8G2.BSDS02          Volume  TOTDDM
Line Cmds: (R-Rename BSDS, M-Move BSDS)

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 4-4 Subsystem setup - storage copy pools and boot strap data sets

At this point, the copy pools have not been created. Therefore, the copy pool name will be shown in red if the default colors in ISPF have not been changed. They are shown in bold in this case.

The boot strap data set section lists all of the boot strap data sets in use by this DB2 data sharing group. Line commands are provided to rename or move the boot strap data sets. The DB2 subsystems must be stopped before attempting to move or rename the boot strap data sets.

Figure 4-5 lists all of the active log data sets in use by this DB2 data sharing group. Line commands are available to rename or move the active log data sets. The DB2 subsystem that is using a given log data set must be stopped in order to rename or move an active log data set.

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/12 18:51:23
Option ==>                               Scroll ==> CSR
-----
Subsystem: D8G1    Active: Yes      Datasharing: Yes
Date of Last Analysis: 02/12/2008  Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
                                                    Row 22 of 49  -+
Active Log Datasets
D8G1 - Log 1      DB8GU.D8G1.LOGCOPY1.DS01          Volume TOTDDM
D8G1 - Log 1      DB8GU.D8G1.LOGCOPY1.DS02          Volume TOTDDM
D8G1 - Log 1      DB8GU.D8G1.LOGCOPY1.DS03          Volume TOTDDM
D8G1 - Log 2      DB8GU.D8G1.LOGCOPY2.DS01          Volume TOTDDL
D8G1 - Log 2      DB8GU.D8G1.LOGCOPY2.DS02          Volume TOTDDL
D8G1 - Log 2      DB8GU.D8G1.LOGCOPY2.DS03          Volume TOTDDL
D8G2 - Log 1      DB8GU.D8G2.LOGCOPY1.DS01          Volume TOTDDM
D8G2 - Log 1      DB8GU.D8G2.LOGCOPY1.DS02          Volume TOTDDM
D8G2 - Log 1      DB8GU.D8G2.LOGCOPY1.DS03          Volume TOTDDM
D8G2 - Log 2      DB8GU.D8G2.LOGCOPY2.DS01          Volume TOTDDL
D8G2 - Log 2      DB8GU.D8G2.LOGCOPY2.DS02          Volume TOTDDL
D8G2 - Log 2      DB8GU.D8G2.LOGCOPY2.DS03          Volume TOTDDL
Line Cmds: (R-Rename Log, M-Move Log)

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 4-5 Subsystem setup - active log data sets

Figure 4-6 shows the final section of the subsystem setup information. The top part is a list of the aliases in use by this DB2 data sharing group. Line commands are provided to display a list of data sets associated with each alias and to merge the alias into a new ICF user catalog.

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/12  19:01:10
Option  ==>                               Scroll ==> CSR
-----
Subsystem: D8G1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/12/2008  Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
Row 37 of 49  -
Alias used with associated MVS User Catalogs
DB8GU          UCAT.TOTDDL          Data Log Other
DB8GUS         UCAT.TOTDDL          Data
Line Cnds: (D-Dataset Display, M-Merge catalog entries, R-Rename Alias)

Volumes used by this DB2 Subsystem
Volume      Data  DataCat  ActLog  ActCat  ArcLog  ArcCat  Other  Pool
DB8GS1      Yes  No       No      No      No      No      No     No
TOTDDL     Yes  Yes      Yes     Yes     No      No      Yes    No
TOTDDM     Yes  No       Yes     No      No      No      Yes    No
TOTDDZ     Yes  No       No      No      No      No      No     No
Line Cnds: (D-Dataset Display, M-Move all Datasets on Volume)

***** Bottom of Data *****

Valid Primary Cnds: (Analyze, Reanalyze)

```

Figure 4-6 Subsystem setup - aliases and volumes in use

Note that the alias DB8GU has data sets that contain DB2 log data, DB2 table space data, and other data sets associated. This must be cleaned up before the subsystem is ready to use the DB2 BACKUP SYSTEM utility.

At the bottom of the subsystem setup information screen, there is a list of the volumes in use by this DB2 subsystem. For each volume, there is a column indicating whether the volume contains any of the following types of data:

- ▶ Data - DB2 data including the DB2 catalog and directory spaces
- ▶ DataCat - the ICF user catalog where the DB2 data is cataloged
- ▶ ActLog - DB2 boot strap or active log data sets
- ▶ ActCat - the ICF user catalog where the DB2 boot strap and active log data sets are cataloged
- ▶ ArcLog - DB2 archive log data sets used by this DB2 subsystem
- ▶ ArcCat - the ICF user catalog where the DB2 archive log data sets are cataloged
- ▶ Other - any other category of data set that resides on the volume
- ▶ Pool - the volume assigned to a SMS copy pool

The volumes appear in various ISPF default colors, depending on the status of the volume. (Note that the colors may be different if the ISPF default colors have been changed.)

- ▶ Dark blue: The volume is optimal.
- ▶ Light blue: The volume contains data other than DB2 data.
- ▶ Pink: Both log and object data reside on the volume.

- Red: The volume cannot be backed up by ARY.

In this case, the volumes TOTDDL and TOTDDM contain a mixture of DB2 and other data, and would be shown in pink. However, since the SMS copy pools are not yet defined, all of the volumes would be shown in red (bold in our figure).

In many cases there will be a -NONE- volume displayed if DB2 Recovery Expert for z/OS finds table space or index space objects defined to Db2, but no data set has been defined for the object. In addition, archive logs that are still defined in the BSDS, but no longer exist, will cause a -NONE- volume to appear also. The "D" line command on the -NONE- volume will list of all the data sets DB2 Recovery Expert for z/OS expected to find but did not.

Now that we have reviewed the subsystem setup information windows, it is time to build a list of things to do to reconfigure this DB2 subsystem to use the DB2 BACKUP SYSTEM UTILITY.

- Figure 4-2 on page 57 shows that we have a single ICF user catalog with three distinct classes of data. This will need to be corrected by creating new ICF user catalogs for the DB2 data and DB2 logs.
- Figure 4-3 on page 58 shows that some aliases reference multiple types of data (DB2 data and log data).
- Figure 4-6 on page 61 shows that the alias entry DB8GU references all three distinct types of data (DB2 table spaces, DB2 logs, and other data). We must rename the data sets and move from one ICF user catalog to another.
- Figure 4-4 on page 59 shows that the SMS copy pools have not been defined. We must define copy pools
- Figure 4-6 on page 61 shows that we also need to clean up the volumes TOTDDL and TOTDDM by moving the DB2 data and the log data off of the volume.

4.3.1 Defining the new ICF user catalogs

The first step to reconfiguring a DB2 subsystem to use the BACKUP SYSTEM utility is to create new ICF user catalogs for the DB2 table space, active logs, and boot strap data sets. Using the “C” line command in the User Catalog section of the subsystem setup information panel, we enter the new catalog names and the volumes where we want the catalogs placed, as shown in Figure 4-7.

```
ARY$SSID      ----- Subsystem Setup Information ----- 2008/02/13 17:19:03
Option ==>                                         Scroll ==> CSR
-----
Subsystem: D8G1   Active: Yes   Datasharing: Yes
Date of Last Analysis: 02/13/2008   Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
Row 1 of 49   +
New MVS User Catalogs to be used by this DB2 Subsystem
C  Log/BSDS Cat1  UCAT.DB8GLOGS                               Volume DB8LG1
C  DB2 Data Cat1  UCAT.DB8GDATA                               Volume DB8GD1
Line Cmds: (C-Create, A-Add Alias, D-Dataset Disp, U-Update, V-View Alias)

Existing MVS User Catalogs used by this DB2 Subsystem
Data Log Other UCAT.TOTDDL                                     Volume TOTDDL
Line Cmds: (D-Dataset Display, V-View Aliases)

Storage Copy Pools
Data Copy Pool DSN$DB8G$DB
Log Copy Pool  DSN$DB8G$LG
Line Cmds: (V-Volume List)

Valid Primary Cmds: (Analyze, Reanalyze)
```

Figure 4-7 Using Subsystem setup to create new ICF user catalogs

When the Enter key is pressed, the update user catalog information panel for the logs catalog shown in Figure 4-8 is displayed.

```
ARY$SSBU      ----- Update User Catalog Information ----- 2008/02/13  18:59:11
Option  ==>>

Subsystem: D8G1

User Catalog Name      ==>  UCAT.DB8GLOGS
User Catalog Volume    ==>
SMS Storage Class      ==>  DB8GLOG1

Data Parameters
  Tracks or Cylinders  ==>  C      (Tracks/Cylinders)
  Primary Quantity     ==>  10
  Secondary Quantity   ==>  5
  Data Buffers         ==>  4

Index Parameters
  Tracks or Cylinders  ==>  C      (Tracks/Cylinders)
  Primary Quantity     ==>  1
  Secondary Quantity   ==>  1
  Index Buffers        ==>  4

User Catalog Aliases   ==>  DB8GL      DB8GA
```

Figure 4-8 Update User Catalog Information - logs catalog

The panel is automatically populated with the catalog name and volume information that we provided. We will update the panel with space information and any aliases that we want associated with this ICF user catalog.

We use the PF3 key or end command once we have completed filling in the information.

The IDCAMS interface module shown in Figure 4-9 is displayed.

```
ARY$IDCM      ----- IDCAMS Interface Module ----- 2008/02/13 19:19:26
Option  ==>                                     Scroll ==> CSR

Action  ==>          (E - EDIT, 0 - ONLINE SUBMISSION, B - BATCH SUBMISSION)

----- IDCAMS Input Cards -----                               Row 1 of 22  +>

SET MAXCC = 0
DEFINE USERCATALOG(NAME('UCAT.DB8GLOGS') -
  STORCLAS(DB8GLOG1) -
  CYLINDERS(10 5) -
  ICFCATALOG) -
DATA -
  (BUFND(4) -
  CYLINDERS(10 5) -
  CONTROLINTERVALSIZE(4096)) -
INDEX -
  (BUFNI(4) -
  CYLINDERS(1 1) -
  CONTROLINTERVALSIZE(2048)) -
  CAT(MCAT.ZOSR18.Z18CAT)
DEFINE ALIAS -
  (NAME(DB8GL) -
  RELATE(UCAT.DB8GLOGS)) -
  CATALOG(MCAT.ZOSR18.Z18CAT)
DEFINE ALIAS -
  (NAME(DB8GA) -
  RELATE(UCAT.DB8GLOGS)) -
  CATALOG(MCAT.ZOSR18.Z18CAT)
```

Figure 4-9 IDCAMS interface module

From this module, we have three options:

- ▶ Edit the IDCAMS input.
- ▶ Online submission to create the new catalog immediately.
- ▶ Batch submission, which will create a batch job that can be submitted later. Since we do not have UPDATE access to the ICF master catalog, we need to use the batch submission option and have a user with appropriate authorization run the job for us. A panel is displayed for us to enter job card information and specify a data set where the generated JCL should be saved.

Note: DB2 Recovery Expert for z/OS automatically determines the ICF master catalog name.

Since we elected to create both catalogs, when we have finished with the logs catalog, the update user catalog information for the DB2 data catalog is displayed. This panel is shown in Figure 4-10. We need to repeat the data entry steps as we did for the logs catalog.

```

ARY$SSBU      ----- Update User Catalog Information ----- 2008/02/13  19:04:03
Option  ==>

Subsystem: D8G1

User Catalog Name      ==>  UCAT.DB8GDATA
User Catalog Volume    ==>
SMS Storage Class      ==>  DB8GDATA

Data Parameters
  Tracks or Cylinders  ==>  C      (Tracks/Cylinders)
  Primary Quantity     ==>  10
  Secondary Quantity   ==>  5
  Data Buffers         ==>  4

Index Parameters
  Tracks or Cylinders  ==>  C      (Tracks/Cylinders)
  Primary Quantity     ==>  1
  Secondary Quantity   ==>  1
  Index Buffers        ==>  4

User Catalog Aliases   ==>

```

Figure 4-10 Update User Catalog Information - data catalog

4.3.2 Creating the DFSMS copy pools

For information about defining copy pools and associated copy pool backup storage groups, refer to Chapter 9, “Defining Copy Pools,” in the *z/OS DFSMS Storage Administration Reference (for DFSMSdfp, DFSMSdss, DFSMShsm)*, SC26-7402-08. The following DB2 naming convention must be used to define the copy pools:

DSN\$locn-name\$cp-type

The variables that are used in this naming convention have the following meanings:

DSN	The unique DB2 product identifier.
\$	A fixed character delimiter.
locn-name	The DB2 location name.
\$	A fixed character delimiter.
cp-type	The copy pool type. Use <i>DB</i> for database and <i>LG</i> for log.

4.3.3 Renaming and moving the BSDS and log data sets

Once the ICF user catalogs have been created, it is time to rename, re-catalog, and move the DB2 boot strap and active log data sets to their new volumes. DB2 Recovery Expert for z/OS can either rename or move the boot strap data sets. Both functions can also be performed in a single operation.

Moving/rename the boot strap data sets

Figure 4-11 shows the boot strap data set section of the subsystem setup information window.

Note: The DB2 subsystem owning the boot strap data sets that are being moved/rename must not be active during the move/rename process. This includes the process of generating the commands as DB2 has the data sets allocated.

```
ARY$SSID      ----- Subsystem Setup Information ----- 2008/02/14 16:06:44
Option ==>                                         Scroll ==> PAGE
-----
Subsystem: D8G1   Active: No      Datasharing: Yes
Date of Last Analysis: 02/14/2008  Analysis Recommended: Y
Message: Subsystem configuration allows only full restore (data and logs).
-----
                                           Row 15 of 49  -+
Boot Strap Datasets
R  D8G1 - BSDS 1  DB8GU.D8G1.BSDS01           Volume  TOTDDL
   D8G1 - BSDS 2  DB8GU.D8G1.BSDS02           Volume  TOTDDM
   D8G2 - BSDS 1  DB8GU.D8G2.BSDS01           Volume  TOTDDL
   D8G2 - BSDS 2  DB8GU.D8G2.BSDS02           Volume  TOTDDM
   Line Cmds: (R-Rename BSDS, M-Move BSDS)

Active Log Datasets
   D8G1 - Log 1   DB8GU.D8G1.LOGCOPY1.DS01     Volume  TOTDDM
   D8G1 - Log 1   DB8GU.D8G1.LOGCOPY1.DS02     Volume  TOTDDM
   D8G1 - Log 1   DB8GU.D8G1.LOGCOPY1.DS03     Volume  TOTDDM
   D8G1 - Log 2   DB8GU.D8G1.LOGCOPY2.DS01     Volume  TOTDDL
   D8G1 - Log 2   DB8GU.D8G1.LOGCOPY2.DS02     Volume  TOTDDL
   D8G1 - Log 2   DB8GU.D8G1.LOGCOPY2.DS03     Volume  TOTDDL

Valid Primary Cmds: (Analyze, Reanalyze)
```

Figure 4-11 Rename BSDS

We use the “R” line command as shown to rename a single boot strap data set. We can optionally rename or move all of the data sets in the section by using the line command on the section header line of the panel.

Once the line command is ready, press Enter to process the command.

Figure 4-12 shows the boot strap data set rename panel. The panel is displayed after entering a line command in the boot strap data set section of the subsystem setup information panel.

```
ARY$SSBD      ----- Boot Strap Dataset Rename ----- 2008/02/14 16:08:56
Option  ==>
-----

Subsystem: D8G1

You have selected the Rename Boot Strap Dataset Function. You have
entered this command on a detail line which updates the alias for
DB8GU.D8G1.BSDS01.
The New Boot Strap Alias will be substituted for the first node of the
the existing dataset name. You may use a symbolic of &SSID for
substituting the DB2 subsystem name.

      New Boot Strap Alias: DB8GL                      Ex: DB2&SSID

You may optionally specify the following:

      New Volume:
      New SMS Storage Class: DB8GLOG1

Press  Enter to process rename.  Press Cancel to cancel request.
```

Figure 4-12 Boot Strap Data Set Rename

On this panel, we enter the new high-level qualifier for the boot strap data sets. This must be one of the high-level qualifiers assigned to the ICF user catalog identified in Figure 4-7 on page 63.

If we are moving the boot strap data sets, we also need to provide either the new volser or the SMS storage class where the data sets will be moved to. In this example, we are using the SMS storage class to control the volumes used for the new boot strap data set location. The values we entered are shown in bold in Figure 4-12.

As shown at the bottom of the panel, press the Enter key to process our request. The IDCAMS interface module panel will be displayed.

Figure 4-13 shows the IDCAMS interface module. We have three options at this point. We can:

- ▶ Edit the IDCAMS input cards.
- ▶ Process the IDCAMS input online.
- ▶ Save the control cars in a batch job to be submitted later.

In this example, we choose to save the input cards and process them as a batch job. We review the results of the batch job in Figure 4-17 on page 73.

```

ARY$IDCM      ----- IDCAMS Interface Module ----- 2008/02/14 16:48:37
Option ==>                                         Scroll ==> PAGE

Action ==> B      (E - EDIT, O - ONLINE SUBMISSION, B - BATCH SUBMISSION)

----- IDCAMS Input Cards ----- Row 1 of 16  >

/*-----*/

DEFINE CLUSTER -
  (NAME('DB8GL.D8G1.BSDS01') -
   STORAGECLASS(DB8GLOG1) -
   MODEL('DB8GU.D8G1.BSDS01'))

IF MAXCC > 0 THEN CANCEL

REPRO INDATASET('DB8GU.D8G1.BSDS01') -
      OUTDATASET('DB8GL.D8G1.BSDS01') -
      REPLACE

IF MAXCC > 0 THEN CANCEL

DELETE 'DB8GU.D8G1.BSDS01'

```

Figure 4-13 IDCAMS Interface Module - BSDS

Moving/rename the active log data sets

Figure 4-14 shows the active log data set section of the subsystem setup information window.

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/14 17:01:18
Option ==>                                     Scroll ==> PAGE
-----
Subsystem: D8G1    Active: No          Datasharing: Yes
Date of Last Analysis: 02/14/2008  Analysis Recommended: Y
Message: Subsystem configuration allows only full restore (data and logs).
-----
                                                    Row 15 of 49  +-
Boot Strap Datasets
  D8G1 - BSDS 1    DB8GL.D8G1.BSDS01          Volume DB8GL1
  D8G1 - BSDS 2    DB8GL.D8G1.BSDS02          Volume DB8GL2
  D8G2 - BSDS 1    DB8GL.D8G2.BSDS01          Volume DB8GL1
  D8G2 - BSDS 2    DB8GL.D8G2.BSDS02          Volume DB8GL2
  Line Cmds: (R-Rename BSDS, M-Move BSDS)

R Active Log Datasets
  D8G1 - Log 1     DB8GU.D8G1.LOGCOPY1.DS01    Volume TOTDDM
  D8G1 - Log 1     DB8GU.D8G1.LOGCOPY1.DS02    Volume TOTDDM
  D8G1 - Log 1     DB8GU.D8G1.LOGCOPY1.DS03    Volume TOTDDM
  D8G1 - Log 2     DB8GU.D8G1.LOGCOPY2.DS01    Volume TOTDDL
  D8G1 - Log 2     DB8GU.D8G1.LOGCOPY2.DS02    Volume TOTDDL
  D8G1 - Log 2     DB8GU.D8G1.LOGCOPY2.DS03    Volume TOTDDL

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 4-14 Rename active log data sets

Note: The DB2 subsystem owning the active log data sets that are being moved/rename must not be active during the move/rename process. This includes the process of generating the commands as DB2 has the data sets allocated.

We can use the “R” line command as shown to rename all of the active log data sets in the data sharing group. We can optionally rename or move individual active log data sets in the section by using the line command on an individual data set line of the panel.

Once the line command is ready, press Enter to process the command.

Figure 4-15 shows the active log data set rename panel. On this panel, we enter the new high-level qualifier for the active log data sets. This must be one of the high-level qualifiers assigned to the Log/BSDS ICF user catalog identified in Figure 4-7 on page 63.

```
ARY$SSAH      ----- Active Log Dataset Rename ----- 2008/02/14 17:01:39
Option  ==>
-----

Subsystem: D8G1

You have selected the Rename Active Log Dataset Function. You have
entered this command on a header line which update the alias for
all Active Log datasets in the list.
The New Active Log Alias will be substituted for the first node of the
the existing dataset names. You may use a symbolic of &SSID for
substituting the DB2 subsystem name.

      New Boot Strap Alias: DB8GL                      Ex: DB2&SSID

You may optionally specify the following:

      New Volume:
      New SMS Storage Class: DB8GLOG1

Press  Enter to process rename.  Press Cancel to cancel request.
```

Figure 4-15 Active Log Data Set Rename

Since we are moving the active log data sets, we also need to provide either the new volser or the SMS storage class where the data sets will be moved to. In this example, we are using the SMS storage class to control the volumes used for the new active log data set location. The values we entered are shown in bold in Figure 4-15.

We press the Enter key to process the command and display the IDCAMS interface module.

Figure 4-16 shows the IDCAMS interface module. We have three options at this point. We can:

- ▶ Edit the IDCAMS input cards.
- ▶ Process the IDCAMS input online.
- ▶ Save the control cards in a batch job to be submitted later.

In this example, we choose to edit the input cards to specify different SMS storage classes for the logcopy1 and logcopy2 data sets. Then we save the input cards and process them as a batch job.

```

ARY$IDCM      ----- IDCAMS Interface Module ----- 2008/02/14 16:48:37
Option ==>
Scroll ==> PAGE

Action ==> B      (E - EDIT, O - ONLINE SUBMISSION, B - BATCH SUBMISSION)

----- IDCAMS Input Cards ----- Row 1 of 16  >

/*-----*/

DEFINE CLUSTER -
    (NAME('DB8GL.D8G1.LOGCOPY1.DS01') -
    STORAGECLASS(DB8GLOG1) -
    MODEL('DB8GU.D8G1.LOGCOPY1.DS01'))

IF MAXCC > 0 THEN CANCEL

REPRO INDATASET('DB8GU.D8G1.LOGCOPY1.DS01') -
    OUTDATASET('DB8GL.D8G1.LOGCOPY1.DS01') -
    REPLACE

IF MAXCC > 0 THEN CANCEL

DELETE 'DB8GU.D8G1.LOGCOPY1.DS01'

```

Figure 4-16 IDCAMS Interface Module - active logs

Let us review what has been accomplished so far. Figure 4-17 on page 73 shows the results of a re-analysis of our DB2 data sharing group after we have completed the previous steps. Note the following changes in bold in Figure 4-17 on page 73:

- ▶ The ICF user catalogs for the BSDS/logs and the DB2 table space data sets have been created. The new catalogs appear in the New MVS User Catalogs to be used by this DB2 Subsystem section of the display.
- ▶ The SMS copy pools have been defined. They will now appear in blue on the panel if the default ISPF color settings have not been changed.
- ▶ The boot strap data sets have been renamed and recataloged.
- ▶ The active log data sets have been renamed and recataloged.
- ▶ The new alias that we used for the BSDS and logs appears in the Alias used with associated MVS User Catalogs section.
- ▶ The volumes DB8GL1 and DB8GL2 now appear in the Volumes used by this DB2 Subsystem section.
- ▶ Volumes DB8GL1, DB8GL2, and DB8GS1 are shown as being associated with a SMS copy pool.


```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/14 18:21:33
Option  ===>                                           Scroll ==> PAGE
-----
Subsystem: D8G1    Active: Yes      Datasharing: Yes
Date of Last Analysis: 02/14/2008  Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
Row 1 of 54  +
New MVS User Catalogs to be used by this DB2 Subsystem
  Log/BSDS Cat1  UCAT.DB8GLOGS                               Volume DB8GL1
  DB2 Data Cat1  UCAT.DB8GDATA                               Volume DB8GD1
  Line Cmds: (C-Create, A-Add Alias, D-Dataset Disp, U-Update, V-View Alias)

Existing MVS User Catalogs used by this DB2 Subsystem
  Log            UCAT.DB8GLOGS                               Volume DB8GL1
  Data Other     UCAT.TOTDDL                                  Volume TOTDDL
  Line Cmds: (D-Dataset Display, V-View Aliases)

Storage Copy Pools
  Data Copy Pool DSN$DB8G$DB
  Log Copy Pool  DSN$DB8G$LG
  Backup Pool    DB8GCPB
  Line Cmds: (V-Volume List)

Boot Strap Datasets
  D8G1 - BSDS 1  DB8GL.D8G1.BSDS01                          Volume DB8GL1
  D8G1 - BSDS 2  DB8GL.D8G1.BSDS02                          Volume DB8GL2
  D8G2 - BSDS 1  DB8GL.D8G2.BSDS01                          Volume DB8GL1
  D8G2 - BSDS 2  DB8GL.D8G2.BSDS02                          Volume DB8GL2
  Line Cmds: (R-Rename BSDS, M-Move BSDS)

Active Log Datasets
  D8G1 - Log 1   DB8GL.D8G1.LOGCOPY1.DS01                   Volume DB8GL1
  D8G1 - Log 1   DB8GL.D8G1.LOGCOPY1.DS02                   Volume DB8GL1
  D8G1 - Log 1   DB8GL.D8G1.LOGCOPY1.DS03                   Volume DB8GL1
  D8G1 - Log 2   DB8GL.D8G1.LOGCOPY2.DS01                   Volume DB8GL2
  D8G1 - Log 2   DB8GL.D8G1.LOGCOPY2.DS02                   Volume DB8GL2
  D8G1 - Log 2   DB8GL.D8G1.LOGCOPY2.DS03                   Volume DB8GL2
  D8G2 - Log 1   DB8GL.D8G2.LOGCOPY1.DS01                   Volume DB8GL1
  D8G2 - Log 1   DB8GL.D8G2.LOGCOPY1.DS02                   Volume DB8GL1
  D8G2 - Log 1   DB8GL.D8G2.LOGCOPY1.DS03                   Volume DB8GL1
  D8G2 - Log 2   DB8GL.D8G2.LOGCOPY2.DS01                   Volume DB8GL2
  D8G2 - Log 2   DB8GL.D8G2.LOGCOPY2.DS02                   Volume DB8GL2
  D8G2 - Log 2   DB8GL.D8G2.LOGCOPY2.DS03                   Volume DB8GL2
  Line Cmds: (R-Rename Log, M-Move Log)

Alias used with associated MVS User Catalogs
  DB8GL          UCAT.DB8GLOGS                               Log
  DB8GU          UCAT.TOTDDL                                  Data Other
  DB8GUS         UCAT.TOTDDL                                  Data
  Line Cmds: (D-Dataset Display, M-Merge catalog entries, R-Rename Alias)

Volumes used by this DB2 Subsystem
  Volume  Data  DataCat  ActLog  ActCat  ArcLog  ArcCat  Other  Pool
  DB8GL1  No   No       Yes    Yes    No     No       No     Yes
  DB8GL2  No   No       Yes    No     No     No       No     Yes
  DB8GS1  Yes  No       No     No     No     No       No     Yes
  TOTDDL  Yes  Yes      No     No     No     No       Yes    No
  TOTDDM  Yes  No       No     No     No     No       Yes    No
  TOTDDZ  Yes  No       No     No     No     No       No     No
  Line Cmds: (D-Dataset Display, M-Move all Datasets on Volume)
***** Bottom of Data *****
Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 4-17 Subsystem information after BSDS and log renames

4.3.4 Moving the DB2 table space data sets to new volumes

Now we are ready to move your DB2 table space data sets to their new volumes. This process is done either one data set at a time or one volume at a time. In this case, we move data sets one volume at a time.

We use the “M” line command as shown in Figure 4-18 to bring up the DB2 data set move panel shown in Figure 4-19 on page 75.

```
ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/14 18:34:19
Option ===>                               Scroll ===> PAGE
-----
Subsystem: D8G1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/14/2008 Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
Row 40 of 54  -+
Alias used with associated MVS User Catalogs
DB8GL          UCAT.DB8GLOGS          Log
DB8GU          UCAT.TOTDDL            Data Other
DB8GUS         UCAT.TOTDDL            Data
Line Cmds: (D-Dataset Display, M-Merge catalog entries, R-Rename Alias)

Volumes used by this DB2 Subsystem
Volume      Data  DataCat  ActLog ActCat  ArcLog ArcCat  Other  Pool
DB8GL1      No   No       Yes   Yes    No     No     No     Yes
DB8GL2      No   No       Yes   No     No     No     No     Yes
DB8GS1      Yes  No       No    No     No     No     No     Yes
M  TOTDDL    Yes  Yes      No    No     No     No     Yes    No
  TOTDDM     Yes  No       No    No     No     No     Yes    No
  TOTDDZ     Yes  No       No    No     No     No     No     No
Line Cmds: (D-Dataset Display, M-Move all Datasets on Volume)

Valid Primary Cmds: (Analyze, Reanalyze)
```

Figure 4-18 Moving all data sets on a volume

We can move all data sets on one volume by using the M line command on a volume listed in the *Volumes used by this subsystem* list.

Figure 4-19 shows the DB2 data set move panel.

```
ARY$SSVM      ----- DB2 Dataset Move ----- 2008/02/14  18:35:05
Option  ==>
-----

Subsystem: D8G1

You have selected to move all DB2 object datasets and/or other datasets that
reside on volume TOTDDL to a new set of volumes.  DB2 Log data cannot be
moved through this interface.  Specifying "Other Data" will generate moves
for all datasets not associated with this DB2 (will not move other DB2
vsam or MVS usercat datasets).

      New Volumes      :
      SMS Storage Class : DB8GDATA
      DB2 Object Data   : Y
      Other Data        : N

Press  PF3 to process move.  Press Cancel to cancel request.
```

Figure 4-19 DB2 data set move

On this panel we enter the volume where we want the data to be moved or the new SMS storage class to be used for the data sets. In this example, we are moving from a non-SMS volume to an SM-managed system, so we enter the storage class and let SMS handle data set placement.

The movement of DB2 table space data sets is repeated for each volume where we no longer want to have DB2 table spaces. In this example, we are moving table space data sets off of TOTDDL, TOTDDZ, and TOTDDM.

Volumes TOTDDL and TOTDDM contain data sets that use the same high-level qualifier as the DB2 table space data sets. These other data sets need to be renamed to use a different high-level qualifier before we can proceed with the final step.

Note: If any of the data sets on the volume are DB2 system catalog or directory data sets, then DB2 Recovery Expert for z/OS detects this and generates a stop for the subsystem. In a data sharing environment, we need to manually stop the members running on other LPARs.

4.3.5 Merging the DB2 table space alias to its new ICF user catalog

We are now ready to merge the alias for the DB2 table space data sets to its new ICF user catalog. In this case, we have two high-lever qualifiers to merge. They are DB8GU and DB8GUS. We use DB8GU for our model. We start by using the “M” line command next to the alias that we want to merge, as shown in Figure 4-20.

```
ARY$SSID      ----- Subsystem Setup Information ----- 2008/02/14 18:34:19
Option ==>                                           Scroll ==> PAGE
-----
Subsystem: D8G1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/14/2008  Analysis Recommended: Y
Message: Subsystem configuration prevents system level backup.
-----
Row 40 of 54  -+
Alias used with associated MVS User Catalogs
DB8GL          UCAT.DB8GLOGS          Log
M DB8GU         UCAT.TOTDDL           Data
DB8GUS         UCAT.TOTDDL           Data
Line Cmds: (D-Dataset Display, M-Merge catalog entries, R-Rename Alias)

Volumes used by this DB2 Subsystem
Volume      Data  DataCat  ActLog  ActCat  ArcLog  ArcCat  Other  Pool
DB8GL1      No   No       Yes     Yes     No      No      No     Yes
DB8GL2      No   No       Yes     No      No      No      No     Yes
DB8GS1      Yes  No       No      No      No      No      No     Yes
TOTDDL      Yes  Yes      No      No      No      No      Yes    No
TOTDDM      Yes  No       No      No      No      No      Yes    No
TOTDDZ      Yes  No       No      No      No      No      No     No
Line Cmds: (D-Dataset Display, M-Move all Datasets on Volume)

Valid Primary Cmds: (Analyze, Reanalyze)
```

Figure 4-20 Moving a catalog alias

The job card information window shown in Figure 4-21 is displayed. The panel contains the data the we entered previously. Press the Enter key to edit the generated job.

```
Edit Generated Job      Y      (Yes/No)

Build job in Dataset    PAOLR5.JCL.CNTL
Member                  CATLOGS

Job Cards:
==> //PAOLR5C JOB (999,POK),HUBBARD,
==> //          NOTIFY=&SYSUID,CLASS=A,MSGCLASS=X,
==> //          MSGLEVEL=(1,1),REGION=0M
==> //          /*

Press ENTER to process or PF3 to Cancel
```

Figure 4-21 Enter jobholder information

The generated JCL to merge the catalog entries is shown in Example 4-1 on page 77. Because we have two alias entries to merge, we need to edit the IDCAMS input cards to add the second alias. The edited control cards are shown in bold in Example 4-1 on page 77.

Note: The generated job contains a step to automatically stop the DB2 subsystem that we used when we collected the configuration information. We need to be sure to include the JOBPARM JCL card to ensure that the job runs on the LPAR where that DB2 subsystem is active.

Note: Since this is a data sharing group, we must manually shut down all other members of the group running on other LPARs or the job will fail. The DB2 table space data sets can not be in use while the mergecat is running.

Example 4-1 JCL and control cards to merge catalog alias

```
//PAOLOR5C JOB (999,POK),HUBBARD,
//          NOTIFY=&SYSUID,CLASS=A,MSGCLASS=X,
//          MSGLEVEL=(1,1),REGION=0M
//*
/*JOBPARM SYSAFF=SC53
//*
//*
//** * * * * *
//*
/* STEP:      DB2STOP
//*
/* DESC:      THIS STEP WILL STOP DB2 SUBSYSTEM D8G1
//*
//** * * * * *
//*
//DB2STOP EXEC PGM=ARY#SDB2,COND=(4,LT),PARM=(D8G1,STOP)
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1MO.SARYLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD SYSOUT=*
//*
//** * * * * *
//*
/* STEP:      IDCAMS
//*
/* DESC:      THIS STEP WILL EXECUTE THE IDCAMS JOB.
//*
//** * * * * *
//*
//*****
/* IDCAMS JOB STEP
//*****
//IDCAMS EXEC PGM=IDCAMS,REGION=0M,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

ALTER UCAT.TOTDDL LOCK
IF MAXCC > 0 THEN CANCEL

ALTER UCAT.DB8GDATA LOCK

IF MAXCC > 0 THEN CANCEL

REPRO -
  INDATASET(UCAT.TOTDDL) -
  OUTDATASET(UCAT.DB8GDATA) -
  LEVEL(DB8GU) -
```

```

MERGECAT

IF MAXCC > 0 THEN CANCEL

REPRO -
  INDATASET(UCAT.TOTDDL) -
  OUTDATASET(UCAT.DB8GDATA) -
  LEVEL(DB8GUS) -
  MERGECAT

IF MAXCC > 0 THEN CANCEL

DELETE DB8GU ALIAS
DEFINE ALIAS (NAME(DB8GU) REL(UCAT.DB8GDATA))

ALTER UCAT.TOTDDL UNLOCK
ALTER UCAT.DB8GDATA UNLOCK

/** * * * * *
/**
/** STEP:      DB2START
/**
/** DESC:      THIS STEP WILL START DB2 SUBSYSTEM D8G1
/**
/** * * * * *
/**
/**DB2START EXEC PGM=ARY#SDB2,COND=(4,LT),PARM=(D8G1,START)
/**STEPLIB DD DISP=SHR,DSN=ARY.V2R1MO.SARYLOAD
/**DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
/**SYSPRINT DD SYSOUT=*
/**
/**
/**
/** * * * * *
/**
/** STEP:      UPDSTAT
/**
/** DESC:      THIS STEP WILL UPDATE THE ANALYSIS STATUS INDICATOR.
/**
/** * * * * *
/**
/*******
/*** UPDATE ANALYSIS STATUS INDICATOR *
/*******
/**UPDSTAT EXEC PGM=ARY#SUPD,COND=(4,LT),PARM=(D8G1)
/**STEPLIB DD DISP=SHR,DSN=ARY.V2R1MO.SARYLOAD
/**ARYMOVER DD DISP=SHR,DSN=ARY.SG7606.MOVDATA
/**SYSPRINT DD SYSOUT=*
/**
/******* Bottom of Data *****

```

4.3.6 Optimal subsystem configuration

Figure 4-22 on page 79 shows the subsystem setup information window after all of our jobs have completed successfully.

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/15 15:36:55
Option ===>                               Scroll ==> CSR
-----
Subsystem: D8G1   Active: Yes      Datasharing: Yes
Date of Last Analysis: 02/15/2008  Analysis Recommended: Y
Message: Subsystem configuration is optimal.
-----
Row 1 of 53  +
New MVS User Catalogs to be used by this DB2 Subsystem
Log/BSDS Cat1  UCAT.DB8GLOGS                               Volume DB8GL1
DB2 Data Cat1  UCAT.DB8GDATA                               Volume DB8GD1
Line Cmds: (C-Create, A-Add Alias, D-Dataset Disp, U-Update, V-View Alias)

Existing MVS User Catalogs used by this DB2 Subsystem
Data           UCAT.DB8GDATA                               Volume DB8GD1
Log            UCAT.DB8GLOGS                               Volume DB8GL1
Line Cmds: (D-Dataset Display, V-View Aliases)

Storage Copy Pools
Data Copy Pool DSN$DB8G$DB
Log Copy Pool  DSN$DB8G$LG
Backup Pool    DB8GCPB
Line Cmds: (V-Volume List)

Boot Strap Datasets
D8G1 - BSDS 1  DB8GL.D8G1.BSDS01                           Volume DB8GL1
D8G1 - BSDS 2  DB8GL.D8G1.BSDS02                           Volume DB8GL2
D8G2 - BSDS 1  DB8GL.D8G2.BSDS01                           Volume DB8GL1
D8G2 - BSDS 2  DB8GL.D8G2.BSDS02                           Volume DB8GL2
Line Cmds: (R-Rename BSDS, M-Move BSDS)

Active Log Datasets
D8G1 - Log 1   DB8GL.D8G1.LOGCOPY1.DS01                     Volume DB8GL1
D8G1 - Log 1   DB8GL.D8G1.LOGCOPY1.DS02                     Volume DB8GL1
D8G1 - Log 1   DB8GL.D8G1.LOGCOPY1.DS03                     Volume DB8GL1
D8G1 - Log 2   DB8GL.D8G1.LOGCOPY2.DS01                     Volume DB8GL2
D8G1 - Log 2   DB8GL.D8G1.LOGCOPY2.DS02                     Volume DB8GL2
D8G1 - Log 2   DB8GL.D8G1.LOGCOPY2.DS03                     Volume DB8GL2
D8G2 - Log 1   DB8GL.D8G2.LOGCOPY1.DS01                     Volume DB8GL1
D8G2 - Log 1   DB8GL.D8G2.LOGCOPY1.DS02                     Volume DB8GL1
D8G2 - Log 1   DB8GL.D8G2.LOGCOPY1.DS03                     Volume DB8GL1
D8G2 - Log 2   DB8GL.D8G2.LOGCOPY2.DS01                     Volume DB8GL2
D8G2 - Log 2   DB8GL.D8G2.LOGCOPY2.DS02                     Volume DB8GL2
D8G2 - Log 2   DB8GL.D8G2.LOGCOPY2.DS03                     Volume DB8GL2
Line Cmds: (R-Rename Log, M-Move Log)

Alias used with associated MVS User Catalogs
DB8GL          UCAT.DB8GLOGS          Log
DB8GU          UCAT.DB8GDATA          Data
DB8GUS         UCAT.DB8GDATA          Data
Line Cmds: (D-Dataset Display, M-Merge catalog entries, R-Rename Alias)

Volumes used by this DB2 Subsystem
Volume  Data  DataCat  ActLog ActCat  ArcLog ArcCat  Other  Pool
DB8GD1    Yes  Yes      No    No      No    No      No    Yes
DB8GD2    Yes  No       No    No      No    No      No    Yes
DB8GL1    No   No       Yes   Yes     No    No      No    Yes
DB8GL2    No   No       Yes   No      No    No      No    Yes
DB8GS1    Yes  No       No    No      No    No      No    Yes
Line Cmds: (D-Dataset Display, M-Move all Datasets on Volume)

***** Bottom of Data *****

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 4-22 Subsystem configuration completed

Take special note of the following changes since our initial analysis:

- ▶ The message in the header now reads Subsystem configuration is optimal. This was our goal when we started.
- ▶ The volumes TOTDDJ, TOTDDM, and TOTDDZ are no longer in the list of volumes used by this DB2 data sharing group. Those volumes no longer contain BSDS data sets, active log data sets, or DB2 table space data sets associated with this DB2 data sharing group.
- ▶ DB2 BSDS and active log data sets are on their own set of volumes along with the associated ICF user catalog.
- ▶ DB2 table space data sets are on their own set of volumes along with the associated ICF user catalog.
- ▶ There are no other data sets in the list of volumes used by this data sharing group.

Note: It is extremely important that the DB2 subsystem be optimally configured before using system level backups. Using system level backups of an improperly configured subsystem or data sharing group can cause unpredictable results.

4.4 Profiles

IBM DB2 Recovery Expert for z/OS System Backup and Restore Services (SBRS) allows you to back up and restore entire subsystems, individual objects, or groups of objects. *Profiles* are used to store the options selected for a particular backup or restore process. DB2 Recovery Expert for z/OS supports and manages the following types of profiles:

- ▶ Backup profiles
- ▶ Object recovery profiles
- ▶ Disaster recovery profiles

Note: Recovery Expert stores the profile specifications described in this chapter in the repository described at “System Backup and Restore Services VSAM repository” on page 8.

4.4.1 Backup profiles

Backup profiles contain information that is passed to System Backup and Restore Services and incorporated into the backup job when it is built. Using System Backup and Restore Services's online interface, you create a backup profile to specify the source volumes to be backed up for a subsystem and its associated target units. In addition, you set other backup options such as the backup type and the number of generations to keep in the profile. Backup profiles are reusable and editable, and are created on a per-subsystem basis. After profile setup has been run, you cannot change the subsystem ID or type of backup in a profile, but you can change other settings as well as redefine the source and target volumes. You can easily rename and delete backup profiles using line commands.

Profile setup is a validation process performed by System Backup and Restore Services before a backup of a subsystem can be taken. This process authenticates the volumes for the subsystem and performs other validations to ensure that the backup can proceed and that the resulting backup will be usable.

A backup profile *must* successfully complete profile setup before a backup can be generated. Once a profile has been set up, it does not need to be set up again unless changes are made

to the source or target volume configuration or unless System Backup and Restore Services detects certain errors while building a backup job. In this case, the profile is flagged as Set up Needed on the Update Backup Profile panel. When building a backup job, if profile setup is needed, it is included as the first job step in the JCL. If you request it, System Backup and Restore Services can build only the profile setup syntax in the JCL. In this case, no backup is taken. This entire process is described in more detail in 5.2, “System level backup and restore using Recover Expert ISPF-based System Backup and Restore Services” on page 99.

4.4.2 Object recovery profiles

The online interface allows you to easily create object profiles to define groups of objects that you might need to recover together. When you build a recovery job from an object profile, System Backup and Restore Services will analyze all objects in the profile and generate the most appropriate recovery method for each object.

If you have used System Backup and Restore Services to take system level backups and enabled the object restore option, System Backup and Restore Services will be able to restore the objects directly from the system level backup.

This may be used in place of an image copy for each table space and or index.

Note: The index must be defined as “COPY YES” in order for a system level backup to be used for recovery.

If there are no system level backups, System Backup and Restore Services will generate recovery jobs using the appropriate DB2 image copies and apply logs when required.

You can recover the objects to a current, to a copy, or to a quiesce point. Whichever you choose, System Backup and Restore Services selects the appropriate recovery methods to bring the objects to the specified point in time.

Object profiles are reusable and editable, and are created on a per-subsystem basis. This means that a given object profile applies to objects on one and only one DB2 subsystem or data sharing group.

Working with object profiles is described in 6.1, “Object recovery with Recovery Expert under DB2 V8” on page 126. Additional information is also available in *the IBM DB2 Recovery Expert for z/OS User's Guide*, SC19-1222-00.

4.4.3 Disaster recovery profiles

In case of a disaster that renders a DB2 subsystem unusable, you can use DB2 Recovery Expert for z/OS to restore the subsystem at a remote site if you have implemented the Disaster Recovery feature. System Backup and Restore Services allows you to configure your disaster recovery strategy to restore from either DB2 image copies or system level backups. See Part 3, “Managing disaster recovery” on page 149.

To ensure an accurate recovery, you need to build several disaster recovery profiles and ensure that the jobs produced from each profile are executed on a regular basis.

System Backup and Restore Services provides an easy-to-use ISPF interface to build disaster recovery (DR) profiles. You do not have to know the names of the logs, BSDS, or DB2 catalog data sets.

Creating, building, and maintaining disaster recovery profiles is discussed in Chapter 5, “Local system level backup and restore” on page 83.

Local system level backup and restore

In this chapter we show the steps needed to accomplish a successful system level backup and restore. First we describe the DB2 BACKUP SYSTEM and Recovery Expert GUI supported restore system steps. Then we show how Recovery Expert performs the system level backup and restore functions from the ISPF interface. Finally, we summarize the different options and provide a comparison between the two methods.

This chapter contains the following sections:

- ▶ “System level backup using DB2 and Recovery Expert GUI to perform the restore” on page 84
- ▶ “System level backup and restore using Recover Expert ISPF-based System Backup and Restore Services” on page 99
- ▶ “ISPF over GUI system level backup and recovery advantages” on page 121

5.1 System level backup using DB2 and Recovery Expert GUI to perform the restore

As of DB2 V8, two new utilities, BACKUP SYSTEM and RESTORE SYSTEM, are provided to utilize and exploit DFSMSHsm™ to take advantage of the fast replication support that was introduced in z/OS V1R5.

BACKUP SYSTEM is a method used for creating a system level point-in-time (PIT) backup, and the RESTORE SYSTEM utility is used to recover the entire system.

Use the RESTORE SYSTEM utility only when you want to recover a subsystem or data sharing group to an arbitrary point in time including the most current one (PK51979). The utility restores only the *database copy pool* 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 or the end of the log. If you want to restore a full system (including the logs) to the point at which a backup was taken, do not use the RESTORE SYSTEM utility. Use the native 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.

In DB2 9 these utilities were enhanced to use new functions available with z/OS V1R8 DFSMSHsm. Appendix C, "Storage and DB2 functions" on page 209, provides an overview of the advanced storage (disk and tape) and DB2 functions that are utilized for backup and restore of DB2 subsystems and the recovery of DB2 objects from volume dumps. The disk functions are provided by the family of Enterprise Storage Servers and follow-on devices.

In the next sections we detail the following seven steps for the backup and restore of a DB2 subsystem or data sharing group:

- ▶ Step 1 - Run a BACKUP SYSTEM utility.
- ▶ Step 2 - Build RESTORE SYSTEM jobs.
- ▶ Step 3 - Prepare the environment for RESTORE SYSTEM.
- ▶ Step 4 - Start DB2.
- ▶ Step 5 - Verify that ICF catalogs volumes are not in use by the catalog.
- ▶ Step 7 - STOP and reSTART DB2.

Step 1 - Run a BACKUP SYSTEM utility

The DB2 online BACKUP SYSTEM utility invokes z/OS DFSMSHsm (Version 1 Release 5 or later) to copy the volumes on which the DB2 data and log information resides for either a DB2 subsystem or a data sharing group. You can use BACKUP SYSTEM to copy all data for a single application, for example, when DB2 is the database server for a resource-planning solution. All the data sets that you want to copy must be Storage Management Subsystem (SMS) managed data sets. Subsequently, you can run the RESTORE SYSTEM utility to recover the data.

In a data sharing environment, if any failed or abnormally quiesced members exist, the BACKUP SYSTEM request fails. The BACKUP SYSTEM utility uses copy pools, which are constructs in z/OS DFSMSHsm V1R5 and later. Each DB2 subsystem must have two copy pools, one for DB2 data (user and catalog/directory) and one for log and bootstrap data sets.

BACKUP SYSTEM copies the volumes that are associated with these copy pools at the time of the copy. The output for BACKUP SYSTEM is the copy of the volumes on which the DB2 data and log information resides. The BACKUP SYSTEM history is recorded in the bootstrap data sets (BSDSs) in the BACKUP SYSTEM UTILITY HISTORY section. Up to 50 backup versions can be stored.

Note: In a data sharing environment, the submitting member records the version in its BSDS and also in the SCA. If you want to see the latest backup system entry, run DSNJU004 with the GROUP DD statement pointing to the BSDS data sets. This will merge all the BSDS information from all the members.

During the system level backup, DB2 will record a recovery base log point (RBLP) in the header page of DBD01. The RBLP is identified as the most recent system checkpoint prior to a backup log point, and is the point at which DB2 starts scanning logs during a RESTORE SYSTEM recovery operation.

Note: All the BACKUP SYSTEM versions are tracked in the BSDS, whereas the RBLP in DBD01 is updated (overwritten) every time that the BACKUP SYSTEM is run.

With DB2 V8 there are two options available when running the BACKUP SYSTEM utility:

- ▶ **FULL** - This option allows you to copy both the database copy pool and the log copy pool. This is also the default.
- ▶ **DATA ONLY** - This option allows you to copy the database copy pool only. The log copy pool is not copied.

Execute the BACKUP SYSTEM (DB2 Utility) outside of the Recovery Expert product. Recovery Expert does not build a BACKUP SYSTEM job stream in the GUI interface. Example 5-1 shows the JCL for BACKUP SYSTEM.

Example 5-1 BACKUP SYSTEM JCL

```
//BACKUP EXEC DSNUPROC,SYSTEM=D9CG,
//          UID='SYSBACK'
//STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
//          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSIN   DD *
          BACKUP SYSTEM FULL
```

Example 5-2 shows the job output of the BACKUP SYSTEM.

Example 5-2 FAST REPLICATION output in DFSMSHsm for the DB COPYPOOL

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB9C$DB, AT 17:07:44 ON 2008/02/12,
  TOKEN=X'C4F9C3F1C1F0D91FD1AE7903C1F0D58C8ED3'
ARC0640I ARCFRTM - PAGE 0001      5695-DF175 DFSMSDSS V1R09.0 DATA SET SERVICES      2008.043 17:07
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 IDY(SBOX5S) ODY(SBOX6A) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY'
...
...
ARC0640I ARCFRTM - ADR012I (SCH)-DSSU (01), 2008.043 17:07:45 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC1805I THE FOLLOWING 00004 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL DSN$DB9C$DB
ARC1805I (CONT.) SBOX5S
ARC1805I (CONT.) SBOX5T
ARC1805I (CONT.) SBOX5U
ARC1805I (CONT.) SBOX5V
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB9C$DB, AT 17:07:45 ON 2008/02/12, FUNCTION RC=0000,
  MAXIMUM VOLUME RC=0000
```

Example 5-3 shows the output of the FAST REPLICATION tasks performed by DFSMSHsm for the DB and LOG COPYPOOLS.

Example 5-3 FAST REPLICATION output in DFSMSHsm for the LOG COPYPOOL

```
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB9C$LG, AT 17:07:45 ON 2008/02/12,
  TOKEN=X'C4F9C3F1C1F0D91FD1AE7903C1F0D58C8ED3'
ARC0640I ARCFRTM - PAGE 0001      5695-DF175 DFSMSDSS V1R09.0 DATA SET SERVICES      2008.043 17:07
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 IDY(SBOX5Q) ODY(SBOX6G) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
..
..
ARC0640I ARCFRTM - ADR241I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000005:0000 AND ENDING AT 000016:0014 IS OVERLAID
ARC0640I ARCFRTM - ADR806I (003)-TOMI (02), VOLUME SBOX5R WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADR006I (003)-STEND(02), 2008.043 17:07:46 EXECUTION ENDS
ARC0640I ARCFRTM - ADR013I (003)-CLTSK(01), 2008.043 17:07:54 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADR006I (019)-STEND(01), 2008.043 17:07:46 EXECUTION BEGINS
..
..
ARC1805I (CONT.) SBOX5L
ARC1805I (CONT.) SBOX5M
ARC1805I (CONT.) SBOX5N
ARC1805I (CONT.) SBOX5O
ARC1805I (CONT.) SBOX5P
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB9C$LG, AT 17:07:55 ON 2008/02/12, FUNCTION RC=0000,
  MAXIMUM VOLUME RC=0000
```

Step 2 - Build RESTORE SYSTEM jobs

Once one or more BACKUP SYSTEM utilities have successfully executed, the Recovery Expert Agent will be able to read these entries from BSDS, and you can use Recovery Expert's GUI interface to build the RESTORE SYSTEM jobs.

From the GUI's launchpad select the Recovery function window. See Figure 5-1.

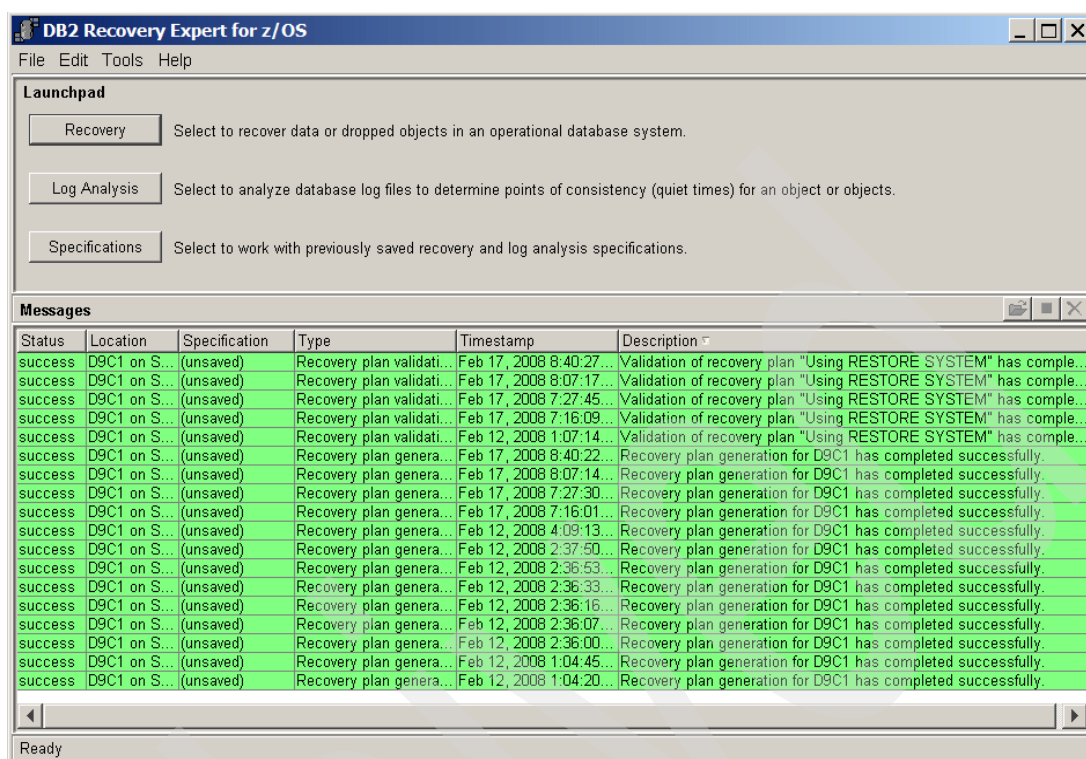


Figure 5-1 Recovery launchpad - recovery function

Next you can select the DB2 subsystem that you want to recover. The options are:

- ▶ z/OS subsystems
- ▶ z/OS data sharing groups
- ▶ z/OS systems
- ▶ z/OS sysplexes

In our example (Figure 5-2) we connect to the DB2 subsystem that we want to recover, from the z/OS Data Sharing Groups selection list.

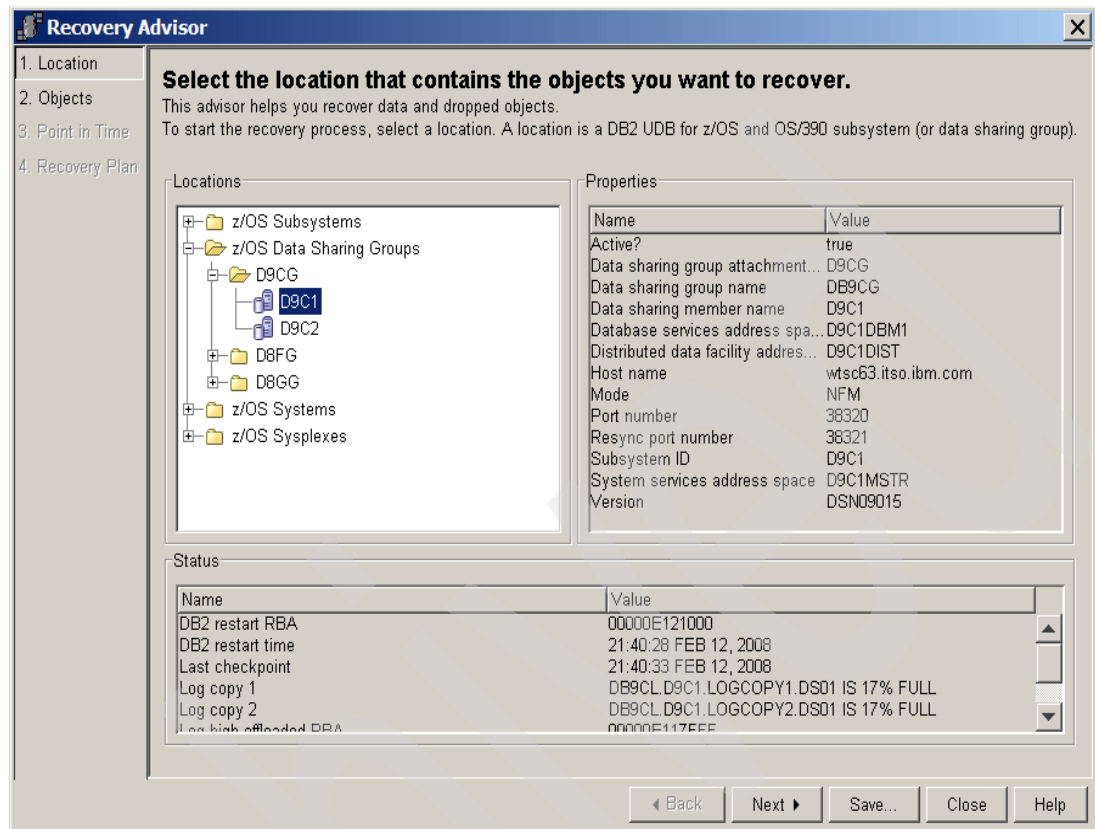


Figure 5-2 Recovery selection hierarchy

Member D9C1 is selected. The DB2 subsystem information, such as current restart RBA, last checkpoint, the current active log information, and so on, is shown in the Status window.

In Figure 5-3 we select member D9C1 and connect to the DB2 subsystem.

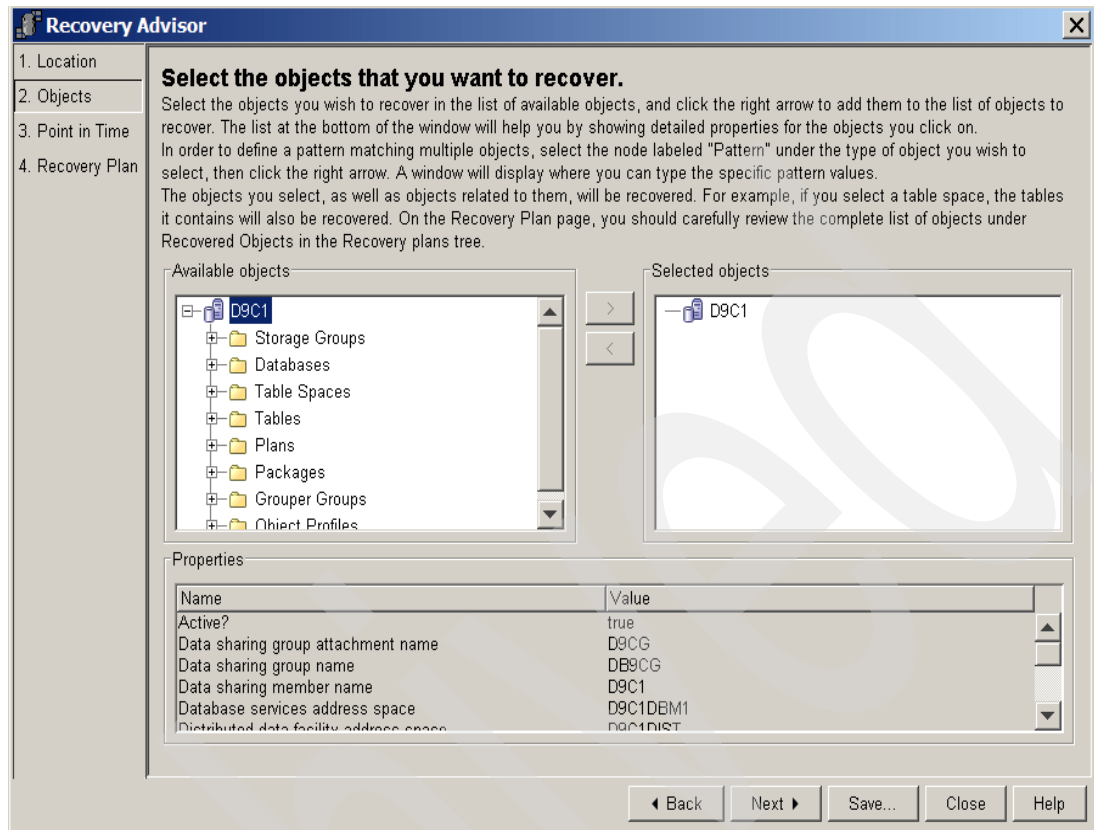


Figure 5-3 Recovery selection - DB2 subsystem level

You now select the **Timestamp** as the point in time to which you want to recover. The details are shown in Figure 5-4.

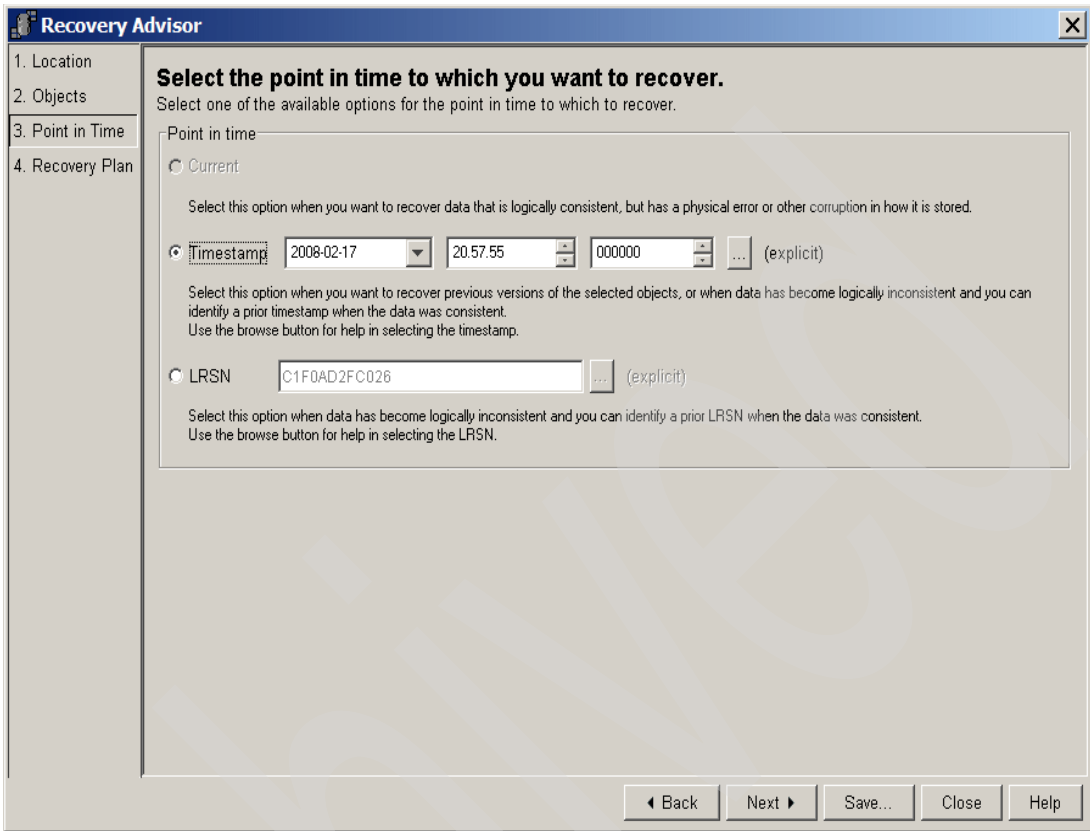


Figure 5-4 Point-in-time selection - using explicit function

Ensure that you select the *explicit* option for Recovery Expert to read the latest backup system information from the BSDS as your selection choice. You can run DSNJU004 to list the contents of the BSDS as shown in Example 5-4.

Example 5-4 BSDS BACKUP SYSTEM UTILITY HISTORY output from DSNJU004

BACKUP SYSTEM UTILITY HISTORY							
SUBSYSTEM ID D9C1							
20:11:08 FEBRUARY 11, 2008							
DATA	START STCK	LOG	RBLP	DATA COMPLETE	DATA/LOG COMPLETE	DATE	LOCATION NAME
				LRSN	LTIME		
C1EBB66942D63C09	C1EBB66C54A45D43	C1EA802CE63B	C1EBB675C2D2	2008/02/08	15:06:02	DB9C	
	TOKEN = C4F9C3F1C1EBB66942D63C09C1EA802CE63B						
C1EA8EA7799EE44E	C1EA8EA9FB0E8443	C1EA802CE63B	C1EA8EB4F110	2008/02/07	17:02:51	DB9C	
	TOKEN = C4F9C3F1C1EA8EA7799EE44EC1EA802CE63B						

You have to choose **Recovery History Events** (Figure 5-5) to view the time stamp that is recorded in the BSDS' BACKUP SYSTEM UTILITY HISTORY section.

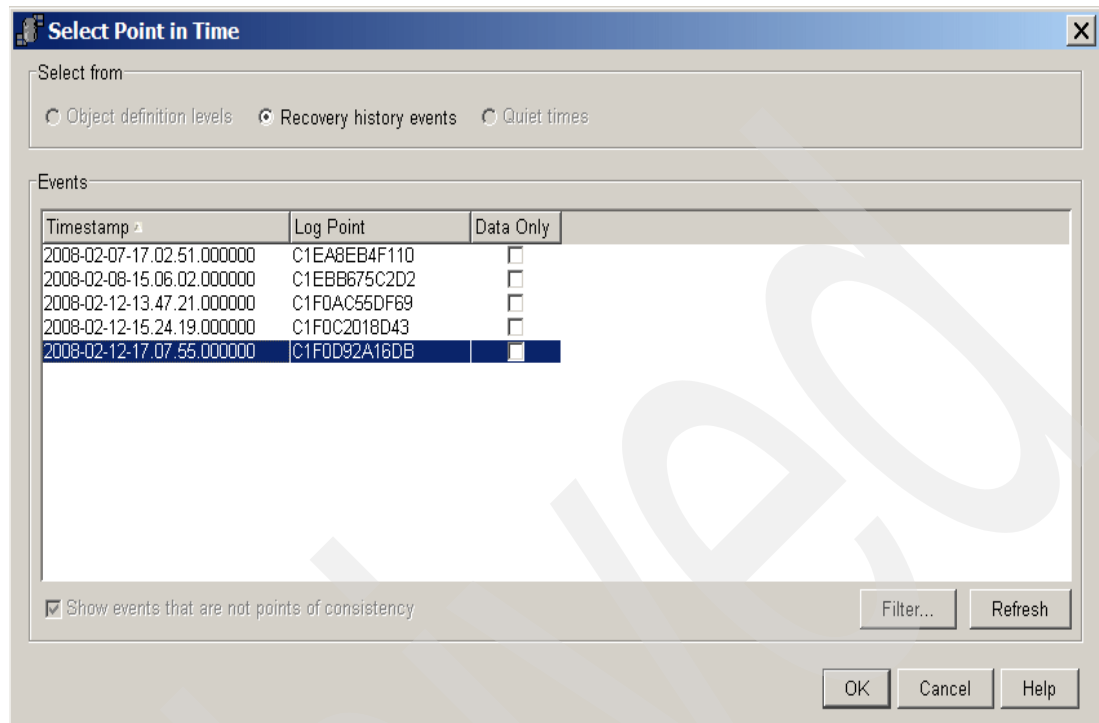


Figure 5-5 Time stamp selection for *RESTORE SYSTEM*

Once you select the time stamp or Logpoint (see Figure 5-5), the associated RBA will be populated as your RBA and will be the time stamp (used for point-in-time system level recovery) to restore to. In this same selection you can choose whether you want to recover the data *and the logs* or just the data.

Note: The DB2 RESTORE utility only restores the data.

Next you have the choice to generate the JCL and jobs to restore the subsystem. See Figure 5-6 for the setting up of the generation.

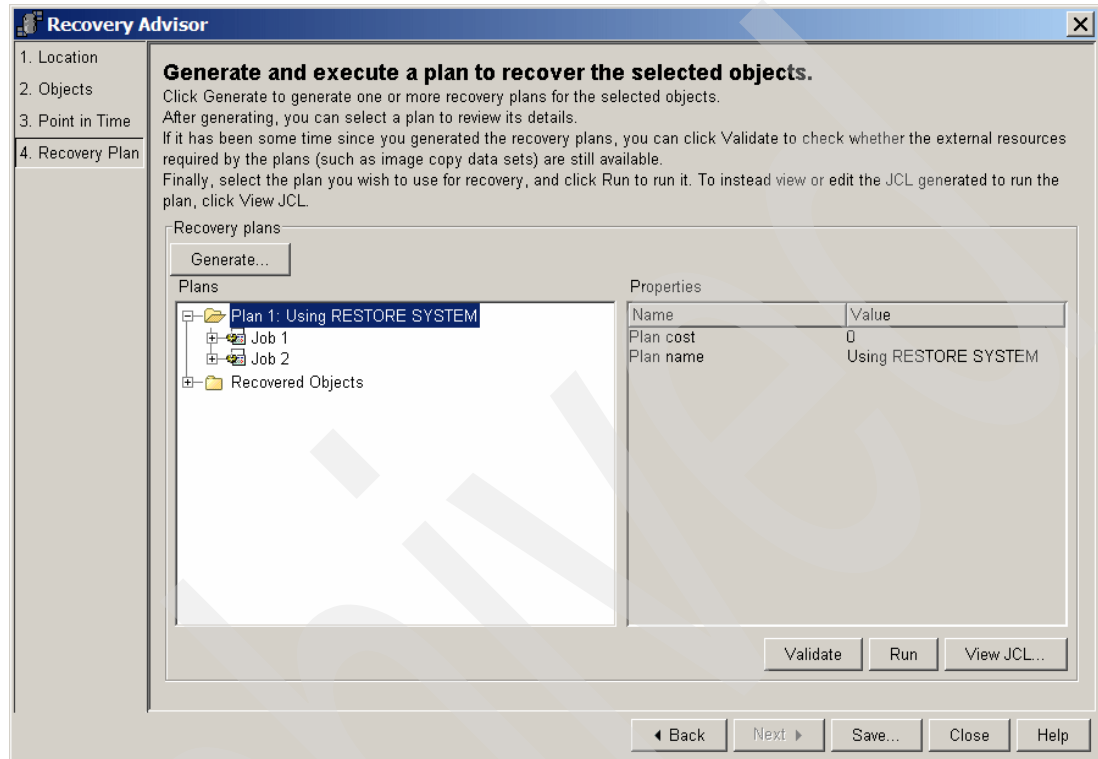


Figure 5-6 Recovery plans generation panel

Figure 5-7 shows the generated JCL. Here you also have the choice to edit the JCL, like the JOBNAME, if needed.

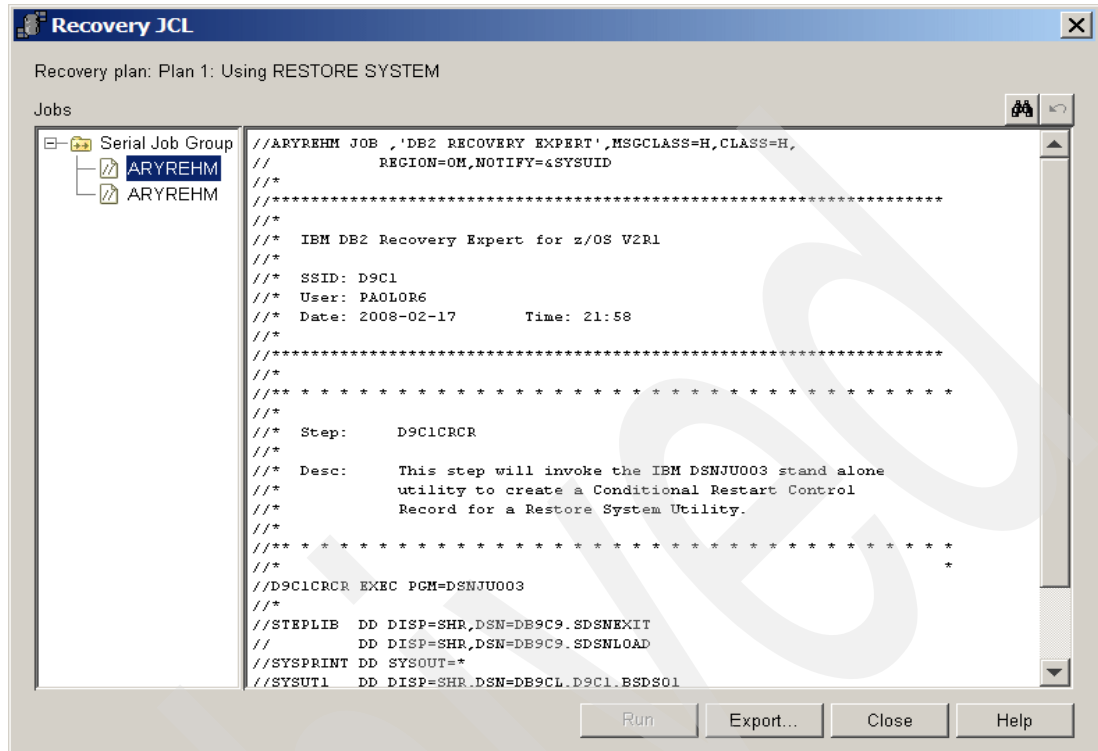


Figure 5-7 View generated JCL before exporting to a z/OS file

Figure 5-8 shows the window where you will be prompted to export the JCL to a z/OS data set or local file on your workstation. You have to export both jobs (JOB1 and JOB2) separately.

Note: The output data set name that you want to export the generated JCL into must not be in use. Recovery Expert requires exclusive control (DISP=OLD) of the data set.

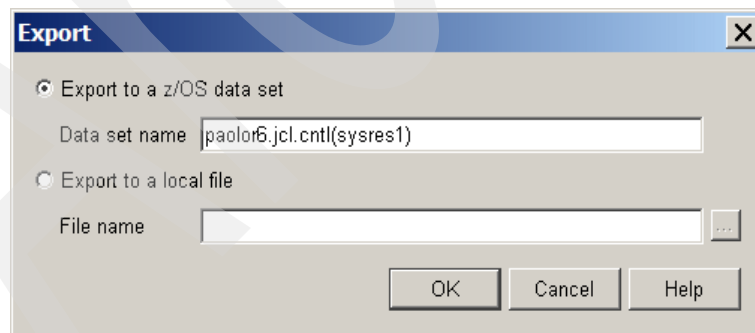


Figure 5-8 Export window to send JCL to z/OS PDS

Note: In a Data Sharing environment you have to run "JOB1" for each member. This is the job that will update the BSDS with a SYSPITR CRCR entry. Currently, Recovery Expert (in the GUI environment only) generates only one job. You have to clone this JCL in the z/OS data set for each data sharing member.

Step 3 - Prepare the environment for RESTORE SYSTEM

There are six steps left to perform a successful system level restore from the BACKUP SYSTEM taken in step 1.

Using the DB2 RESTORE SYSTEM utility allows you to recover a subsystem or data sharing group to an arbitrary point in time. The utility restores only the database copy pool and then applies logs until it reaches current or 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 (in step 5).

The DB2 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 will manage 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).

If you want to restore a full DB2 system to the point at which a backup was taken, then the RESTORE SYSTEM utility should be considered as only as the first step. You additionally need to recover the log cypool using native fast replication functions. Another option is to use Recovery Expert ISPF-based system level recovery as explained later.

You now have created the RESTORE SYSTEM JCL via the GUI interface and saved the RESTORE SYSTEM JCL into a PDS (step 2).

Next, stop the DB2 subsystem. In a data sharing group, you have to stop all members of the group. In our example we stopped both D9C1 and D9C2 DB2 subsystems.

Step 3.1 - Create the SYSPITR CRCR for each DB2 subsystem

Recovery Expert generated two jobs. One is for creating a SYSPITR CRCR (conditional restart control record) called JOB1, which you saved into a PDS in step 2. The other job, JOB2, is the actual RESTORE SYSTEM job that you will run in step 6.

The Recovery Expert ISPF-based system restore (as explained later) creates the CRCR jobs for all members.

JOB1 is a DB2 DSNJU003 utility that you need to run for each of the DB2 members and that specifies a log truncation point that corresponds to the selected point in time. See Example 5-5.

In our example we run "JOB1" for D9C2 and D9C2.

Example 5-5 DSNJU003 - Recovery Expert JOB1

```
//D9C1CRCR EXEC PGM=DSNJU003
//*
//STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS01
//SYSUT2 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS02
//SYSIN DD *
CRESTART CREATE,SYSPITR=C1F0D92A16DB,FORWARD=YES,BACKOUT=YES
```

The output is listed in Example 5-6.

Example 5-6 Output of "JOB1" - create a conditional restart control record in the BSDS

```
CRESTART CREATE,SYSPIR=C1F0D92A16DB, FORWARD=YES, BACKOUT=YES
DSNJ408I DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00000E114090, TIME = 21:51:45 FEBRUARY 12, 2008
DSNJ411I DSNRJRCR CRESTART CREATE FOR CRCRID = 0003, DDNAME = SYSUT1
DSNJ408I DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00000E114090, TIME = 21:51:45 FEBRUARY 12, 2008
DSNJ411I DSNRJRCR CRESTART CREATE FOR CRCRID = 0003, DDNAME = SYSUT2
DSNJ225I CRESTART OPERATION COMPLETED SUCCESSFULLY
DSNJ200I DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

Step 3.2 - Delete all CF structures

This step is only required for data sharing. It is very important to perform this step, as not doing so will cause a lot of unnecessary problem determination when the you try to restart DB2.

Note: The commands to display and delete the XCF structures may require additional authority. Depending on your installation, this function may have to be performed by your z/OS systems programmer or someone with the necessary authority.

To verify the two coupling facility structure names and to check their current status, use the command "D XCF,STRUCTURE" to display them. See Example 5-7 for the command and output.

Example 5-7 Displaying the XCF structures

D XCF,STRUCTURE

IXC359I	14.41.01	DISPLAY XCF 454		
STRNAME		ALLOCATION TIME	STATUS	TYPE
DB9CG_LOCK1		02/12/2008 17:18:45	ALLOCATED	LOCK
DB9CG_SCA		02/12/2008 17:18:44	ALLOCATED	LIST

In our example we have the DB2 lock structure (LOCK1) and the shared communications area (SCA) to be rebuilt.

Example 5-8 shows the command and related output to delete the lock structure.

Example 5-8 Deleting the lock structure

```
SETXCF FORCE,STRUCTURE,STRNAME=DB9CG_LOCK1
IXC579I NORMAL DEALLOCATION FOR STRUCTURE DB9CG_LOCK1 IN 064
          COUPLING FACILITY 002094.IBM.02.00000002991E
          PARTITION: OF      CPCID: 00
HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: C1F0D586 6171CEC6
INFO116: 132C2081 01 2800 00000011
TRACE THREAD: 001079E8.
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE 065
DB9CG_LOCK1 WAS COMPLETED:
STRUCTURE DELETED BUT ALSO RESULTED IN DELETED CONNECTION(S)
```

Example 5-9 shows the command and related output to delete the SCA structure.

Example 5-9 Deleting the SCA structure

```
SETXCF FORCE,STRUCTURE,STRNAME=DB9CG_SCA
IXC579I NORMAL DEALLOCATION FOR STRUCTURE DB9CG_SCA IN 067
      COUPLING FACILITY 002094.IBM.02.00000002991E
      PARTITION: 0F    CPCID: 00

HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: C1F0D585 326CFE84
INF0116: 132C2081 01 2800 00000010
TRACE THREAD: 001079F7.
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE 068
DB9CG_SCA WAS COMPLETED:
STRUCTURE WAS DELETED
```

Once both structures have been deleted, you can issue another display to verify that the cleanup took place as in Example 5-10.

Example 5-10 Display to verify delete

```
D XCF,STR
IXC359I 17.18.01 DISPLAY XCF 070
STRNAME      ALLOCATION TIME  STATUS      TYPE
DB9CG_LOCK1  --      --    NOT ALLOCATED
DB9CG_SCA    --      --    NOT ALLOCATED
```

Step 4 - Start DB2

The next step is to start *all* the DB2 members. In our example members D9C1 and D9C2 are started.

When you start DB2, three events occur:

- ▶ You have to acknowledge the conditional restart record truncation. Reply Y for DB2 to continue. DB2, in ACCESS(MAINT), prevents access by users until the recovery is complete. During restarting, in-flight and in-abort URs are backed out, in-commit are committed, and in-doubts are noted on the console log. See Example 5-11.

Example 5-11 Acknowledgement of the restart record truncation

```
DSNY001I -D9C2 SUBSYSTEM STARTING
DSNJ127I -D9C2 SYSTEM TIMESTAMP FOR BSDS= 08.043 17:14:24.32
142 DSNJ245I -D9C2 CONDITIONAL RESTART RECORD INDICATES TRUNCATION
AT LRSN C1F0D92A16DB.  REPLY Y TO CONTINUE, N TO CANCEL
-D9C1 START DB2
```

- ▶ DB2 is restarted with ACCESS(MAINT). See Example 5-12.

Example 5-12 DB2 restarted with ACCESS(MAINT)

```
*DSNR050I -D9C1 DSNRRPRC DB2 STARTED IN SYSTEM RECOVER PENDING MODE
DSNR002I -D9C1 RESTART COMPLETED
DSNY014I -D9C1 DSNYSTRT DB2 WAS STARTED WITH ACCESS(MAINT)
```

- The XCF structures are rebuilt, as shown in Example 5-13.

Example 5-13 The SCA structure being rebuild

```
IXC582I STRUCTURE DB9CG_SCA ALLOCATED BY SIZE/RATIOS. 080
  PHYSICAL STRUCTURE VERSION: C1F0DB95 26B4AAD6
  STRUCTURE TYPE:                LIST
  CFNAME:                        CF1
  ALLOCATION SIZE:                 8192 K
  POLICY SIZE:                   49152 K
  POLICY INITSIZE:               8192 K
  POLICY MINSIZE:                 0 K
  IXLCONN STRSIZE:               0 K
  ENTRY COUNT:                   6464
  ELEMENT COUNT:                 12607
  ENTRY:ELEMENT RATIO:           1 :    2
  ALLOCATION SIZE IS WITHIN CFRM POLICY DEFINITIONS
IXL014I IXLCONN REQUEST FOR STRUCTURE DB9CG_SCA 081
```

Step 5 - Verify that ICF catalogs volumes are not in use by the catalog

This is a very critical step for the RESTORE SYSTEM to complete successfully. If any of the volumes are in use by the catalog address space or the LOG and DATA catalogs are still open, the RESTORE SYSTEM (which needs them for the APPLY phase) will not complete, and you may spend a long time troubleshooting the problem.

Note: The ICF catalog for the data must be on a separate volume from the ICF catalog for the logs. Your implementation of your SMS and copy pools will affect this task.

Note: Ensure that the DB2 9 address space for the administrative scheduler is also shut down before you verify that the ICF catalog volumes are down. Here is an example of how to shut down the address space:

```
F xxxxADMT,APPL=SHUTDOWN
```

You need to UNALLOCATE your ICF catalogs. In our example, we have two user catalogs to close: UCAT.DB9CLOGS and UCAT.DB9CDATA. See Example 5-14.

Note: Although the UNALLOCATE command CLOSEs the ICF catalogs, we chose to CLOSE the ICF CATALOGS separately to *verify and ensure* that the ICF catalogs are properly closed.

Example 5-14 Closing the ICF catalogs

```
F CATALOG,CLOSE(UCAT.DB9CLOGS)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
F CATALOG,CLOSE(UCAT.DB9CDATA)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Next you have to unallocate the ICF catalogs, as shown in Example 5-15. This ensures that the ICF catalogs for the DB2 data are not active and are not allocated.

Example 5-15 Unallocating the ICF catalogs

```

F CATALOG,UNALLOCATE(UCAT.DB9CLOGS)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEF196I IGD104I UCAT.DB9CLOGS                                RETAINED,
IEF196I DDNAME=SYS00029
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
F CATALOG,UNALLOCATE(UCAT.DB9CDATA)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEF196I IGD104I UCAT.DB9CDATA                                RETAINED,
IEF196I DDNAME=SYS00030
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED

```

Step 6 - Run RESTORE SYSTEM

This is a critical step. If this step does not complete with success you may not be able to continue. You may also experience unpredictable errors, which will cause you to spend unnecessary time doing problem determination.

Note: The RESTORE SYSTEM utility requires you to have INSTALL SYSADM authority to successfully run the job.

RESTORE SYSTEM uses the point in time that you specified for recovery, as indicated by the SYSPITR CRCR RBA/LSRN (which you created in step 3), and locates the backup system entry in the BSDS that immediately precedes that point in time (RBA/LRSN).

Tip: With PTF UK31997 for APAR PK51979, the RESTORE SYSTEM utility can be used to recover DB2 to the current end of the log, with no log truncation required.

Next, DFSMSHsm functions are used to restore the version of the database copy pool that is also associated with the entry that is retrieved from the BSDS.

DFSMSHsm will analyze the version token passed to it from DB2, and locate the appropriate backup in its copy pool backup. It will next invoke the fast replication functionality to restore all the volumes that belongs to the database copy pool.

Once the database copy pool has been restored, DB2 locates the RBLP from the header page in DBD01, which is the starting point for log forward recovery.

Now RESTORE SYSTEM can start the log apply phase and uses the fast log apply function to recover all objects in parallel from the log, up to the point in time (RBA/LRSN) specified by the conditional restart.

Example 5-16 shows the JCL.

Example 5-16 Example of the RESTORE SYSTEM job

```

//D9C1RSSY EXEC PGM=DSNUTILB,COND=(4,LT),
//          PARM=(D9C1,RSSY)
//*
//STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
//          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*

```

```
//UTPRINT DD SYSOUT=*
//SYSIN DD *
RESTORE SYSTEM
```

Example 5-17 shows the job output.

Example 5-17 Recovery messages in DFSMSHsm log

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR COPY POOL DSN$DB9C$DB, AT 21:37:41 ON 2008/02/12
ARC0640I ARCFRTM - PAGE 0001      5695-DF175 DFSMSDSS V1R09.0 DATA SET SERVICES      2008.043 21:37
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 IDY(SBOX6A) ODY(SBOX5S) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - COPY IDY(SBOX6B) ODY(SBOX5T) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - COPY IDY(SBOX6C) ODY(SBOX5U) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 004 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - COPY IDY(SBOX6D) ODY(SBOX5V) DUMPCOND FR(REQ) PUR ALLX ALLD(*)
ARC0640I ARCFRTM - ADR101I (R/I)-RI01 (01), TASKID 005 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - ADR109I (R/I)-RI01 (01), 2008.043 21:37:41 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED
ARC0640I ARCFRTM - ADR014I (SCH)-DSSU (02),
  2008.043 21:37:41 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
..
..
ARC0401I LARGEST EXTENTS FOR SBOX5V ARE CYLINDERS      9214, TRACKS      138210
ARC0402I VTOC FOR SBOX5V IS 0180 TRACKS(09000 DSCBS), 08836 FREE DSCBS(98% OF TOTAL)
ARC1805I THE FOLLOWING 00004 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION RECOVERY OF COPY POOL DSN$DB9C$DB
ARC1805I (CONT.) SBOX5S
ARC1805I (CONT.) SBOX5T
ARC1805I (CONT.) SBOX5U
ARC1805I (CONT.) SBOX5V
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR COPY POOL DSN$DB9C$DB, AT 21:37:42 ON 2008/02/12, FUNCTION RC=0000
```

Once all the copy pools have been recovered successfully, you can continue to the next step.

Step 7 - STOP and reSTART DB2

You can now stop and restart all the DB2 members. This process puts DB2 back in normal processing mode (from maintenance mode).

5.2 System level backup and restore using Recover Expert ISPF-based System Backup and Restore Services

IBM DB2 Recovery Expert for z/OS System Backup and Restore Services (SBRS) allows you to back up and restore the entire subsystem on any supported DB2 version via the ISPF interface. Currently, the GUI portion of Recovery Expert does not build backup JCL.

In addition to the DB2 V9 and beyond BACKUP SYSTEM and RESTORE SYSTEM utilities, Recovery Expert adds extensive execution steps and validation checking that simplify the backup and restore when running these services.

The following sections document the six steps needed to accomplish a successful Recovery Expert backup and restore at a subsystem or data sharing group level.

- ▶ Step 1 - Create BACKUP and RECOVERY profiles - one-time setup.
- ▶ Step 2 - Build and execute the BACKUP SYSTEM Job.
- ▶ Step 3 - Build the RESTORE SYSTEM jobs.

- Step 4 - Prepare the environment and run the RESTORE.
- Step 5 - Start DB2.
- Step 6 - Run the RESTORE SYSTEM jobs.

Note: There are two parameters in the parmlib: RESET_COPY_PENDING_TS and RESET_COPY_PENDING_IX. If these are set to Y (yes), all copy pending table and index spaces included in the backup will have the COPYP reset. See “Step 17 - Change defaults for backup restore” on page 31.

Step 1 - Create BACKUP and RECOVERY profiles - one-time setup

The preparation of the environment for a system level backup in Recovery Expert is part of the original setup. Once you have defined your system backup profile, you can reuse it. The system backup profile is used for both the system backup and system restore functions. Unlike the GUI interface, the ISPF interface can generate backup JCL and be used for the submission thereof.

The profile setup step is also a validation process performed by System Backup and Restore Services before a backup of a subsystem can be started. This process then authenticates the volumes for the subsystem; checks the locations of the user data, logs, and user catalogs; and performs other validations to ensure that the backup can proceed and that the resulting backup will be usable.

A backup profile *must* successfully complete the profile setup step before a backup can be generated. Once a profile has been set up, it does not need to be set up again unless you want to make changes to the source or target volume configuration or unless System Backup and Restore Services detect certain errors while building a backup job.

In our example, we ignore the warning that non-DB2 data will be backed up and recovered during our system level backup system and restore system. When the DB2 BACKUP SYSTEM and RESTORE SYSTEM were completed in the previous section, we were not aware of this issue. Refer to Chapter 4, “Configuring DB2 for system backup and restore” on page 53, where we used Recovery Expert to fix this problem by moving the non-DB2 data off the DB2 data and log volumes.

We strongly recommend that your SMS environment, copy pool structures, and so on, be set up so that your DB2 data, DB2 logs, and other DB2 libraries be in *separate* ICF user catalogs and separate volumes. Not doing so will result in unpredictable and corrupted data. DB2 requires two copy pools, and within each one the data sets must be in sync with the ICF catalog definitions.

If the ICF catalog for the DB2 data does not reside on a volume in the database copypool, it will not be restored with the DB2 pagesets and will not be in sync with the DB2 pageset data sets after restore.

If the DB2 log data sets are cataloged in the same ICF catalog as DB2 data data sets (and it does not reside in the appropriate log copypool) it will be restored to the point in time of the backup and will no longer reflect the current location of the DB2 log data sets.

Before you can create a backup profile for the system level recovery (and related backup, as in our example), you may need to perform two one-time steps: Complete the product setup (option S) and the perform a DB2 subsystem analysis (option M) from the main menu in Figure 5-9. You only need to do this once. This is mentioned here as a verification step.

```
ARY$MAIN      ----- Recovery Expert for z/OS -----
Option  ===>

User: PAOLOR6 - ARY

-----

  B. System Backup Profiles
  R. System Restore and Offload
  O. Object Recovery Profiles
  D. Disaster Recovery Profiles
  M. DB2 Subsystem Analysis and Setup
  S. Product Setup
  X. Exit

Enter END command to return to ISPF.
```

Figure 5-9 Recovery Expert main menu

In Figure 5-10 you see the default values when you start defining a backup system profile for the first time.

```
ARYPNLO ----- Setup Parameters ----- 2008/02/19 14:56:42
Option ==>

Current User Ind      ==> ARY      (From Startup Clist)
DB2 Control Dataset   ==> ARY.DB2PARMS.CONTROL
(Pre-allocated)

Enter or Update Default Values
Backup Scope          ==> F        (Full/Data)
Backup Generations     ==> 01      (01 - 99)
Offline Generations    ==> 02      (00 - 99)
Backup Method          ==> F        (Bcv/Snap/Flash/Db2)
Issue Log Suspend      ==> N        (Yes/No)
Validate Volumes       ==> Y        (Yes/No)
Backup Repository      ==> Y        (Yes/No)
Work File Unit Device  ==> SYSALLDA (SYSDA, DISK, etc.)

Enter DB2 Subsystem Info:
DB2 Subsystem ID      ==> D9C1     (1-4 Char Subsystem ID)

Valid command selection values are
1: ZPARM, BSDS, and Load Library Information
2: DB2 Plans
```

Figure 5-10 Creating a backup and restore profile for a DB2 system level backup and restore

In Figure 5-11 we update the profile to indicate that this profile is for a full DB2 backup (data and logs) (F) and that DB2 (BACKUP SYSTEM) is the backup method (D). We also make changes for the work file unit device name from SYSDA to SYSALLDA (which is the unit name for our test environment) and provide the DB2 subsystem ID (D9C1).

```

ARYPNLO ----- Setup Parameters ----- 2008/02/19 14:57:25
Option ==>

Current User Ind      ==> ARY      (From Startup Clist)
DB2 Control Dataset   ==> ARY.DB2PARMS.CONTROL
(Pre-allocated)

Enter or Update Default Values
Backup Scope          ==> F        (Full/Data)
Backup Generations     ==> 01      (01 - 99)
Offline Generations    ==> 02      (00 - 99)
Backup Method          ==> D        (Bcv/Snap/Flash/Db2)
Issue Log Suspend      ==> N        (Yes/No)
Validate Volumes       ==> Y        (Yes/No)
Backup Repository      ==> Y        (Yes/No)
Work File Unit Device  ==> SYSALLDA (SYSDA, DISK, etc.)

Enter DB2 Subsystem Info:
DB2 Subsystem ID      ==> D9C1     (1-4 Char Subsystem ID)

Valid command selection values are
1: ZPARM, BSDS, and Load Library Information
2: DB2 Plans

```

Figure 5-11 Updating the backup and restore profile values

Note: Recovery Expert does validate the UNIT name and will report if this is incorrect.

You also need to specify a valid DB2 SSID. This panel also enables you to view and update the DSNZPARM, BSDS information and review ARY's DB2 plan names to use.

In the following panels (Figure 5-12 to Figure 5-15 on page 106) you see the validation analysis that occurred online for the profile that is being created. This information gives you a complete overview of your recovery assets from a single source. Without Recovery Expert you will have to gather the same information from DB2, DFSMSHsm, and ICF catalog interrogation, among others. Recovery Expert stores all this information in the system level backup and Recovery Services repository, a VSAM file.

```

ARY$SSID      ----- Subsystem Setup Information ----- 2008/02/19 15:59:00
Option  ==>                                         Scroll ==> PAGE
-----
Subsystem: D9C1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/19/2008  Analysis Recommended: N
Message: Other non-DB2 data will be backed up and restored.
-----
                                                    Row 1 of 68  +
New MVS User Catalogs to be used by this DB2 Subsystem
  Log/BSDS Cat1                                         Volume
  DB2 Data Cat1                                         Volume
  Line Cmds: (C-Create, A-Add Alias, D-Dataset Disp, U-Update, V-View Alias)

Existing MVS User Catalogs used by this DB2 Subsystem
  Data Other      UCAT.DB9CDATA                        Volume  SB0X5U
  Log             UCAT.DB9CLOGS                       Volume  SB0X5A
  Line Cmds: (D-Dataset Display, V-View Aliases)

Storage Copy Pools
  Data Copy Pool  DSN$DB9C$DB
  Log Copy Pool   DSN$DB9C$LG
  Backup Pool     DB9CCPB

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 5-12 Subsystem setup - screen 1 of 4

In these windows you can see the SMS Copy Pool names and the ICF catalogs used for your data and logs as well as all related volumes. This is very important information to determine the health of your system level backup and recovery implementation.

Figure 5-13 and Figure 5-14 show the information related to bootstrap data sets and log data sets.

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/19 16:06:40
Option ==>>>                                           Scroll ==> PAGE
-----
Subsystem: D9C1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/19/2008 Analysis Recommended: N
Message: Other non-DB2 data will be backed up and restored.
-----
Row 15 of 68  -+
Line Cmds: (V-Volume List)

Boot Strap Datasets
D9C1 - BSDS 1  DB9CL.D9C1.BSDS01          Volume SBOX5A
D9C1 - BSDS 2  DB9CL.D9C1.BSDS02          Volume SBOX5I
D9C2 - BSDS 1  DB9CL.D9C2.BSDS01          Volume SBOX5D
D9C2 - BSDS 2  DB9CL.D9C2.BSDS02          Volume SBOX5N
Line Cmds: (R-Rename BSDS, M-Move BSDS)

Active Log Datasets
D9C1 - Log 1    DB9CL.D9C1.LOGCOPY1.DS01   Volume SBOX5E
D9C1 - Log 1    DB9CL.D9C1.LOGCOPY1.DS02   Volume SBOX5B
D9C1 - Log 1    DB9CL.D9C1.LOGCOPY1.DS03   Volume SBOX5G
D9C1 - Log 2    DB9CL.D9C1.LOGCOPY2.DS01   Volume SBOX5L

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 5-13 Subsystem setup - screen 2 of 4

```

ARY$SSID          ----- Subsystem Setup Information ----- 2008/02/19 16:07:03
Option ==>>>                                           Scroll ==> PAGE
-----
Subsystem: D9C1    Active: Yes    Datasharing: Yes
Date of Last Analysis: 02/19/2008 Analysis Recommended: N
Message: Other non-DB2 data will be backed up and restored.
-----
Row 29 of 68  -+
D9C1 - Log 2    DB9CL.D9C1.LOGCOPY2.DS02   Volume SBOX5K
D9C1 - Log 2    DB9CL.D9C1.LOGCOPY2.DS03   Volume SBOX50
D9C2 - Log 1    DB9CL.D9C2.LOGCOPY1.DS01   Volume SBOX5C
D9C2 - Log 1    DB9CL.D9C2.LOGCOPY1.DS02   Volume SBOX5F
D9C2 - Log 1    DB9CL.D9C2.LOGCOPY1.DS03   Volume SBOX5H
D9C2 - Log 2    DB9CL.D9C2.LOGCOPY2.DS01   Volume SBOX5M
D9C2 - Log 2    DB9CL.D9C2.LOGCOPY2.DS02   Volume SBOX5J
D9C2 - Log 2    DB9CL.D9C2.LOGCOPY2.DS03   Volume SBOX5P
Line Cmds: (R-Rename Log, M-Move Log)

Alias used with associated MVS User Catalogs
DB9CA          UCAT.DB9CLOGS                Log
DB9CD          UCAT.DB9CDATA                Data Other
DB9CL          UCAT.DB9CLOGS                Log

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 5-14 Subsystem setup - screen 3 of 4

```

ARY$SSID      ----- Subsystem Setup Information ----- 2008/02/19 16:07:18
Option ==>                                           Scroll ==> PAGE
-----
Subsystem: D9C1   Active: Yes   Datasharing: Yes
Date of Last Analysis: 02/19/2008 Analysis Recommended: N
Message: Other non-DB2 data will be backed up and restored.
-----
Row 43 of 68  -+
Line Cmds: (D-Dataset Display, M-Merge catalog entries, R-Rename Alias)

Volumes used by this DB2 Subsystem
Volume      Data  DataCat  ActLog ActCat  ArcLog ArcCat  Other  Pool
SBOX5A      No   No       Yes   No     No     Yes   No     Yes
SBOX5B      No   No       Yes   No     No     No    No     Yes
SBOX5C      No   No       Yes   No     No     No    No     Yes
SBOX5D      No   No       Yes   No     No     No    No     Yes
SBOX5E      No   No       Yes   No     No     No    No     Yes
SBOX5F      No   No       Yes   No     No     No    No     Yes
SBOX5G      No   No       Yes   No     No     No    No     Yes
SBOX5H      No   No       Yes   No     No     No    No     Yes
SBOX5I      No   No       Yes   No     No     No    No     Yes
...
SBOX5M      No   No       Yes   No     No     No    No     Yes
SBOX5N      No   No       Yes   No     No     No    No     Yes
SBOX5O      No   No       Yes   No     No     No    No     Yes
SBOX5P      No   No       Yes   No     No     No    No     Yes
SBOX5S      Yes  No       No    No     No     No    Yes    Yes
SBOX5T      Yes  No       No    No     No     No    No     Yes
SBOX5U      Yes  Yes      No    No     No     No    No     Yes
SBOX5V      Yes  No       No    No     No     No    No     Yes
Line Cmds: (D-Dataset Display, M-Move all Datasets on Volume)

***** Bottom of Data *****

Valid Primary Cmds: (Analyze, Reanalyze)

```

Figure 5-15 Subsystem setup - screen 4 of 4

As can be seen in Figure 5-15, there exists other non-DB2 data that can be backed up and restored from the volumes when flashed (during the BACKUP SYSTEM and RESTORE SYSTEM processes). The DB2 log and object data are properly segregated, but Recovery Expert has detected other non-DB2 data sets on one or more subsystem volumes.

The volumes in question are SBOX5S, SBOX5T, SBOX5U, and SBOX5V.

These other data sets (non DB2) will also be backed up and RESTORED when the DB2 system is restored. This may or may not be your desired result. If there is a z/OS user catalog not being used by DB2 on any of the listed volumes, it will not be automatically disconnected and will result in errors when the DB2 system is restored. These catalogs should be moved to other volumes that are not being used by DB2. Empty DB2 catalogs must be manually unallocated (disconnected) from the catalog address space at restore time.

Figure 5-16 shows the panel that you will see to create a new profile.

```

ARY$BPRD          ----- Backup Profile Display ----- 2008/02/19 16:09:03
Option ==>                               Scroll ==> PAGE
-----
Profile   Like  *
Creator   Like  *                               Row 1 of 1  >
-----

Cmd  Name                               Creator  SSID  Updt
C    Press Enter to Create Profile
***** Bottom of Data *****

ARYR031I - No Profiles were found that match your selection criteria. Press
enter to create a new profile or change the selection criteria.

```

Figure 5-16 Creating a backup profile for DB2 D9C1

Figure 5-17 shows what was entered.

```

----- Enter New Backup Profile Options -----
Creator      PAOLOR6
Profile Name  D9C1
Description   D9C1 System Level Backup
Update Option U (Update, View only, No access)
Press ENTER to process or PF3 to Cancel
-----

```

Figure 5-17 Creating a backup profile - enter your values

Figure 5-18 on page 108 shows the results. In this window the following fields are read only:

- ▶ **Current generation** - This field is set to 00 when you first create a profile. After the profile has been built and submitted, this field contains the generation that is currently mirroring DB2.
- ▶ **Setup Needed** - This field is set to Y when you create a profile. Once profile setup has been performed, this field contains an N when you update the profile.

The following fields should be specified:

- ▶ **DB2 Subsystem** - The subsystem for this profile in the DB2 Subsystem field.
- ▶ **Backup Method** - B for BCV backup.
- ▶ **Backup Scope** - A full backup to be taken (both data and logs) or to back up data only.
- ▶ **Backup Generations** - The number of generations of backups that you want to keep. Valid values are 1 to 8.

- For more information, see “Creating a BCV backup profile” in *DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1*, SC19-1222.

Figure 5-18 Backup profile detail window

Once you have entered your choices (in our example we chose D for a DB2 system level backup and restore method) you can save the information. See Figure 5-19.

```

ARY$BPRD          ----- Backup Profile Display ----- 2008/02/19  16:20:01
Option ==>>>                                           Scroll ==> PAGE
-----
Profile   Like  *
Creator   Like  *
                                           Row 1 of 1  >
-----

Cmd  Name                      Creator  SSID  Updt
   D9C1                      PAOLOR6  D9C1  U
----- Bottom of Data -----

| ARYR069I - Profile "PAOLOR6.D9C1" saved |
Valid Line Cmds: -----date, V-View)

```

Figure 5-19 Results from saving your newly created profile

The profile will be saved and will be used for both backup and recovery processes using Recovery Expert.

Step 2 - Build and execute the BACKUP SYSTEM Job

Next we want to build the system level backup using the profile that we just created and saved.

We use the build function (option B) to build jobs and related JCL to perform the backup process. See Figure 5-20.

```

ARY$BPRD          ----- Backup Profile Display ----- 2008/02/19  16:22:21
Option ==>>>                                           Scroll ==> PAGE
-----
Profile   Like  *
Creator   Like  *
                                           Row 1 of 1  >
-----

Cmd  Name                      Creator  SSID  Updt
   B D9C1                      PAOLOR6  D9C1  U
----- Bottom of Data -----

Valid Line Cmds: (B-Build, C-Create, D-Delete, R-Rename, U-Update, V-View)

```

Figure 5-20 Using backup profile to build JCL

This builds the job in two phases. The first job does a SETUP and the second job performs the actual BACKUP SYSTEM. The SETUP process reads the PROFILE and validates the volumes. This is a safeguard to ensure that the system level backup will not encounter any volume-related or DFSMSHsm-related problems.

Figure 5-21 shows the option to edit the generated JCL and build the job with the SETUP option. This is also where you can change the JOBCARD information. This information will be stored in your ISPF PROFILE data set for use in the future.

```
Edit Generated Job      Y      (Yes/No)
Setup Job               Y      (Yes/No)
Perform Offload        N      (Yes/No)
Backup Repository      N      (Yes/No)

Build job in Dataset   PAOL0R6.SLB.JCL
Member                DB2BACK

Job Cards:
==> //PAOL0R6Z JOB (XXX,P0K),HENNIE,CLASS=A,MSGCLASS=X
==> /*JOBPARM SYSAFF=SC63
==> /*
==> /*

Press ENTER to process or PF3 to Cancel
```

Figure 5-21 Building backup job - specify JOB card and data set name that will contain JCL

The JCL generated is listed in Figure 5-22.

```

/** * * * * *
/**
/** Step:    ARYBACK
/**
/** Desc:    This step will invoke the System Backup job.
/**
/** * * * * *
/**
//ARYBACK EXEC PGM=ARY@MAIN,REGION=006M,COND=(4,LT)
/**
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//          DD DISP=SHR,DSN=DB9C9.SDSNEXIT
//          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//ARYBPROF DD DISP=SHR,DSN=ARY.PROFILES
//ARYBOFFL DD DISP=SHR,DSN=ARY.OFFOPTS
//ARYBPMAP DD DISP=SHR,DSN=ARY.PROFILE.MAPS
//ARYBPCAT DD DISP=SHR,DSN=ARY.PROFILE.CATS
//ARYSBACK DD DISP=SHR,DSN=ARY.SYSBACK
//ARYSBOBJ DD DISP=SHR,DSN=ARY.SYSBACK.OBJS
//ARYSBVOL DD DISP=SHR,DSN=ARY.SYSBACK.VOLS
//ARYSBSSD DD DISP=SHR,DSN=ARY.SYSBACK.SSIDS
//ARYBREPT DD DISP=SHR,DSN=ARY.BREPORT
//SYSOUT   DD SYSOUT=*
//ARYOUT   DD SYSOUT=*
//ARY#REPT DD SYSOUT=*
//ARYSNAPO DD SYSOUT=*
//ARY#PARM DD DSN=ARY.V2R1M0.SARYSAMP(ARY#PARM),DISP=SHR
//ARYIN    DD *
           BACKUP PAOLOR6.D9C1
           SETUP
/*

```

Figure 5-22 Generated backup JCL - with SETUP

This JCL will not do the backup, but will validate and save the validated information in the BACKUP PROFILE. You are now ready to perform a system level backup verification. The output is shown in Example 5-18.

Example 5-18 Output of backup JCL with SETUP

```

ARYS001I - Recovery Expert for DB2 z/OS Starting. Version 02.01.013
ARYS162I - Parmlib used for this execution
ARYS003I - Control Cards:
ARYS004I -   BACKUP PAOLOR6.D9C1
ARYS004I -       SETUP
ARYS013I - Backup profile PAOLOR6.D9C1 was read from the repository.
ARYS038I - Performing profile volume map validation...
ARYS039I - Volume map validation complete.
ARYS075I - Performing DB2 source volume validation...
ARYS076I - DB2 source volume validation complete. All DB2 volumes are in this profile.
ARYS079I - Profile setup is complete. Remove the "SETUP" control card to run the backup.
ARYS002I - Recovery Expert for DB2 z/OS complete. RC=000.

```

You can now run the BACKUP SYSTEM job.

There are two ways to run the job:

- ▶ Edit the JCL online and remove “SETUP” and re-submit.
- ▶ Press PF3 and the Recovery Expert will remove the “SETUP” keyword and then submit the job.

The output of the system level backup invokes BACKUP SYSTEM via Recovery Expert's ARY@MAIN program. This is shown in Example 5-19.

Example 5-19 Output of Recovery Expert generated system level backup

```

ARYS001I - Recovery Expert for DB2 z/OS Starting. Version 02.01.013
ARYS162I - Parmlib used for this execution
ARYS003I - Control Cards:
ARYS004I - BACKUP PAOLOR6.D9C1
ARYS004I -
ARYS013I - Backup profile PAOLOR6.D9C1 was read from the repository.
ARYS039I - Volume map validation complete.
ARYS075I - Performing DB2 source volume validation...
ARYS076I - DB2 source volume validation complete. All DB2 volumes are in this profile.
ARYS275I - DB2 checkpoint taken at RBA/LRSN C1F99FB8D64A
ARYS004I - Invoking IBM System Level Backup utility...
ARYS020I - IBM System Backup Utility Complete. Token = C4F9C3F1C1F99FBBA7F5594BC1F88875EE61
ARYS080I - Backup with timestamp 2008/02/19-16:39:06, generation 01 was saved in the repository.
ARYS002I - Recovery Expert for DB2 z/OS complete. RC=000.
          Recovery Expert for DB2 z/OS
                Backup Summary Report

```

```

Utility Executed:..... Backup
Profile Name:..... PAOLOR6.D9C1
DB2 Subsystem:..... D9C1
DB2 Version:..... 0910
Backup Type:..... IBM System Level Backup
Backup Contains:..... Object Data and Log Data
      Nbr of Volumes:..... 0022
      HSM Backup Token:..... C4F9C3F1C1F99FBBA7F5594BC1F88875EE61
      Backup LRSN:..... C1F99FBBA7F5
      Last Checkpoint LRSN:..... C1F99FB8D64A
      Backup Date:..... 02/19/2008
      Backup Time:..... 16:39:06
      Consistency Method:..... IBM System Level Backup
      Supports Object Restore:.. No
      I/O Suspend Time:..... 2008-02-19-16.38.53.613941
      I/O Resume Time:..... 2008-02-19-16.39.06.195328
      Backup Elapsed:..... 12.58 Seconds

```

Recovery Expert for DB2 z/OS
Backup Volume Detail Report

<-Source Volumes->			<-Targets->		<-----Data Types----->					
Volser	Ucb#	Devtyp	Volser	Ucb#	Obj	OCat	ALog	ACat	RLog	RCat
SBOX5A	D800	3390-9	SBOX7I	D807	No	No	Yes	No	No	Yes
SBOX5B	D900	3390-9	SBOX7J	D907	No	No	Yes	No	No	No
SBOX5C	DA00	3390-9	SBOX7K	DA07	No	No	Yes	No	No	No
....										
SBOX5R	D902	3390-9	SBOX7H	DF06	No	No	No	No	No	No
SBOX5S	DA02	3390-9	SBOX7A	D806	Yes	No	No	No	No	No

SBOX5T	DB02	3390-9	SBOX7B	D906	Yes	No	No	No	No	No
SBOX5U	DC02	3390-9	SBOX7C	DA06	Yes	Yes	No	No	No	No
SBOX5V	DD02	3390-9	SBOX7D	DB06	Yes	No	No	No	No	No

The output differs from a standard DB2 BACKUP SYSTEM utility job. It will show the volumes that were successfully copied (which you have to view from the SYSLOG or DFSMSHsm log). It records the backup in the BSDS as well as in the Recovery Services repository. The usual updates that BACKUP SYSTEM performs (updating the RBLP in DBD01) also happen.

Step 3 - Build the RESTORE SYSTEM jobs

You are now ready to build the Restore System JCL using Recovery Expert's ISPF interface.

When the BACKUP SYSTEM job in step 2 completed, it stored the BACKUP SYSTEM information not just in DB2's BSDS (and updated the header page in DBD01, as previously discussed, it also stored information about the volumes, ICF catalogs, copy pool information, and so on, in Recovery Expert's own system level backup Restore System repository. This is currently a set of VSAM files.

The SBRS Repository is used by Recovery Expert to generate and build JCL for System Restore functions for the entire system as well as object-level restore (if required). It also enables the DB2 DBA to view, in one single source, the backup resources from a BACKUP SYSTEM.

From Recovery Expert's main window, you now select option **R** (System Restore and Offload), which enables you to:

1. View all the system level recoveries (as recorded in Recovery Expert's SLBS repository).
2. View the details of the system level backups (via an online report).
3. Select the recovery point to recover to.
4. Build the JOB.

The LRSN (in our recovery example we are restoring the DB2 subsystem in a data sharing environment) in the following figures is highlighted in order to follow the next sequence of events via the screen captures.

In the next series of screens in Figure 5-23 through Figure 5-25 on page 115 we select a recovery point, view a detailed display, and build the RESTORE SYSTEM jobs. Three jobs are built:

- ▶ Creating the CRCR record in the members' BSDS
- ▶ Creating the RESTORE SYSTEM JCL
- ▶ Creating the LOG RESTORE JCL

```

ARY$MAIN          ----- Recovery Expert for DB2 z/OS -----
Option  ==> R

User: PAOLOR6 - ARY

-----

B. System Backup Profiles

R. System Restore and Offload

O. Object Recovery Profiles

D. Disaster Recovery Profiles

M. DB2 Subsystem Analysis and Setup

S. Product Setup

X. Exit

Enter END command to return to ISPF.
  
```

Figure 5-23 Selecting the system restore option to build RESTORE SYSTEM JCL

```

ARY$REST          ----- Restore System Display ----- 2008/02/19 17:06:04
Option  ==>                                           Scroll ==> PAGE

DB2 Subsystem ID  *

-----

Select a row from the list of recovery points displayed below.
To enter an RBA/LRSN and have a recovery point automatically
selected for you, enter the RESTORE primary command followed
by a DB2 subsystem ID.

-----

Row 1 of 2  >
Cmd  Date      Time      Data  Mixed  On   On   Obj   Nbr
    02/19/2008 16:39:06 No   No   Yes  No   No   DB2  0022
V SSID: D9C1  RBA/LRSN: C1F99FBB7F5
***** Bottom of Data *****

Valid Line Commands: ( S - Select, D - Delete, V - View Report, O - Offload )
  
```

Figure 5-24 High-level view of the system level backup

```

ARY$BREP          ----- Backup Report Display ----- 2008/02/19 17:07:58
Option ===>                               Scroll ===> PAGE

```

```

-----
Row 1 of 49  +>

```

```

Recovery Expert for DB2 z/OS
Backup Summary Report

```

```

Utility Executed:..... Backup
Profile Name:..... PAOLOR6.D9C1
DB2 Subsystem:..... D9C1
DB2 Version:..... 0910
Backup Type:..... IBM System Level Backup
Backup Contains:..... Object Data and Log Data
Nbr of Volumes:..... 0022
HSM Backup Token:..... C4F9C3F1C1F99FBBA7F5594BC1F88875EE61
Backup LRSN:..... C1F99FBBA7F5
Last Checkpoint LRSN:..... C1F99FB8D64A
Backup Date:..... 02/19/2008
Backup Time:..... 16:39:06
Consistency Method:..... IBM System Level Backup
Supports Object Restore:.. No
I/O Suspend Time:..... 2008-02-19-16.38.53.613941
I/O Resume Time:..... 2008-02-19-16.39.06.195328

```

```

Recovery Expert for DB2 z/OS
Backup Volume Detail Report

```

<-Source Volumes->			<-Targets->		<-----Data Types----->					
Volser	Ucb#	Devtyp	Volser	Ucb#	Obj	OCat	ALog	ACat	RLog	RCat
SB0X5A	D800	3390-9	SB0X7I	D807	No	No	Yes	No	No	Yes
SB0X5B	D900	3390-9	SB0X7J	D907	No	No	Yes	No	No	No
SB0X5C	DA00	3390-9	SB0X7K	DA07	No	No	Yes	No	No	No
...										
SB0X5R	D902	3390-9	SB0X7H	DF06	No	No	No	No	No	No
SB0X5S	DA02	3390-9	SB0X7A	D806	Yes	No	No	No	No	No
SB0X5T	DB02	3390-9	SB0X7B	D906	Yes	No	No	No	No	No
SB0X5U	DC02	3390-9	SB0X7C	DA06	Yes	Yes	No	No	No	No
SB0X5V	DD02	3390-9	SB0X7D	DB06	Yes	No	No	No	No	No
***** Bottom of Data *****										

Figure 5-25 Viewing the detailed backup system report

The system level recovery information of Figure 5-26 is sourced from the Recovery Expert SLBRS Repository.

```

ARY$REST          ----- Restore System Display ----- 2008/02/19 17:12:26
Option ===>                               Scroll ===> CSR

DB2 Subsystem ID *
-----
      Select a row from the list of recovery points displayed below.
      To enter an RBA/LRSN and have a recovery point automatically
      selected for you, enter the RESTORE primary command followed
      by a DB2 subsystem ID.
-----
                                     Row 1 of 2  >
Cmd  Date      Time      Data Mixed On   On      Obj      Nbr
   S  02/19/2008 16:39:06 Only Data Disk Offload Rcvr Type Vols
      SSID: D9C1 RBA/LRSN: C1F99FBBA7F5
***** Bottom of Data *****

Valid Line Commands: ( S - Select, D - Delete, V - View Report, O - Offload )

```

Figure 5-26 Selecting the system level backup for recovery

As mentioned before, the BACKUP SYSTEM time stamp in BSDS is updated, too (in step 2), and in Example 5-20 the job output from a DSNJU004 utility run is listed to show the contents for informational purposes only.

Example 5-20 DSNJU004 Output - BACKUP SYSTEM UTILITY HISTORY - showing the backup entry

BACKUP SYSTEM UTILITY HISTORY							
SUBSYSTEM ID D9C1							
22:52:03 FEBRUARY 19, 2008							
DATA	START STCK	LOG	RBLP	DATA COMPLETE LRSN	DATA/LOG COMPLETE DATE	COMPLETE LTIME	LOCATION NAME
C1F99FBBA7F5594B	C1F99FBD2B868C46	C1F88875EE61	C1F99FC6EC25	2008/02/19 16:39:06	DB9C	TOKEN = C4F9C3F1C1F99FBBA7F5594BC1F88875EE61	
C1F888B380EEBD43	C1F888B42BFF8904	C1F8886C1B39	C1F888BD41E9	2008/02/18 19:50:42	DB9C	TOKEN = C4F9C3F1C1F888B380EEBD43C1F8886C1B39	
C1F86E8E03F0E5C3	C1F86E90A8DEA183	C1F1161A0D5F	C1F86E9AB8B0	2008/02/18 17:53:46	DB9C	TOKEN = C4F9C3F1C1F86E8E03F0E5C3C1F1161A0D5F	

In Figure 5-27 you can see the window that enables you to build the RESTORE SYSTEM JCL.

Restore Only Data	Y
Select a Recovery Point	N
Recover to RBA/LRSN	C1F99FBBA7F5

Select whether to Restore Data and Logs or Data only. If restoring Data and Logs you will not be able to select a Timestamp Recovery Point or change the Recover to RBA/LRSN.

Press ENTER to generate the Restore Job or PF3 to Cancel

Figure 5-27 Selection choice for the recovery

Recovery Expert SBRS optionally restores *both* data volumes *and* log volumes. This is not possible with DB2 RESTORE only. If both data and logs are restored, DB2 will be restored back to the point of the full system backup. When DB2 is started, it will backout any transactions that were uncommitted at the time of the system backup.

In the screen shown in Figure 5-28 you enter the data set and member names for the RESTORE SYSTEM JCL. The first member, SLBRES0, updates the BSDS of each member in the data sharing group (our example has two members, D9C1 and D9C2) in two steps.

The JCL has basic comments about the instructions for running the RESTORE SYSTEM jobs.

Note: Recovery Expert SBRS will help you with most of the manual steps needed to successfully RESTORE an entire DB2 subsystem from a system level backup.

Edit Generated Job	Y	(Yes/No)
--------------------	---	----------

Build job in Dataset	PAOLOR6.SLB.JCL	
Member	SLBRES0	Conditional Restart Member
Member	SLBRES1	Restore System Member
Member	SLBRES2	Log Restore Member

Job Cards:

```
==> //PAOLOR6Y JOB (XXX,POK),HENNIE,CLASS=A,MSGCLASS=X
==> /*JOBPARM SYSAFF=SC63
==> /*
==> /*
```

Press ENTER to process or PF3 to Cancel

Figure 5-28 JCL generation window with data set name and members for restore system jobs

Figure 5-29 shows the conditional restart JCL for the two DB2 members.

```
//ARYCRCR EXEC PGM=DSNJU003,REGION=006M,COND=(4,LT)
/*
//STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS01
//SYSUT2 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS02
//SYSIN DD *
    CRESTART CREATE,SYSPITR=C1F99FBBA7F5,FORWARD=YES,BACKOUT=YES
/*
/*
//ARYCRCR EXEC PGM=DSNJU003,REGION=006M,COND=(4,LT)
/*
//STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=DB9CL.D9C2.BSDS01
//SYSUT2 DD DISP=SHR,DSN=DB9CL.D9C2.BSDS02
//SYSIN DD *
    CRESTART CREATE,SYSPITR=C1F99FBBA7F5,FORWARD=YES,BACKOUT=YES
/*
```

Figure 5-29 Conditional restart control record create for each DB2 member

This is the two-step job that uses the DB2 utility DSNJU003 to update both members in our data sharing group with the correct SYSPITR LRSN to enable the RESTORE SYSTEM job to execute successfully.

Figure 5-30 shows the step to invoke the Recovery Expert restore job.

```

/** * * * * *
/**
/** Step:    ARYREST
/**
/** Desc:    This step will invoke the System Restore job.
/**
/**          When this job is complete. DB2 can be restarted. If
/**          you did not restore the DB2 logs, you will need to
/**          respond Yes to the WTOR:
/**          XXXX CONDITIONAL RESTART RECORD INDICATES TRUNCATION
/**          AT RBA XXXXXXXXXXXX. REPLY Y OR N
/**
/** * * * * *
/**
/**ARYREST EXEC PGM=ARY@MAIN,REGION=006M,COND=(4,LT)
/**
/**STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
/**DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
/**ARYBPROF DD DISP=SHR,DSN=ARY.PROFILES
/**ARYBOFFL DD DISP=SHR,DSN=ARY.OFFOPTS
/**ARYBPMAP DD DISP=SHR,DSN=ARY.PROFILE.MAPS
/**ARYBPCAT DD DISP=SHR,DSN=ARY.PROFILE.CATS
/**ARYSBACK DD DISP=SHR,DSN=ARY.SYSBACK
/**ARYSBOBJ DD DISP=SHR,DSN=ARY.SYSBACK.OBJS
/**ARYSBVOL DD DISP=SHR,DSN=ARY.SYSBACK.VOLS
/**ARYSBSSD DD DISP=SHR,DSN=ARY.SYSBACK.SSIDS
/**ARYBREPT DD DISP=SHR,DSN=ARY.BREPORT
/**ARY#REPT DD SYSOUT=*
/**SYSOUT DD SYSOUT=*
/**ARYOUT DD SYSOUT=*
/**ARYSNAPO DD SYSOUT=*
/**ARY#PARM DD DSN=ARY.V2R1M0.SARYSAMP(ARY#PARM),DISP=SHR
/**ARYIN DD *
      RESTORE PAOL0R6.D9C1
      GENERATION 01
      DATE 02/19/2008
      TIME 16:39:06
/**
/**

```

Figure 5-30 Recovery Expert system level restore JCL

Figure 5-31 shows the Recovery Expert job for the restore log only.

```

/** * * * * *
/**
/** Step:    ARYLOG
/**
/** Desc:    This step will invoke the IBM DSNUTILB stand alone
/**           utility to Restore the System Logs.
/**
/**           DB2 must be up and you must have responded Yes to the
/**           WTOR:
/**           XXXX CONDITIONAL RESTART RECORD INDICATES TRUNCATION
/**           AT RBA XXXXXXXXXXXX. REPLY Y OR N
/**
/** * * * * *
/**
/** ARYLOG    EXEC PGM=DSNUTILB,REGION=006M,PARM=(D9C1,)
/**
/** STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
/**          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
/** SYSPRINT DD SYSOUT=*
/** SYSOUT   DD SYSOUT=*
/** UTPRINT  DD SYSOUT=*
/** SYSIN    DD *
/**          RESTORE SYSTEM LOGONLY
/**

```

Figure 5-31 Recovery Expert RESTORE SYSTEM LOGONLY JCL

Step 4 - Prepare the environment and run the RESTORE

In order for a successful DB2 RESTORE SYSTEM execution, certain steps need to be taken. In our example we have two members in a data sharing group. The data sharing environment needs to have the XCF structures deleted, as well as inculcating and closing the ICF user catalogs.

For examples on how to UNALLOCATE and CLOSE the ICF catalogs, see Example 5-15 on page 98 and Example 5-16 on page 98.

Step 4.1 - Run the jobs to create the SYSPITR CRCR for each DB2 subsystem

Note: Before you run the jobs to create the SYSPITR CRCR, ensure that all DB2 members in a data sharing environment are stopped. In a non-data sharing environment, Recovery Manager's SBRS function will generate a step to stop DB2 for you.

Run job SLBRES0, which was built and saved in the previous step. This job has two steps, one for each DB2 member, to update the BSDS with the CRCR information.

See "Step 3 - Prepare the environment for RESTORE SYSTEM" on page 94.

Step 4.2 - Delete all the CF structures

This is a task that you currently perform outside of DB2 and Recovery Expert.

Note: With the PTF for APAR PK59675 applied, when using the ISPF interface, Recovery Expert resets the CF structures before restore and you do not need to perform step 4.2.

See the commands and description as well as examples in “Step 5 - Verify that ICF catalogs volumes are not in use by the catalog” on page 97.

Step 4.3 - Run the job to restore the system database volumes

This job restores the system database volumes to the backup points.

Step 5 - Start DB2

This task, like before, is performed outside of Recovery Expert.

Note: In a non-data sharing group, Recovery Manager's SBRS function will start the DB2 member for you. Currently, you have to manually start all the DB2 members in a data sharing group.

You reply to DB2's outstanding message and DB2 will be started in MAINTENANCE mode. See “Step 6 - Run RESTORE SYSTEM” on page 98 in for examples and output.

In a data sharing environment, ensure that you start all the DB2 members.

Step 6 - Run the RESTORE SYSTEM jobs

You are now ready to let Recovery Expert execute the RESTORE SYSTEM jobs that were generated by Recovery Expert in step 3.

Refer to “Step 6 - Run RESTORE SYSTEM” on page 98 for examples and output from a RESTORE SYSTEM execution.

During this step, Recovery Expert's SBRS function issues the relevant DFSHSM commands to restore the DATABASE copypool back to the point of the chosen system backup. In this example, the LOG volumes will not be restored because you chose to roll forward the log.

ISPF over GUI system level backup and recovery advantages

Table 5-1 on page 122 summarizes the differences in performing tasks for system level backup between the GUI and ISPF interfaces of Recovery Expert. The advantages that Recovery Expert's SBRS function provides are listed as they relate to system level backup and restore.

Table 5-1 Comparing Recovery Expert GUI and ISPF system level backup and recovery

Task	GUI	SBRS	Comments
Build and run the BACKUP SYSTEM job.		X	<p>GUI comments:</p> <p>You currently do not build the BACKUP SYSTEM JCL in the GUI interface. This is something you build and run outside of the GUI interface. You use the standard DB2 BACKUP SYSTEM utility JCL.</p> <p>ISPF comments:</p> <p>You can build various system level backups via the ISPF interface. In our example in this chapter we build the BACKUP SYSTEM job using the ISPF interface. This was done after we defined a BACKUP PROFILE in step 1.</p> <p>Notes:</p> <p>In both our examples we executed the BACKUP SYSTEM utility. Both performed the same function.</p> <p>Recovery Expert records all the information about the backup, its copy pools, the volumes in the stogroups, and user catalogs in its repository. This is in addition to the DB2 BACKUP SYSTEM utility updating DB2 with the relevant system level backup information.</p> <p>Recovery Expert will source this information when building recovery (system level) or restore (object level) jobs in any supported version of DB2.</p>
Build the system level restore jobs.	X	X	
Unallocate and close the ICF catalogs.		X	<p>GUI comments:</p> <p>You have to perform this function manually.</p> <p>ISPF:</p> <p>Recovery Expert's SBRS function will perform this function as part of the recovery.</p>
Delete the XCF structures for the data sharing environment.		X	<p>GUI comments:</p> <p>You have to perform this function manually outside of Recovery Expert.</p> <p>ISPF:</p> <p>With APAR PK59675 applied RE will reset the structures for you.</p>

Task	GUI	SBRS	Comments
Start DB2.		X	<p>GUI comments: You have to start DB2 manually outside of the GUI.</p> <p>ISPF: The ISPF interface's SBRS function will start DB2 in a non-data sharing environment. For data sharing, you currently still need to start it manually.</p>

In addition to the specific benefits listed above, Recover Expert's System Backup and Restore Services offer a comprehensive list of validity checking. It:

- ▶ Ensures that the ICF user catalogs are included in the backup, ensuring integrity of the restoration.
- ▶ Verifies that the target volumes for a backup are not in use by another backup profile.
- ▶ Verifies that the log and object data are on separate volumes. If not, a backup is still allowed but System Backup and Restore Services will issue warnings as seen in our examples.
- ▶ Validates that DB2 log data sets and object data sets are cataloged in separate user catalogs. If not, the backup is still allowed but System Backup and Restore Services will issue warnings, and you can only back up and restore the full subsystem (both logs and data).
- ▶ System Backup and Restore Services's subsystem setup facility collects and displays information about the DB2 subsystem's user catalogs, boot strap data sets, active logs, and related DB2 object data sets.

Archived

Object recovery functions from system level backup

In this chapter we provide various scenarios that describe object level recovery from backup resources. We show how to perform the following object recovery scenarios using the ISPF interface:

- ▶ To current using SLB
- ▶ To LRSN/RBA or quiesce point using SLB
- ▶ To copy (last full copy, last incremental copy)

We also show examples of how Recovery Expert can choose the optimal alternative when several recovery options are available.

This chapter discusses:

- ▶ Object recovery with Recovery Expert under DB2 V8
- ▶ Object Recovery using DB2 9 for z/OS system level backup
- ▶ Object recovery with Recovery Expert using system level backup
- ▶ Optimal recovery action with DB2 Recovery Expert

6.1 Object recovery with Recovery Expert under DB2 V8

This section shows the step-by-step procedure to perform table space or index space recovery using volume backup generated using Recovery Expert with *Backup Method = D* (that is, DB2 system level backup). The sample scenario discussed in this section was carried out on a Version 8 subsystem. With Recovery Expert V2.1, you can perform data set level recovery using a copy pool backup on disk performed by Recovery Expert under DB2 V8. And you will be able to use the same procedure to recover an object in any of the following scenarios:

- ▶ On V7 or V9 subsystem (see 6.3, “Object recovery with Recovery Expert using system level backup” on page 143)
- ▶ For multiple objects in a database or one or more indexes in an object recovery profile
- ▶ Using FlashCopy by IBM or Business Continuity Volumes (BCV) or Snap backup by EMC Technology

We show how the object recovery options with Recovery Expert can be tailored to meet your recovery needs in 6.1.1, “Managing object profiles for recovery” on page 126, through 6.1.4, “Recovery object to copy (full or incremental copy or quiesce)” on page 138.

6.1.1 Managing object profiles for recovery

For all object recovery scenarios you need to create and manage object recovery profiles. Object profile information is stored in the VSAM repository. The interface for creating and managing the object recovery profile can access the similar object profiles of the DB2 Automation Tool.

Note: The object profiles created in the DB2 Automation Tool can be used from a GUI, but the object profiles created in Recovery Expert can only be used from ISPF.

Create an objects profile for recovery

In this section we show how to create an objects profile for recovery.

Key in option 0 after you bring up Recovery Expert. You will see the objects profile name search window, as shown in Figure 6-1.

Enter Objects Profile Like to Display	
Profile Like	*
Creator Like	PA0L0R7

Figure 6-1 Objects profile name search

If there are no profiles previously defined, press Enter and you will see the panel shown in Figure 6-2.

```

ARY$OPRD          ----- Objects Profile Display ----- 2008/02/14 16:33:01
Option ===>                                           Scroll ===> PAGE
-----
Profile   Like  *
Creator   Like  PAOLOR7                               Row 1 of 1      >
-----

Cmd  Name                                           Creator  SSID  Updt
C    Press Enter to Create Profile
***** Bottom of Data *****

-----
ARYR031I - No Profiles were found that match your selection criteria. Press
enter to create a new profile or change the selection criteria.

```

Figure 6-2 Objects Profile Display window

Press Enter again in the Objects Profile Display window with a C under the Line Commands column (Cmd). You will get to the window shown in Figure 6-3, which serves as the entry window for the objects profile name.

```

Enter New Objects Profile Data

Creator      PAOLOR7

Profile Name  OBJECT RECOVERY PROFILE#1

Description  A MEANINGFUL DESCRIPTION

DB2 SSID     D8F1

Update Option U  (Update, View only, No access)

Press ENTER to process or PF3 to Cancel

```

Figure 6-3 Objects profile name

In the window shown in Figure 6-3:

- Profile Name** Can be up to 30 characters long and is a required field
- Creator** Is the profile creator.
- DB2 SSID** Is the DB2 subsystem ID for which the profile was created, obtained from the ISPF profile pool.
- Update Option** This field indicates how users other than the profile creator may use the profile. U(pdate) Allows other users to update the profile. V(iew) allows other users to view but not update the profile. N(o access) Prevents other users from viewing or updating the profile.
- Description** Is the profile description and is optional.

Press Enter at the window shown in Figure 6-3 on page 127 to switch to the Add Objects to the Object Profile window shown in Figure 6-4.

Add objects to object profile

Figure 6-4 shows the initial window where we begin to add an object (this could be one or more table spaces or databases). You can get to this window later on to add more objects from the Objects Profile Display window, shown in Figure 6-2 on page 127, also.

Add Objects to the Object Profile		
Add Tablespaces	Y	(Yes/No)
Add Indexes	N	(Yes/No)
Press ENTER to process or PF3 to Cancel		

Figure 6-4 Add Objects to the Object Profile

Type Y at Add Tablespaces. This option can include the table space and all indexes dependent on this table space. The indexes can be confirmed in the next panel, as shown in Figure 6-5.

Enter Tablespaces Like to Display				
Database	Like	SAMPLE1	Wildcard	N (Y or N)
Tablespace	Like	RAVI*	Exclude	I (E or I)
Creator	Like	*		>
Process Dependent Indexes Y (Y or N)				

Figure 6-5 Enter recovery object (database or table space) name

If you would like to include indexes selectively, then type Y for the Add Indexes option, also.

Then press Enter to go to the window shown in Figure 6-5, where you can type the object name as shown.

Press PF3 if you have typed Y for the Add Indexes option to go to the window shown in Figure 6-6 to add indexes selectively.

Enter Indexes Like to Display				
Database	Like	Y	Wildcard	N (Y or N)
Creator	Like	*	> Exclude	I
Index	Like	*		>

Figure 6-6 Enter recovery object (Indexes)

The window shown in Figure 6-5 allows you to:

- Set a filter to limit the table spaces from which to select on the window that follows.
- Automatically select all table spaces that meet your selection criteria by using the Wildcard option field as shown. If you set the Wildcard option to Y, you will not be presented with a table space selection list. However, all table spaces that qualify the selection criteria will be included in the objects profile. If new table spaces are added that qualify the selection

criteria then those objects will be automatically included when the profile is reused afterwards

- Specify whether the objects that you select are to be included or excluded from the profile.
- Specify whether indexes dependent on the selected table spaces are to be included.

To select table spaces from a list, enter a database, table space, or creator name or mask in the appropriate fields.

For DB2 9 for z/OS and later subsystems only, you can specify whether clone tables are to be processed, as shown in Figure 6-7. Type Y in the Process Cloned Tables field to process only clone tables. Type B to process both clone table and base tables. Or type N to process only the base table.

Enter Tablespaces Like to Display			
Database	Like	*	Wildcard N (Y or N)
Tablespace	Like	*	Exclude I (E or I)
Creator	Like	*	>
Process Dependent Indexes		Y	(Y or N)
Process Cloned Tables		N	(Y, N or B)

Figure 6-7 Recovery object (database or table space) name for DB2 9 for z/OS

For DB2 V8 and later subsystems, the Creator Like field allows up to 128 bytes. To scroll in this field, place the cursor in the field and use the PF11 key to scroll right and the PF10 key to scroll left. Leave N in the Wildcard field.

When you press Enter, you will see the panel shown in Figure 6-8.

ARY\$OPRT --- Include Tablespace Selection --- 2008/02/14 18:31:43									
Option ==> Scroll ==> PAGE									

									Row 1 of 1 >
Database	Like	*							
Tablespace	Like	RJTS2							
Creator	Like	*							

Cmd	DBNAME	TSNAME	PART	CREATOR	DBID	OBID	PSID	BP00L	LOCKRULE
	RJDB2	RJTS2	0	PAOLOR4	277	1	2	BP1	P
***** Bottom of Data *****									
Valid Line Commands: (Select)									

Figure 6-8 Tablespace selection window

You can scroll to the right by using the PF11 key and to the left using the PF10 key to view all the attributes of the table space being displayed. You can enter the line command “s” to select the object and press PF3 to modify the object selection queue. You will see the message window shown in Figure 6-9.

ARYR338I - Object Queue has been Modified.

Figure 6-9 Object Queue message

If you press Enter then you will see the window shown in the Figure 6-10. Here you may use the Explode option to verify the selected object. This window also allows you to add or remove objects.

ARY\$OPRU

Option ==>

----- Update Object Profile Display -----

2008/02/14 20:14:01

Scroll ==> PAGE

Creator: PAOLOR7

Share Option: U (Upd,View,No)

DB2 Subsystem: D8F1

Name: OBJECT RECOVERY PROFILE#1

Description:

Update Recovery Options: N (Yes/No)

User: PAOLOR7

Row 1 of 1

>

Cmd	Type	Card	Wild Process	Include/	IX Crtr/	IX Name/	IX DB Name/	
			Indexes	Exclude	DB Name	TS Name	TS Crtr	Part
	TS	N	Y	INC	RJDB2	RJTS2	PAOLOR4	0

***** Bottom of Data *****

Valid Line Commands: (Add, Delete, Explode)

Primary Commands: (Explode)

Figure 6-10 Update Object Profile Display

You can press PF3 from this window to go back to the Objects Profile Display window, which would display a window as shown in Figure 6-11.

ARYR419I - Profile PAOLOR7.OBJECT RECOVERY PROFILE#1 has been successfully updated.

Figure 6-11 Profile update confirmation

Updating an object profile

You can update an object profile at any time to add or delete an object or change the recovery options. To update an object profile:

1. Access the Objects Profile Display.
2. Type U in the command line next to the profile that you want to update and press Enter.
3. The Update Object Profile window appears. You can then make changes to the profile.
4. Press PF3 when you have finished to save the changes. To cancel and exit without making changes, type CAN in the Option line and press Enter.

Viewing an object profile

You can view your own object profile or the one created by another user if the profile was created with a share option of view or update.

To view a profile:

1. Access the Objects Profile Display.
2. Type V in the command line next to the profile that you want to view and press Enter.
3. The View Objects Profile Display appears. You can view profile details, but cannot make any changes.
4. To view recovery options, enter Y in the Update Recovery Options window and press Enter. You can view recovery options, but you cannot make any changes.
5. To view index rebuild options, enter Y in the Edit Rebuild IX Options window and press Enter. You can view the index rebuild options, but cannot make any changes.

Renaming an object profile

You can use the rename line command to change the name or description of an object profile. You can rename profiles created under your user ID, regardless of the share option. You can also rename a profile created by another user if the profile was created with a share option of update.

To rename a profile:

1. Access the Objects Profile Display.
2. Type R in the command line next to the profile that you want to rename.
3. In the Rename window that pops up, type the new profile name in the Profile Name field. You can also enter a new description in the Profile Description field. The profile creator cannot be modified.
4. Press Enter when you have finished. To cancel the rename, press PF3 on the Rename Profile window.

Deleting an object profile

You can delete profiles created under your user ID, regardless of the Share option. You can also delete a profile created by another user if the profile was created with a share option of update.

To delete a profile:

1. Access the Objects Profile Display.
2. Type D in the command line next to the profile that you want to delete and press Enter.
3. To delete the profile, type Y in the Delete field of the confirmation window and press Enter. A message appears to confirm deletion. To cancel deletion, press PF3.

Editing recovery options

You can edit the recovery options from two spots, one from the Build Recovery Job option window shown in Figure 6-15 on page 133 (by entering Y next to the Edit Recovery Options option) and the other through the Build Recovery Options window (by entering U on the command line against the Object Recovery profile, as shown in Figure 6-12).

```

ARY$OPRD      ----- Objects Profile Display ----- 2008/02/15  01:22:49
Option  ==>                                         Scroll ==> PAGE
-----
Profile  Like  *
Creator  Like  PAOLOR7                               Row 1 of 1      >
-----

Cmd  Name                                     Creator  SSID  Updt
U   OBJECT RECOVERY PROFILE#1             PAOLOR7  D8F1  U
***** Bottom of Data *****

Valid Line Cmds: (B-Build, C-Creat, D-Delete, R-Rename, V-View, U-Update)

```

Figure 6-12 Update Objects Profile Display

Then enter Y next to the Edit Recovery Options option, as shown in Figure 6-13.

```

ARY$OPRU      ----- Update Object Profile Display ----- 2008/02/15  03:32:25
Option  ==>                                         Scroll ==> PAGE
-----
Creator: PAOLOR7   Name: OBJECT RECOVERY PROFILE#1   User: PAOLOR7
Share Option: U   (Upd,View,No)   Description:
DB2 Subsystem: D8F1   Update Recovery Options: Y   (Yes/No)
----- Row 1 of 1      >
      Wild Process  Include/ IX Crtr/  IX Name/      IX DB Name/
Cmd  Type Card Indexes  Exclude DB Name  TS Name      TS Crtr  Part
    TS    N    Y      INC  RJDB2    RJTS2      PAOLOR4    0
***** Bottom of Data *****

Valid Line Commands: (Add, Delete, Explode)
Primary Commands:   (Explode)

```

Figure 6-13 Update recovery options through Update Object Profile Display

Various recovery options are discussed in detail in 6.1.2, “Recover object to current using system level backup” on page 135, through 6.1.4, “Recovery object to copy (full or incremental copy or quiesce)” on page 138.

Building object recovery job and editing recovery options

Enter B on the command line against the object recovery profile that you are interested in, as shown in Figure 6-14, and press Enter.

```
ARY$OPRD      ----- Objects Profile Display ----- 2008/02/15 01:22:49
Option ==>                                         Scroll ==> PAGE
-----
Profile  Like  *
Creator  Like  PAOLOR7                               Row 1 of 1      >
-----

Cmd  Name                                     Creator  SSID  Updt
  B  OBJECT RECOVERY PROFILE#1             PAOLOR7  D8F1  U
***** Bottom of Data *****

Valid Line Cmds: (B-Build, C-Creat, D-Delete, R-Rename, V-View, U-Update)
```

Figure 6-14 Build recovery job selection from Objects Profile Display

You get to the window shown in Figure 6-15 for the build job.

```
Build Job for PAOLOR7.OBJECT RECOVERY PROFILE#1

Build Online or Batch  0          (Online/Batch)
Edit Generated Job     Y          (Yes/No)
Edit Recovery Options  Y          (Yes/No)

Build job in Dataset   PAOLOR7.JCL.CNTL
Member                LINK1

Job Cards:
==> //PAOLOR7R JOB REGION=OM,CLASS=A,
==> //          NOTIFY=&SYSUID,MSGLEVEL=(1,1),MSGCLASS=X
==> /*JOBPARM S=SC53
==> /*

Press ENTER to process or PF3 to Cancel
```

Figure 6-15 Build object recovery job

The fields in the window shown in Figure 6-15 indicate:

► **Build Online or Batch**

When you want to build the job:

- Enter 0 for online. When you choose to build the job online, System Backup and Restore Services builds the object restore job immediately.
- Enter B for batch. When you choose this option, System Backup and Restore Services builds a job that, when executed, builds the object restore job.

► **Edit Generated Job**

Type Y to view the job in an ISPF edit session after generation. If you enter N, after the job is generated you will return to the Objects Profile Display.

- ▶ **Edit Recovery Options**

Type Y if you want to edit or review the recovery options before building the JCL. When you press Enter, the Object Recovery Options window appears.

- ▶ **Build job in data set/Member**

Enter the fully qualified data set name (without quotation marks) where you want to save the generated job. This data set must exist and can be sequential or a PDS. If the data set is a PDS, enter a member name. If the member does not exist, System Backup and Restore Services will create it.

- ▶ **Job Cards**

Enter a valid job card for your site.

Ensure that a valid SLB is available:

- ▶ Make sure that the SLB was taken with the Enable Obj Restore option set to Y in the corresponding backup profile. Creating and managing backup profiles is discussed in Chapter 4, “Configuring DB2 for system backup and restore” on page 53.
- ▶ If you do not have a full image copy available for the object that you are dealing with then the last SLB taken from RE will be used by RE to restore the object to current, as described in this section.

To perform object recovery to current:

1. Go to the Edit Recovery Options window as described in “Building object recovery job and editing recovery options” on page 133.
2. If you need to edit recovery options from the window shown in Figure 6-15 on page 133, type Y in the Edit Recovery Options field to get to the window shown in Figure 6-16 on page 135.

When the profile build is performed and the recover point is To Current, RBA/LRSN, or Last Quiesce, Recovery Expert analyzes both its system backup repository and SYSIBM.SYSCOPY and generates JCL to restore the object from the first backup that it finds preceding the point in time to which you wish to restore the object. Figure 6-16 shows the options. This may be restoring table space or index data sets from system backups or from image copies, whichever is found first.

6.1.2 Recover object to current using system level backup

In this section we show how to recover a table space to the current point in time.

```

ARY$RECO      ----- Object Recovery Options ----- 2008/02/17  18:20:24
Option  ==>

-----
Creator: PAOLOR7      Name: OBJECT RECOVERY PROFILE#1      User: PAOLOR7
Share Option:      (Upd,View,No)      Description:
DB2 Subsystem: D8F1
-----

Enter the Recovery options to associate with this Object profile:

Recovery Point      ==> 1  ( 1 To Current, 2 RBA/LRSN, 3 Last Copy,
                        4 Last Incr, 5 Last Full, 6 Last Quiesce)
RBA/LRSN            ==>
Site                ==> Z  (Zparm/Local site/Recovery site)
From Offload        ==> N  (Yes/No)
Reuse (IBM recover) ==> Y  (Yes/No)
Utility ID          ==> RESTORE.CURRENT1
Parallel Tasks      ==> 04 (01 - 99)
Number of Tape Units ==> 02 (01 - 99)
Edit Rebuild IX Options ==> N  (Yes/No)

Enter END command to return to the previous window

```

Figure 6-16 Object Recovery Options for recover to current

The top portion of the window shows the object profile information. Under "Enter the Recovery options to associate with this Object Profile," specify appropriate values for the fields as discussed below:

- Recovery Point
 - Enter 1 to recover the object to the current point in time.
- RBA/LRSN
 - This field is required only for option 2.
- Site
 - Specify which copies are to be used in the recovery. Enter L for local site, R for recovery site, or Z to use the site specified in the DSNZPARM member for this DB2 subsystem.
- From Offload
 - Enter Y in this field to specify that if the generated recovery JCL will use an System Backup and Restore Services system backup, the recovery is to use an appropriate offloaded backup (either local or remote), even if the backup still resides on disk.

- ▶ Reuse (IBM recover)
Enter Y to specify that the IBM RECOVER utility is to logically reset and reuse DB2-managed data sets without deleting and redefining them. If you enter N, DB2-managed data sets are deleted and redefined.
- ▶ Utility ID
Enter a 1-to-16-character utility ID to be used to uniquely identify the utility to DB2. The ID must begin with a letter. The remainder can be alphanumeric or the following special characters: #, \$, @, ¢, !, , , .
- ▶ Parallel tasks
If you want the utility to process objects in parallel from backups on tape, enter the maximum number of objects to be processed in parallel.
- ▶ Number of tape units
If specifying processing in parallel, indicate the maximum number of tape drives to be dynamically allocated.
- ▶ Edit Rebuild IX Options
When indexes are included in the objects to be recovered, System Backup and Restore Services chooses to either recover or rebuild the index, depending on the index and the type of recovery. Enter Y in this field to set options for REBUILD INDEX.
- ▶ Online Rebuild Index (DB2 9 for z/OS and later)
Type Y in this field to specify that the REBUILD INDEX should be performed online, as shown in Figure 6-18 on page 137.
- ▶ Edit Online Rbld opts (DB2 9 for z/OS and later)
To set options for an online REBUILD INDEX, type Y in this field and press Enter.

If you have any SYSCOPY entries with ICTYPE='P' then you may see the message shown in Figure 6-17 when you press Enter to generate the job.

ARYS314W - Partial Recovery has been found. Recovery still allowed for object: ARYS324I - Tablespace DBNAME.TSNAME

Figure 6-17 Partial recovery message

You can press PF3 from here to see the JCL.

```
ARY$RECO ----- Object Recovery Options ----- 2008/02/27 19:26:12
Option ==>
-----
Creator: PAOLOR7      Name: AASDFA                      User: PAOLOR4
Share Option:      (Upd,View,No)      Description:
DB2 Subsystem: D9C1
-----
Enter the Recovery options to associate with this Object profile:

Recovery Point      ==> 1  ( 1 To Current, 2 RBA/LRSN, 3 Last Copy,
                          4 Last Incr, 5 Last Full, 6 Last Quiesce)
RBA/LRSN            ==>
Site                ==> Z  (Zparm/Local site/Recovery site)
From Offload        ==> N  (Yes/No)
Reuse (IBM recover) ==> Y  (Yes/No)
Utility ID          ==> RESTORE.CURRENT1
Parallel Tasks      ==> 04 (01 - 99)
Number of Tape Units ==> 02 (01 - 99)
Edit Rebuild IX Options ==> N  (Yes/No)
Online Rebuild Index ==> Y  (Yes/No)
  Edit Online Rbld opts ==> Y  (Yes/No)
```

Figure 6-18 Object Recovery Options panel DB2 9 for z/OS

The job generated typically has the following steps:

► Step RECOBJST

This step restores the data sets of table spaces or index spaces from a system level backup. This step executes the ARY@MAIN program and contains a control statement similar to:

```
ORESTORE "PAOLOR4"."BKUP-1"
          GENERATION 01
          DATE 02/15/2008
          TIME 04:49:20
          MAX-TASKS 04
          START-SPACE-MODE NONE
          LOCAL-SITE
          TABLESPACE RJDB2.RJTS2
          INDEXSPACE RJDB2.RJIX2
```

► Step START#UT

This step starts all the spaces being recovered in UT (utility access only) mode.

► Step: RCVRLOGO

This step recovers objects via LOGONLY that had their data sets recovered from a system level backup.

► Step: START#RW

This step starts all the spaces back to RW (read write) mode.

Submit the JCL generated to restore the object to current.

6.1.3 Recover object to point in time using system level backup

Figure 6-19 shows the selected recovery options.

```
ARY$RECO ----- Object Recovery Options ----- 2008/02/17 18:58:39
Option  ==>

-----
Creator: PAOLOR7      Name: OBJECT RECOVERY PROFILE#1      User: PAOLOR7
Share Option:  (Upd,View,No)      Description:
DB2 Subsystem: D8F1
-----

Enter the Recovery options to associate with this Object profile:

Recovery Point      ==> 2  ( 1 To Current, 2 RBA/LRSN, 3 Last Copy,
                               4 Last Incr, 5 Last Full, 6 Last Quiesce)
RBA/LRSN            ==>
Site                ==> Z  (Zparm/Local site/Recovery site)
From Offload        ==> N  (Yes/No)
Reuse (IBM recover) ==> Y  (Yes/No)
Utility ID          ==> RESTORE.LRSN1
Parallel Tasks      ==> 04 (01 - 99)
Number of Tape Units ==> 02 (01 - 99)
Edit Rebuild IX Options ==> N  (Yes/No)

Enter END command to return to the previous screen
```

Figure 6-19 Object Recovery Options for recover to RBA/LRSN

The recovery process statements generated depend on the type of backup (SLB or Imagecopy) available. We described this situation in 6.1.2, “Recover object to current using system level backup” on page 135.

In order for a table space or index space be restored using a system backup, the requested recovery point must be the RBA/LRSN of the backup + 1.

6.1.4 Recovery object to copy (full or incremental copy or quiesce)

Using the ISPF Recovery Expert interface, the options provided in recovering the objects are to current, RBA, last copy, last incremental, last full, and last quiesce without having to find when/what copy was taken.

If you want to recover directly to a specific full copy in the past, you can use the GUI interface or specify the RBA+1 of the full copy.

Figure 6-20 shows that we have selected option 3 to let DB2 Recovery Expert select the latest image copy available, either last full imagecopy or last incremental copy.

```

ARY$RECO ----- Object Recovery Options ----- "DOWN" is not active
Option ==>

-----
Creator: PAOLOR4      Name: REST-1                      User: PAOLOR4
Share Option: U (Upd,View,No)   Description: OBJECT RECOVERY PROFILE
DB2 Subsystem: D8F1
-----

Enter the Recovery options to associate with this Object profile:

Recovery Point          ==> 3 ( 1 To Current, 2 RBA/LRSN, 3 Last Copy,
                               4 Last Incr, 5 Last Full, 6 Last Quiesce)
RBA/LRSN                ==>
Site                    ==> Z (Zparm/Local site/Recovery site)
From Offload            ==> N (Yes/No)
Reuse (IBM recover)     ==> N (Yes/No)
Utility ID              ==> PAOLOR4.RECORAJ1
Parallel Tasks          ==> 01 (01 - 99)
Number of Tape Units    ==> 01 (01 - 99)
Edit Rebuild IX Options ==> N (Yes/No)

Enter END command to return to the previous screen

```

Figure 6-20 Object recovery options

DB2 Recovery Expert generates the RECOVER TOLOGPOINT utility statement to the LRSN value that corresponds to either the last full imagecopy or last incremental copy.

Choose option 4 to specify that you would like DB2 Recovery Expert to recover from the last incremental copy. DB2 Recovery Expert generates a RECOVER TOLOGPOINT utility statement to the LRSN value that corresponds to the last incremental copy.

Choose option 5 to specify that you would like DB2 Recovery Expert to recover from the last full image copy. DB2 Recovery Expert generates a RECOVER TOLOGPOINT utility statement to the LRSN value that corresponds to last full imagecopy.

Choose option 6 to specify that you would like DB2 Recovery Expert to recover to the last quiesce point. The type of backup (SLB or imagecopy) available prior to the quiesce point influences the recovery process statements generated by DB2 Recovery Expert.

6.2 Object Recovery using DB2 9 for z/OS system level backup

The backup taken by the BACKUP SYSTEM utility (system level backups) in V8 is used for a restore of the entire DB2 system. DB2 9 for z/OS has an enhancement to allow a DB2 object to be recovered from the system level backup. Recovery is via the RECOVER utility, which is now capable of using system level backups for the restore phase in addition to the usual image copies. In order to allow RECOVER to consider using system level backups, specify YES for the new SYSTEM_LEVEL_BACKUPS installation option in panel DSNTIP6. See also "BACKUP and RESTORE enhancements with DB2 9 for z/OS" on page 238. For details, refer to *DB2 Version 9.1 for z/OS Utility Guide and Reference*, SC18-9855-02.

In our scenario we want to recover a table space and related indexes from a system level backup. We take a system level backup as shown in Example 6-1.

Example 6-1 SMS output on successful system level backup

```

ARC1805I THE FOLLOWING 00004 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL DSN$DB9C$DB
ARC1805I (CONT.) SBOX5S
ARC1805I (CONT.) SBOX5T
ARC1805I (CONT.) SBOX5U
ARC1805I (CONT.) SBOX5V
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB9C$DB, AT 16:59:15 ON 2008/04/17, FUNCTION RC=0000,
MAXIMUM VOLUME RC=0000
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB9C$LG, AT 16:59:15 ON 2008/04/17,
TOKEN=X'C4F9C3F1C24281BC99549809C24274AD53E6'
ARC1805I THE FOLLOWING 00018 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL DSN$DB9C$LG
ARC1805I (CONT.) SBOX5Q
ARC1805I (CONT.) SBOX5R
ARC1805I (CONT.) SBOX5A
ARC1805I (CONT.) SBOX5B
ARC1805I (CONT.) SBOX5C
ARC1805I (CONT.) SBOX5D
ARC1805I (CONT.) SBOX5E
ARC1805I (CONT.) SBOX5F
ARC1805I (CONT.) SBOX5G
ARC1805I (CONT.) SBOX5H
ARC1805I (CONT.) SBOX5I
ARC1805I (CONT.) SBOX5J
ARC1805I (CONT.) SBOX5K
ARC1805I (CONT.) SBOX5L
ARC1805I (CONT.) SBOX5M
ARC1805I (CONT.) SBOX5N
ARC1805I (CONT.) SBOX5O
ARC1805I (CONT.) SBOX5P
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB9C$LG, AT 16:59:21 ON 2008/04/17, FUNCTION RC=0000,
MAXIMUM VOLUME RC=0000

```

We use the RECOVER utility for the partitioned table space. DB2 recognizes that the object is available from a volume of the recent system level backup and executes it. The RECOVER utility chooses the most recent backup (among possible image copies, concurrent copies, or system level backups) to restore based on the recovery point for the table spaces or indexes (with the COPY YES attribute) being recovered.

Our indexes have not been defined with the COPY YES attribute, so we add a REBUILD INDEX(ALL) as an additional step within the RECOVER.

Example 6-2 shows the RECOVER job output, which provides details on the execution of the FlashCopy restore and the log apply phase. In our case fast log apply was not enabled and log apply found no records in the log to apply.

Example 6-2 Output from a DB2 RECOVER utility that selects FLASHCOPY as input

```

IEF375I JOB/PAOLOR6P/START 2008112.1919
IEF376I JOB/PAOLOR6P/STOP 2008112.1919 CPU OMIN 00.32SEC SRB OMIN 00.00SEC
1DSNU000I 112 19:19:19.16 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR6.PAOLOR6P
DSNU1044I 112 19:19:19.29 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
ODSNU050I 112 19:19:19.38 DSNUGUTC - RECOVER TABLESPACE DSN8D91A.DSN8S91E DSNUM ALL
DSNU1520I 112 19:19:19.75 DSNUCBMT - THE RECOVERY BASE FOR TABLESPACE DSN8D91A.DSN8S91E IS THE SYSTEM LEVEL BACKUP
WITH DATE = 20080417, TIME 165921, AND TOKEN X'C4F9C3F1C24281BC99549809C24274AD53E6'
DSNU1527I 112 19:19:20.34 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 1 WAS SUCCESSFULLY RESTORED FROM A FLASHCOPY,
ELAPSED TIME=00:00:00
DSNU1527I 112 19:19:20.55 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 2 WAS SUCCESSFULLY RESTORED FROM A FLASHCOPY,
ELAPSED TIME=00:00:00
DSNU1527I 112 19:19:20.81 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 3 WAS SUCCESSFULLY RESTORED FROM A FLASHCOPY,

```

```

ELAPSED TIME=00:00:00
DSNU1527I 112 19:19:21.06 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 4 WAS SUCCESSFULLY RESTORED FROM A FLASHCOPY,
ELAPSED TIME=00:00:00
DSNU1527I 112 19:19:21.27 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 5 WAS SUCCESSFULLY RESTORED FROM A FLASHCOPY,
ELAPSED TIME=00:00:00
DSNU1511I -D9C1 112 19:19:21.47 DSNUCALA - FAST LOG APPLY WAS NOT USED FOR RECOVERY
DSNU1510I 112 19:19:21.47 DSNUCBLA - LOG APPLY PHASE COMPLETE, ELAPSED TIME = 00:00:00
DSNU500I 112 19:19:22.11 DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:02
ODSNU050I 112 19:19:22.12 DSNUGUTC - REBUILD INDEX(ALL) TABLESPACE DSN8D91A.DSN8S91E SORTDEVT SYSDA SORTNUM 4
DSNU395I 112 19:19:22.37 DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 6
DSNU555I -D9C1 112 19:19:22.45 DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=0
DSNU555I -D9C1 112 19:19:22.48 DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=84
DSNU705I 112 19:19:22.48 DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:00
DSNU394I -D9C1 112 19:19:23.89 DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=42 FOR INDEX DSN8910.XEMP2
DSNU394I -D9C1 112 19:19:23.97 DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=42 FOR INDEX DSN8910.XEMP1
DSNU391I 112 19:19:23.98 DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 2
DSNU392I 112 19:19:23.98 DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:01
DSNU010I 112 19:19:24.13 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
ICE091I 0 OUTPUT LRECL = 12, TYPE = F
ICE080I 0 IN MAIN STORAGE SORT
ICE055I 0 INSERT 42, DELETE 42
ICE054I 0 RECORDS - IN: 0, OUT: 0
ICE134I 0 NUMBER OF BYTES SORTED: 504
ICE165I 0 TOTAL WORK DATA SET TRACKS ALLOCATED: 0 , TRACKS USED: 0
ICE199I 0 MEMORY OBJECT STORAGE USED = 0M BYTES
ICE180I 0 HIPERSPACE STORAGE USED = 0K BYTES
ICE188I 0 DATA SPACE STORAGE USED = 0K BYTES
ICE052I 0 END OF DFSORT

```

In Example 6-3 we list the output from SMS on the FlashCopy FRR execution, which lists the five partitions of the table space restored by fast replication from COPYPOOL=DSN\$DB9C\$DB.

Example 6-3 DFSHSms output - RECOVER TS with 5 data sets selecting FlashCopy backup

```

ARC1801I FAST REPLICATION DATA SET RECOVERY IS 597
ARC1801I (CONT.) STARTING FOR DATA SET
ARC1801I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A001, AT
ARC1801I (CONT.) 19:19:19 ON 2008/04/21
$HASP100 IEESYSAS ON STCINRDR
IEF196I IEF237I D51C ALLOCATED TO SYS00183
$HASP373 IEESYSAS STARTED
IEF403I IEESYSAS - STARTED - TIME=19.19.19 - ASID=0097 - SC63
IEF196I IEF285I SYS1.CSSLIB KEPT
IEF196I IEF285I VOL SER NOS= Z19RE1.
ARC1861I THE FOLLOWING 0001 DATA SET(S) WERE 604
ARC1861I (CONT.) SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA
ARC1861I (CONT.) SET RECOVERY:
ARC1861I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A001, COPYPOOL=DSN$DB9C$DB,
DEVTYPE=DASD
ARC1802I FAST REPLICATION DATA SET RECOVERY HAS 606
ARC1802I (CONT.) COMPLETED FOR DATA SET
ARC1802I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A001, AT
ARC1802I (CONT.) 19:19:20 ON 2008/04/21, FUNCTION RC=0000, MAXIMUM
ARC1802I (CONT.) DATA SET RC=0000
ARC1801I FAST REPLICATION DATA SET RECOVERY IS 607
ARC1801I (CONT.) STARTING FOR DATA SET
ARC1801I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A002, AT
ARC1801I (CONT.) 19:19:20 ON 2008/04/21
ARC1861I THE FOLLOWING 0001 DATA SET(S) WERE 608
ARC1861I (CONT.) SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA
ARC1861I (CONT.) SET RECOVERY:

```

```

ARC1861I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A002, COPYPOOL=DSN$DB9C$DB,
DEVTYPE=DASD
ARC1802I FAST REPLICATION DATA SET RECOVERY HAS 610
ARC1802I (CONT.) COMPLETED FOR DATA SET
ARC1802I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A002, AT
ARC1802I (CONT.) 19:19:20 ON 2008/04/21, FUNCTION RC=0000, MAXIMUM
ARC1802I (CONT.) DATA SET RC=0000
ARC1801I FAST REPLICATION DATA SET RECOVERY IS 611
ARC1801I (CONT.) STARTING FOR DATA SET
ARC1801I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A003, AT
ARC1801I (CONT.) 19:19:20 ON 2008/04/21
ARC1861I THE FOLLOWING 0001 DATA SET(S) WERE 612
ARC1861I (CONT.) SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA
ARC1861I (CONT.) SET RECOVERY:
ARC1861I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A003, COPYPOOL=DSN$DB9C$DB,
DEVTYPE=DASD
ARC1802I FAST REPLICATION DATA SET RECOVERY HAS 614
ARC1802I (CONT.) COMPLETED FOR DATA SET
ARC1802I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A003, AT
ARC1802I (CONT.) 19:19:20 ON 2008/04/21, FUNCTION RC=0000, MAXIMUM
ARC1802I (CONT.) DATA SET RC=0000
ARC1801I FAST REPLICATION DATA SET RECOVERY IS 615
ARC1801I (CONT.) STARTING FOR DATA SET
ARC1801I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A004, AT
ARC1801I (CONT.) 19:19:20 ON 2008/04/21
ARC1861I THE FOLLOWING 0001 DATA SET(S) WERE 616
ARC1861I (CONT.) SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA
ARC1861I (CONT.) SET RECOVERY:
ARC1861I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A004, COPYPOOL=DSN$DB9C$DB,
DEVTYPE=DASD
ARC1802I FAST REPLICATION DATA SET RECOVERY HAS 618
ARC1802I (CONT.) COMPLETED FOR DATA SET
ARC1802I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A004, AT
ARC1802I (CONT.) 19:19:21 ON 2008/04/21, FUNCTION RC=0000, MAXIMUM
ARC1802I (CONT.) DATA SET RC=0000
ARC1801I FAST REPLICATION DATA SET RECOVERY IS 619
ARC1801I (CONT.) STARTING FOR DATA SET
ARC1801I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A005, AT
ARC1801I (CONT.) 19:19:21 ON 2008/04/21
ARC1802I FAST REPLICATION DATA SET RECOVERY HAS 618
ARC1802I (CONT.) COMPLETED FOR DATA SET
ARC1802I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A004, AT
ARC1802I (CONT.) 19:19:21 ON 2008/04/21, FUNCTION RC=0000, MAXIMUM
ARC1802I (CONT.) DATA SET RC=0000
ARC1801I FAST REPLICATION DATA SET RECOVERY IS 619
ARC1801I (CONT.) STARTING FOR DATA SET
ARC1801I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A005, AT
ARC1801I (CONT.) 19:19:21 ON 2008/04/21
IEF196I IEF237I D51C ALLOCATED TO SYS00184
IEF196I IEF285I   SYS1.LINKLIB                               KEPT
IEF196I IEF285I   VOL SER NOS= Z19RE1.
ARC1861I THE FOLLOWING 0001 DATA SET(S) WERE 623
ARC1861I (CONT.) SUCCESSFULLY PROCESSED DURING FAST REPLICATION DATA
ARC1861I (CONT.) SET RECOVERY:
ARC1861I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A005, COPYPOOL=DSN$DB9C$DB,
DEVTYPE=DASD
ARC1802I FAST REPLICATION DATA SET RECOVERY HAS 625
ARC1802I (CONT.) COMPLETED FOR DATA SET
ARC1802I (CONT.) DB9CD.DSNDBC.DSN8D91A.DSN8S91E.I0001.A005, AT
ARC1802I (CONT.) 19:19:21 ON 2008/04/21, FUNCTION RC=0000, MAXIMUM

```

6.3 Object recovery with Recovery Expert using system level backup

In Example 6-4 we list the full output of the job steps created by DB2 Recovery Expert under a DB2 9 for z/OS environment. The setup follows the same steps as described in 6.1, "Object recovery with Recovery Expert under DB2 V8" on page 126.

The job steps have been created for the same partitioned table space and indexes of 6.2, "Object Recovery using DB2 9 for z/OS system level backup" on page 139.

Example 6-4 Job steps created by Recovery Expert for object recovery

```

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
21.50.17 JOB02621 ---- MONDAY,    21 APR 2008 ----
21.50.17 JOB02621 IRR010I  USERID PAOLOR6  IS ASSIGNED TO THIS JOB.
21.50.17 JOB02621 ICH70001I PAOLOR6  LAST ACCESS AT 21:42:50 ON MONDAY, APRIL 21, 2008
21.50.17 JOB02621 $HASP373 PAOLOR60  STARTED - INIT 1    - CLASS A - SYS SC64
21.50.17 JOB02621 IEF403I  PAOLOR60  - STARTED - TIME=21.50.17 - ASID=0032 - SC64
21.50.17 JOB02621 -
21.50.17 JOB02621 -JOBNAME  STEPNAME  PROCSTEP  RC  EXCP  CPU  SRB  CLOCK  SERV PG  PAGE  SWAP  VIO SWAPS
STEPNO
21.50.17 JOB02621 -PAOLOR60          START#UT  00   162   .00   .00   .00   775
0    0    0    0    0    1
21.50.18 JOB02621 -PAOLOR60          RCVRFRIC  00   596   .00   .00   .02  1451
0    0    0    0    0    2
21.50.23 JOB02621 -PAOLOR60          REBUILD   00   857   .00   .00   .07  33974
0    3    0    0    0    3
21.50.23 JOB02621 -PAOLOR60          START#RW  00   159   .00   .00   .00   758
0    0    0    0    0    4
21.50.23 JOB02621 IEF404I  PAOLOR60  - ENDED - TIME=21.50.23 - ASID=0032 - SC64
21.50.23 JOB02621 -PAOLOR60 ENDED.  NAME=DB2 UTILITY          TOTAL CPU TIME=   .00 TOTAL ELAPSED TIME=   .10
21.50.23 JOB02621 $HASP395 PAOLOR60  ENDED
0----- JES2 JOB STATISTICS -----
- 21 APR 2008 JOB EXECUTION DATE
- 134 CARDS READ
- 426 SYSOUT PRINT RECORDS
- 0 SYSOUT PUNCH RECORDS
- 26 SYSOUT SPOOL KBYTES
- 0.10 MINUTES EXECUTION TIME
1 //PAOLOR60 JOB (ACCNT#),'DB2 UTILITY',                      JOB02621
//      REGION=QM,NOTIFY=&SYSUID,
//      MSGCLASS=X,
//      CLASS=A
/*JOBPARM SYSAFF=SC64
/*
/******
/*
/* Profile:  PAOLOR6.D9C2 SINGLE OBJECT
/* Job:      01 of 01
/* Desc:     DB DSN8D91A
/* User:     PAOLOR6
/* Date:     Monday 21, 2008
/* Time:     21:41:10.71
/*
/******
/*
/******

```

```

/*
/* Step:      START#UT
/*
/* Desc:      This step will start all the spaces being recovered
/*              in UT (Utility Access Only) mode.
/*
/*** * * * * *
/*
IEFC653I SUBSTITUTION JCL - (ACCNT#),'DB2UTILITY',REGION=OM,NOTIFY=PAOLOR6,MSGCLASS=X,CLASS=A
2 //START#UT EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
3 //STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
4 //          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
5 //SYSTSPRT DD SYSOUT=*
6 //SYSPRINT DD SYSOUT=*
7 //SYSYSIN DD *
/*
/*** * * * * *
/*
/* Step:      RCVRFRIC
/*
/* Desc:      This step will recover objects requiring a DB2
/*              image copy (or log records) as the starting point
/*              of the recovery.
/*
/*** * * * * *
/*
8 //RCVRFRIC EXEC PGM=DSNUTILB,REGION=006M,COND=(4,LT),
//          PARM=(D9C2)
/*
9 //STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
10 //         DD DISP=SHR,DSN=DB9C9.SDSNLOAD
11 //SYSPRINT DD SYSOUT=*
12 //SYSOUT DD SYSOUT=*
13 //UTPRINT DD SYSOUT=*
/*
14 //SYSIN DD *
/*
/*** * * * * *
/*
/* Step:      REBUILD
/*
/* Desc:      This step will rebuild all indexes that could not
/*              be recovered via the RECOVER Utility.
/*
/*** * * * * *
/*
15 //REBUILD EXEC PGM=DSNUTILB,REGION=006M,COND=(4,LT),
//          PARM=(D9C2)
/*
16 //STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
17 //         DD DISP=SHR,DSN=DB9C9.SDSNLOAD
18 //SYSPRINT DD SYSOUT=*
19 //SYSOUT DD SYSOUT=*
20 //UTPRINT DD SYSOUT=*
/*
21 //SYSIN DD *
/*
/*** * * * * *
/*
/* Step:      START#RW
/*
/* Desc:      This step will start all the spaces back to RW (Read
/*              Write) mode.
/*
/*** * * * * *
/*

```



```

22 //START#RW EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
23 //STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT
24 // DD DISP=SHR,DSN=DB9C9.SDSNLOAD
25 //SYSTSPRT DD SYSOUT=*
26 //SYSPRINT DD SYSOUT=*
27 //SYSTSIN DD *
    /*
...
...
IREADY
  DSN SYSTEM(D9C2)
  DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(001) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.DSN8S91E.0001 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(002) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.DSN8S91E.0002 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(003) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.DSN8S91E.0003 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(004) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.DSN8S91E.0004 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(005) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.DSN8S91E.0005 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(XEMP1) PART(001) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP1 .0001 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(XEMP2) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP2 .0001 IS CURRENTLY START UT.
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP2 .0002 IS CURRENTLY START UT.
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP2 .0003 IS CURRENTLY START UT.
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP2 .0004 IS CURRENTLY START UT.
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP2 .0005 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(XEMP1) PART(002) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP1 .0002 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(XEMP1) PART(003) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP1 .0003 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(XEMP1) PART(004) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP1 .0004 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN -STA DB(DSN8D91A) SP(XEMP1) PART(005) ACCESS(UT)
  DSNT309I -D9C2 DSNTDCST PARTITION DSN8D91A.XEMP1 .0005 IS CURRENTLY START UT.
  DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
  DSN END
1DSNU000I 112 21:50:17.33 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR6.PAOLOR60
  DSNU1044I 112 21:50:17.37 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
ODSNU050I 112 21:50:17.37 DSNUGUTC - RECOVER TABLESPACE DSN8D91A.DSN8S91E DSNUM(1) TABLESPACE DSN8D91A.DSN8S91E
  DSNUM(2) TABLESPACE DSN8D91A.DSN8S91E DSNUM(3) TABLESPACE DSN8D91A.DSN8S91E DSNUM(4)
TABLESPACE DSN8D91A.DSN8S91E DSNUM(5) PARALLEL(4) TAPEUNITS(2) REUSE LOCALSITE
  DSNU427I 112 21:50:17.43 DSNUCBMT - OBJECTS WILL BE PROCESSED IN PARALLEL, NUMBER OF OBJECTS = 4
DSNU1520I 112 21:50:17.43 DSNUCBMT - THE RECOVERY BASE FOR TABLESPACE DSN8D91A.DSN8S91E DSNUM 1 IS THE SYSTEM LEVEL
  BACKUP WITH DATE = 20080417, TIME 165921, AND TOKEN X'C4F9C3F1C24281BC99549809C24274AD53E6'
  DSNU1527I 112 21:50:17.68 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 1 WAS
SUCCESSFULLY RESTORED FROM A FLASHCOPY, ELAPSED TIME=00:00:00
  DSNU1520I 112 21:50:17.68 DSNUCBMT - THE RECOVERY BASE FOR TABLESPACE DSN8D91A.DSN8S91E DSNUM 2 IS THE SYSTEM LEVEL
  BACKUP WITH DATE = 20080417, TIME 165921, AND TOKEN X'C4F9C3F1C24281BC99549809C24274AD53E6'
  DSNU1527I 112 21:50:17.84 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 2 WAS
SUCCESSFULLY RESTORED FROM A FLASHCOPY, ELAPSED TIME=00:00:00
  DSNU1520I 112 21:50:17.84 DSNUCBMT - THE RECOVERY BASE FOR TABLESPACE DSN8D91A.DSN8S91E DSNUM 3 IS THE SYSTEM LEVEL
  BACKUP WITH DATE = 20080417, TIME 165921, AND TOKEN X'C4F9C3F1C24281BC99549809C24274AD53E6'
  DSNU1527I 112 21:50:18.06 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 3 WAS

```

```

SUCCESSFULLY RESTORED FROM A FLASHCOPY, ELAPSED TIME=00:00:00
DSNU1520I 112 21:50:18.06 DSNUCBMT - THE RECOVERY BASE FOR TABLESPACE DSN8D91A.DSN8S91E DSNUM 4 IS THE SYSTEM LEVEL
BACKUP WITH DATE = 20080417, TIME 165921, AND TOKEN X'C4F9C3F1C24281BC99549809C24274AD53E6'
DSNU1527I 112 21:50:18.24 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 4 WAS
SUCCESSFULLY RESTORED FROM A FLASHCOPY, ELAPSED TIME=00:00:00
DSNU1520I 112 21:50:18.24 DSNUCBMT - THE RECOVERY BASE FOR TABLESPACE DSN8D91A.DSN8S91E DSNUM 5 IS THE SYSTEM LEVEL
BACKUP WITH DATE = 20080417, TIME 165921, AND TOKEN X'C4F9C3F1C24281BC99549809C24274AD53E6'
DSNU1527I 112 21:50:18.42 DSNUCBMT - TABLESPACE DSN8D91A.DSN8S91E DSNUM 5 WAS
SUCCESSFULLY RESTORED FROM A FLASHCOPY, ELAPSED TIME=00:00:00
DSNU1511I -D9C2 112 21:50:18.56 DSNUCALA - FAST LOG APPLY WAS NOT USED FOR RECOVERY
DSNU1510I 112 21:50:18.56 DSNUCBLA - LOG APPLY PHASE COMPLETE, ELAPSED TIME = 00:00:00
DSNU500I 112 21:50:18.67 DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:01
DSNU010I 112 21:50:18.68 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
DSNU000I 112 21:50:18.75 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PAOLOR6.PAOLOR60
DSNU1044I 112 21:50:18.79 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I 112 21:50:18.79 DSNUGUTC - REBUILD INDEX(ALL) TABLESPACE DSN8D91A.DSN8S91E SORTDEVT SYSDA SORTNUM 4
DSNU395I 112 21:50:19.05 DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 6
DSNU555I -D9C2 112 21:50:19.14 DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=0
DSNU555I -D9C2 112 21:50:19.21 DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=84
DSNU705I 112 21:50:19.21 DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:00
DSNU394I -D9C2 112 21:50:23.14 DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=42 FOR INDEX DSN8910.XEMP2
DSNU394I -D9C2 112 21:50:23.29 DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=42 FOR INDEX DSN8910.XEMP1
DSNU391I 112 21:50:23.29 DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 2
DSNU392I 112 21:50:23.29 DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:04
DSNU010I 112 21:50:23.36 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
IREADY
DSN SYSTEM(D9C2)
DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(001) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(002) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(003) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(004) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(DSN8S91E) PART(005) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(XEMP1) PART(001) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(XEMP2) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(XEMP1) PART(002) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(XEMP1) PART(003) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(XEMP1) PART(004) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN -STA DB(DSN8D91A) SP(XEMP1) PART(005) ACCESS(RW)
DSN9022I -D9C2 DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN END
...
...
1

```

The sample output shows how Recovery Expert automatically built the steps for:

- ▶ Starting all the spaces being recovered in UT (utility access only) mode
- ▶ Recovering the table space
- ▶ Rebuilding all indexes that could not be recovered via the RECOVER utility
- ▶ Starting all the spaces back to RW (read write) mode

6.4 PPRC/XRC restriction for using data set level recover from FlashCopy

If the production volumes have been defined as PPRC or XRC primary, there are some restrictions. They are the same as described in APAR PK23288 for CHECK INDEX SHARELEVEL CHANGE.

Since DFSHSM does not currently have support for FlashCopy to a PPRC or XRC Primary volume, the attempt to do a FlashCopy fails and standard I/O is used. If the recovery always needs to be done to a PPRC or XRC Primary volume, then SETSYS FASTREPLICATION(DSR(NONE)) should be issued before. This eliminates the overhead of always failing the FlashCopy and then dropping back to standard I/O.

6.5 Optimal recovery action with DB2 Recovery Expert

In this section we describe how DB2 Recovery Expert selects the optimal solution for a given recovery situation during object level recovery via an ISPF Interface. The diagram in Figure 6-21 depicts the sequence of events.

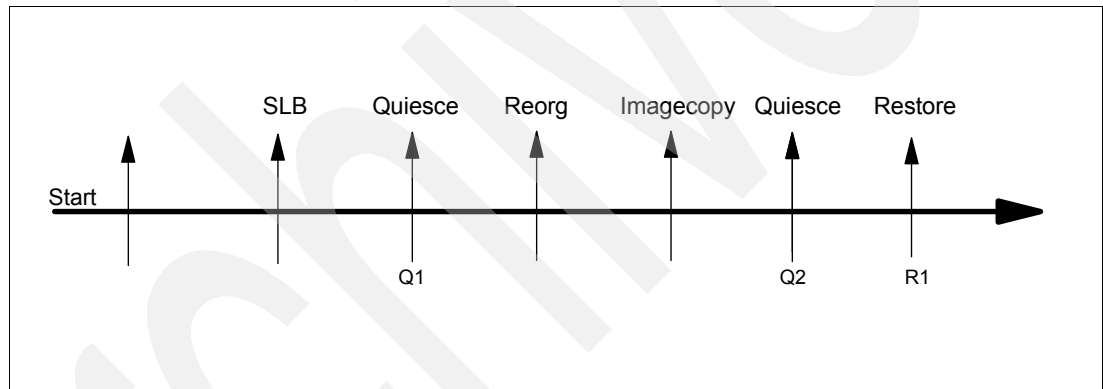


Figure 6-21 Object recovery - scenario 1

In this restore scenario 1 (R1), when the object is recovered to the quiesce point (Q2), Recovery Expert detects an imagecopy from the logs and RECOVERs to the LOGPOINT that corresponds to the LRSN of Q2. This involves applying the imagecopy and rolling forward through the logs to the LRSN of Q2.

Though there is an SLB prior to the imagecopy, it is expensive to use SLB and roll through the logs. The best solution in this case is to use the imagecopy and roll through the logs as demonstrated in this restore scenario (R1).

In restore scenario 2 (R2), shown in Figure 6-22, the object needs to be restored to the quiesce point (Q2). DB2 Recovery Expert detected an SLB closer to the recovery point (Q2) and restores the object to the SLB. Following this, it rolls forward through the logs to complete the recovery to the LRSN of Q2.

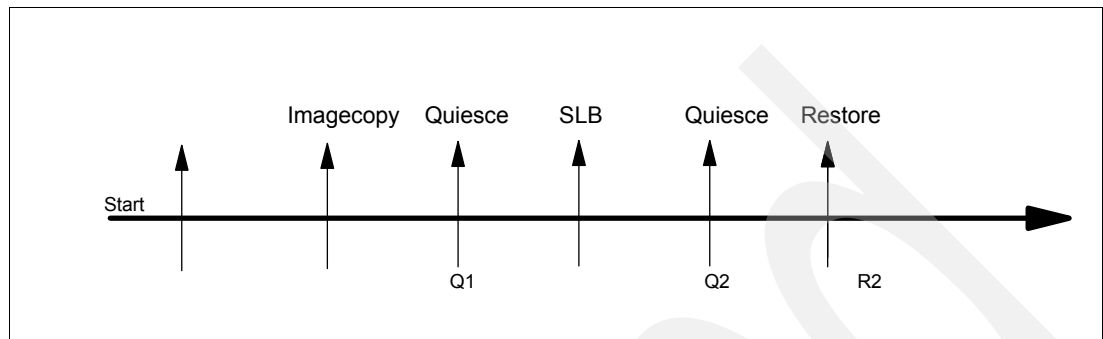


Figure 6-22 Object recovery - scenario 2

Though there was a full imagecopy available prior to the SLB, there could be a large amount of log records between these two points, and hence picking up the SLB would be the optimal recovery solution for the recovery scenario 2 (R2).

Note: The SLBs shown in this scenario are taken from RE V2.1 with the Enable Obj Restore option set to Y on the corresponding backup profiles.

Recovery Expert verifies the availability of the components for the possible solutions and chooses the most productive. If a copy has become unavailable at the execution time, Recovery Expert will automatically revert to the previous copy.

Managing disaster recovery

IBM provides a broad range of functions to be used for disaster recovery of DB2 subsystems. Some are general functions applicable to the entire 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. For background information about the various techniques of disaster recovery, refer to *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370.

The basic functions and the off-site recovery procedures for DB2 for z/OS are documented in “Performing remote site recovery from a disaster at a local site” in the *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840.

In this part we discuss how Recovery Expert can help you manage DB2’s disaster recovery functions with two DB2-based techniques:

- ▶ Chapter 7, “Traditional image copy recovery” on page 151
- ▶ Chapter 8, “System level backup based remote recovery” on page 173

Archived

Traditional image copy recovery

In this chapter we describe the traditional procedures for DB2 subsystem backup (dump and image copy) and recovery at a recovery site using DB2 Recovery Expert for z/OS V2.1. 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, the 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 Version 9.1 for z/OS Administration Guide*, SC18-9840
- ▶ *DB2 Version 9.1 for z/OS Utility Guide and Reference*, SC18-9855

7.1 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 7-1, 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.

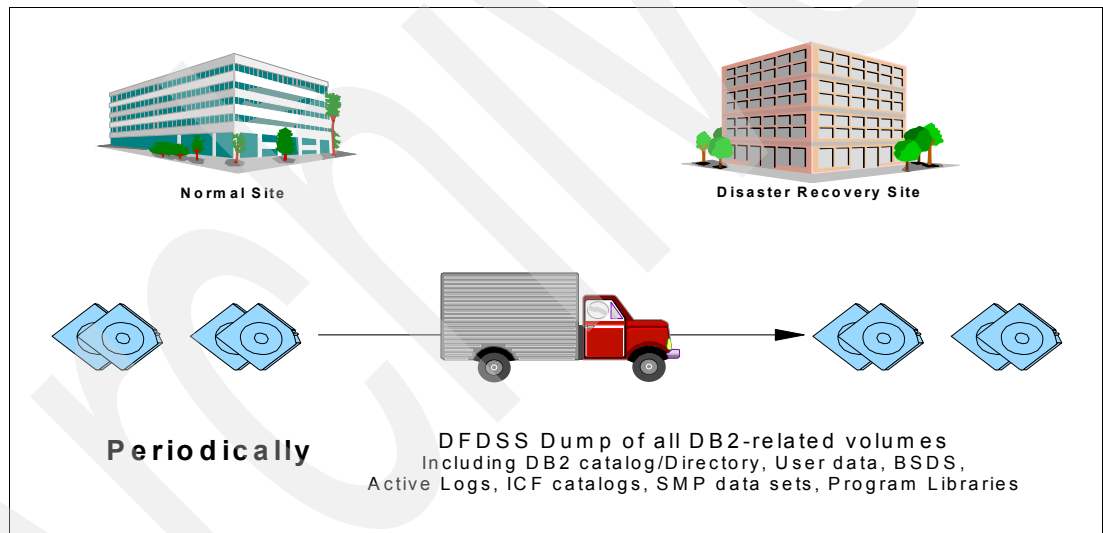


Figure 7-1 Dump and restore of DB2 environment

In Figure 7-2, on a more frequent basis, for example, *nightly*, at least, we additionally consider taking and shipping image copies of the DB2 catalog, the DB2 directory, user table spaces, and the integrated catalog facility (ICF) catalog backup. We also force the log to archive in order to be able to recover to the point in time when the archive log was created.

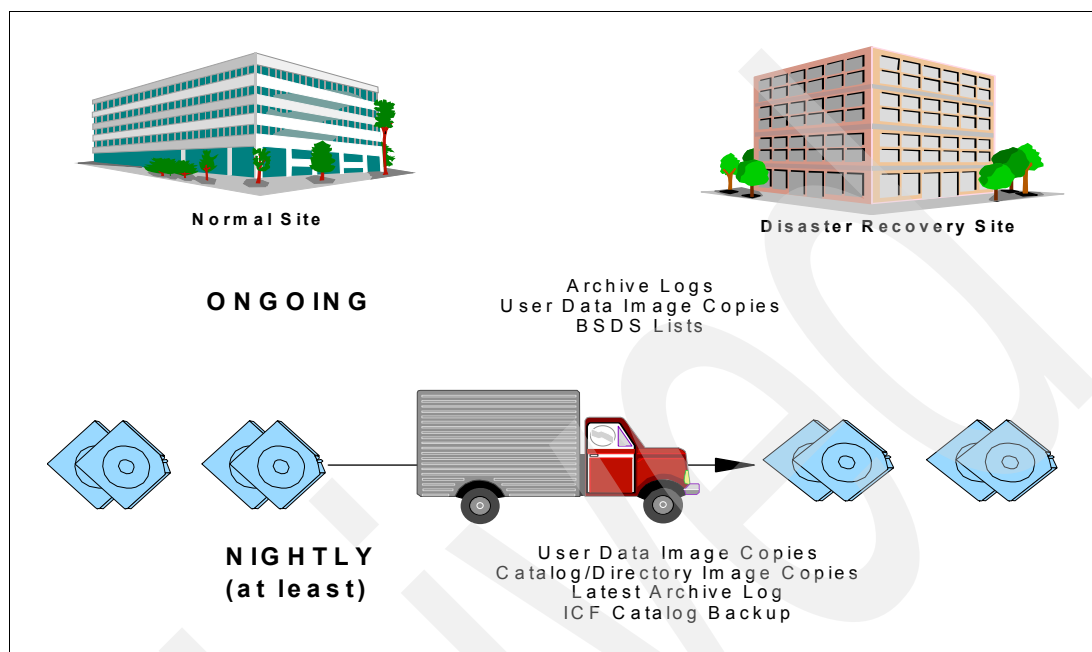


Figure 7-2 A daily business approach

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.

With a daily business approach that includes a nightly basis and an ongoing basis (see Figure 7-2), 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.

7.2 Planning for disaster recovery

There are multiple ways of preparing for disaster recovery. Independent of the way that 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.

7.2.1 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.

7.2.2 Consistent point-in-time recovery

With this approach, the DB2 environment is reestablished, whereby the DB2 Catalog, the DB2 Directory, and the user table 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 Figure 7.2.1 on page 153):

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

Run the COPY utility with RECOVERYDDN (in most cases, 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.

7.2.3 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 table spaces to the recovery site. Whenever an archive log data set is created, you also send a copy of it and the BSDS list to the recovery site. In case of disaster, the data will be recovered up to date through the last archive sent.

In 7.4, “Recovery procedures” on page 162, we 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 the previous method (see 7.2.2, “Consistent point-in-time recovery” on page 154), 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 the COPY utility with RECOVERYDDN (in most cases on tape). If you have a lot of changes to the DB2 catalog and the DB2 directory in the daytime, 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 the 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 that you need to restore from and the RBA that you will need to give in the 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.

7.3 Preparing for recovery at the local site

DB2 Recovery Expert for z/OS provides support for both consistent point in time and continuous archive log transfer recoveries at a remote location. Both methods are supported using profiles that are created and maintained using the DB2 Recovery Expert for z/OS ISPF client.

7.3.1 Create the disaster recovery profiles

We now create the disaster recovery profiles using DB2 Recovery Expert for z/OS. This is the first step in preparing for remote site recovery using DB2 Recovery Expert for z/OS. Two profiles are necessary in order to support the continuous archive log transfer method of recovery. We walk through creating both profiles in this section.

Create the primary disaster recovery profile

Figure 7-3 on page 157 shows the Update Disaster Recovery Profile window. The window contains the following fields:

- Disaster Recovery Method

Set this field to I, indicating that the recoveries will use traditional DB2 image copies.

► Archive Logs used at DR

Set this field to C to have System Backup and Restore Services create copies of the DB2 archive logs that will be used at the disaster recovery site. This option allows us to leave our original archive logs at the local site where they will be available for recoveries or other processing. Once we select this option, we must also fill out the following fields on the window:

– Copy Local site Logs

Specify which archive logs should be used to create the copies for use at the disaster recovery site.

– Unit for copying archive logs

Specify the device type for the copies of the archive logs.

– DR Archive Log Prefix 1

The data set name prefix to be used when creating copy 1 of the archive logs. This prefix will replace the prefix used when the original archive log was created.

– DR Archive Log Prefix 2

The data set name prefix to be used when creating copy 2 of the archive logs. This prefix will replace the prefix used when the original archive log was created.

Note: If the archive logs are not copied for use at the DR site, the fields in the list above are ignored.

► Force a checkpoint before Archiving

Type Y to force a checkpoint before archiving the logs. System Backup and Restore Services issues a SET LOG LOGLOAD (0) command when this field is set to Y.

► Force the Active log to Archive

This field allows us to force an ARCHIVE LOG command. The current active log is archived when this field is set to Y. For a data sharing group, each member's logs are archived individually. If a member of a data sharing group is excluded via the Process Datasharing Subsystems field, that member's log is not archived.

► Only run Archive Log Update Process

The value in this field must be set to N for the primary profile.

Note: Using a value of Y causes System Backup and Restore Services to only update the archive logs, and back up and rebuild the BSDS. This activity must be performed in a secondary disaster recovery profile.

► Process Datasharing Subsystems

This field allows you to choose how to process archive log subtasks for data sharing group members.

– A- Process all data sharing group members on all LPARs.

– S - Process only the SSID under which the recovery profile is being built.

This profile must also be generated under this SSID. If you choose this setting, you should ensure that only one of the profiles built for the data sharing group forces a checkpoint and archives the active log. The other profiles should have the Only run Archive Log Update Process field set to Y. This avoids duplicating work already done by the DR job on other subsystems.

- L - Process only the data sharing group members on the current LPAR (under which the profile is being built)

If you are building profiles separately for the other LPARs in the data sharing group, then only one of the profiles needs to force a checkpoint and archive the active log. The other profiles should have the Only run Archive Log Update Process field set to Y. This avoids duplicating work already done by the DR job on other subsystems.

Note: We recommend that when processing data sharing groups, the profiles specify A for all members of the group.

For details on the fields in Figure 7-3, see “Creating disaster recovery profiles” in *DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1*, SC19-1222.

```

ARY$DPRU      ---- Update Disaster Recovery Profile ---- 2008/02/26  13:12:08
Option ==>

-----
Creator: PAOLOR6      Name: D9CG                                User: PAOLOR6
Share Option: U      (Upd,View,No)      Description:
DB2 Subsystem: D9C1
-----
Disaster Recovery Method      ==> I      (Image copy/System backup)
Archive Logs used at DR      ==> C      (Copied/1/2)
Copy Localsite Logs          ==> B      (1/2/Both/Create 2 copies from 1)
Force a checkpoint before Archiving ==> Y      (Yes/No)
Force the Active log to Archive ==> Y      (Yes/No)
Only run Archive Log Update Process ==> N      (Yes/No)
Process Datasharing Subsystems ==> A      (All, Ssid, Lpar)
Archive Logs needed at DR      ==> 014 (days) and/or 000 (hours)
Copy Archive Logs to DASD      ==> 007 (days) and/or 000 (hours)
Unit for copying Archive Logs  ==> CART
DR Archive Log Prefix 1        ==> D9C1.ARCHLOG1.DR
DR Archive Log Prefix 2        ==> D9C1.ARCHLOG2.DR
Image copy Options
Image Copies (or SLB) used at DR ==> R      (Localsite/Recoverysite)
Catalog x days of Image Copies at DR ==> 007 (0-365)

Enter END command to Update Disaster Recover Profile

```

Figure 7-3 Primary DR profile detailed definition window

Create the secondary disaster recovery profile

Note: This step is only necessary if using the *continuous archive log transfer* method of recovery. Only a primary profile is needed for a consistent point-in-time recovery.

The secondary disaster recovery profile must be built using nearly the same specifications as those used in the primary profile with the following exceptions:

- Force a checkpoint before Archiving
This value should be set to N in the secondary profile.
- Force the Active log to Archive
This value should be set to N in the secondary profile.

- Only run Archive Log Update Process

The value in this field must be set to Y in the secondary profile.

Figure 7-4 shows the settings for our secondary profile.

```

ARY$DPRU      ---- Update Disaster Recovery Profile ---- 2008/02/26 13:12:08
Option ==>

-----
Creator: PAOLOR6      Name: D9CG                                User: PAOLOR6
Share Option: U      (Upd,View,No)      Description:
DB2 Subsystem: D9C1
-----
Disaster Recovery Method      ==> I      (Image copy/System backup)
Archive Logs used at DR      ==> C      (Copied/1/2)
Copy Localsite Logs          ==> B      (1/2/Both/Create 2 copies from 1)
Force a checkpoint before Archiving ==> N      (Yes/No)
Force the Active log to Archive ==> N      (Yes/No)
Only run Archive Log Update Process ==> Y      (Yes/No)
Process Datasharing Subsystems ==> A      (All, Ssid, Lpar)
Archive Logs needed at DR      ==> 014 (days) and/or 000 (hours)
Copy Archive Logs to DASD      ==> 007 (days) and/or 000 (hours)
Unit for copying Archive Logs  ==> CART
DR Archive Log Prefix 1        ==> D9C1.ARCHLOG1.DR
DR Archive Log Prefix 2        ==> D9C1.ARCHLOG2.DR
Image copy Options
Image Copies (or SLB) used at DR ==> R      (Localsite/Recoverysite)
Catalog x days of Image Copies at DR ==> 007 (0-365)

Enter END command to Update Disaster Recover Profile

```

Figure 7-4 Secondary DR profile detailed definition window

7.3.2 Create the application recovery jobs

Since we have elected to use traditional DB2 image copies, we need to be sure that we have created recovery jobs for all of the applications. There are many options for creating the recovery jobs. The jobs can be created using DB2 Recovery Expert for z/OS object profile as described in 6.5, “Optimal recovery action with DB2 Recovery Expert” on page 147.

The jobs can also be created using the DB2 Automation Tool. There is also recovery JCL created during the profile build process. This JCL is automatically shipped to the remote site when the PDS created by building profiles is shipped.

7.3.3 Build the disaster recovery profiles

In this section we discuss building the disaster recovery profiles.

Building the primary profile

The process of building the primary disaster recovery profile should be a regularly scheduled batch job.

Note: Disaster recovery profiles can only be built using the batch build process.

The build should be scheduled as frequently as we want to establish a point of consistency for recovery. This can be daily or weekly depending on recovery objectives. The output from the build job is a PDS containing the JCL necessary to recover the DB2 subsystem at the remote site.

Note: A PDS/E should be used to hold the output from the build jobs. This eliminates the need to frequently compress the output data set.

When the primary profile is built, System Backup and Restore Services performs the following tasks:

1. Searches the catalog and finds all appropriate image copies that will be needed at the recovery site.
2. Generates control cards to catalog the image copies at the DR site.
3. Creates the copies and the archive logs to be used at the DR site if requested in the profile.
4. Generates an updated BSDS for use at the DR site with the copied archive log names and stores the resulting file in the PDS to be shipped to the recovery site. The PDS that contains the DR jobs and control records will contain the contents of the BSDS. This eliminates the need to mount tapes at the recovery site to build the BSDS. Also, the BSDS is updated with the new archive log name created if the archive logs were copied.
5. Generates IDCAMS control cards to delete all table space and index space data sets and to create all user-defined table spaces at the DR site. It uses the current table space and index space data set attributes to ensure that the data sets are created with the same attributes at the DR site.

Note: A copy of the BSDS is created prior to any updates by System Backup and Restore Services. The original BSDS is never updated by System Backup and Restore Services.

The PDS is now ready to be shipped to the recovery site along with the application image copies and the newly created copies of the archive logs. Figure 7-5 on page 160 shows the JCL generated to build the primary profile.

```

/** * * * * *
/**
/** Step:    ARY@BULD
/**
/** Desc:    This job will generate the JCL for DR profile
/**          PAOLOR6.D9CGIC in a batch mode.
/**          The generated job will be placed in dataset
/**          PAOLOR6.IC.JCL.
/**
/** Return Codes:
/**
/**
/** (00) - Disaster Recovery Jobs built successfully
/** (12) - Problem occurred during the DR build process
/**
/** * * * * *
/** Run ARY DR Build
/** * * * * *
//ARY@BULD EXEC PGM=ARY@BULD,REGION=006M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1MO.SARYLOAD
//          DD DISP=SHR,DSN=DB9C9.SDSNEXIT
//          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//ARYBPROF DD DISP=SHR,DSN=ARY.PROFILES
//ARYBOFFL DD DISP=SHR,DSN=ARY.OFFOPTS
//ARYBPMAP DD DISP=SHR,DSN=ARY.PROFILE.MAPS
//ARYBPCAT DD DISP=SHR,DSN=ARY.PROFILE.CATS
//ARYSBACK DD DISP=SHR,DSN=ARY.SYSBACK
//ARYSBOBJ DD DISP=SHR,DSN=ARY.SYSBACK.OBJS
//ARYSBVOL DD DISP=SHR,DSN=ARY.SYSBACK.VOLS
//ARYSBSSD DD DISP=SHR,DSN=ARY.SYSBACK.SSIDS
//ARYBREPT DD DISP=SHR,DSN=ARY.BREPOR
//ISPFIL DD DISP=SHR,DSN=PAOLOR6.IC.JCL
//ISPLG DD SYSOUT=*,DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//ISPRK1 DD UNIT=SYSALLDA,SPACE=(CYL,(30,30)),
//          DCB=(RECFM=FB,LRECL=133,BLKSIZE=1330)
//ARYERROR DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//ARYDEBUG DD SYSOUT=*
//ARY#DATA DD *
    DISASTER_RECOVERY (
        DB2_SUBSYSTEM D9C1
        GEN_TO_DATASET PAOLOR6.IC.JCL
        PROFILE_NAME 'D9CGIC'
        PROFILE_CREATOR PAOLOR6
        EXECUTION_LIB_1 ARY.V2R1MO.SARYLOAD
        JOB_CARD_1_1 ' //PAOLOR62 JOB (XXX,POK),DRJOB2,CLASS=A,'
        JOB_CARD_1_2 'MSGCLASS=X'
        JOB_CARD_2_1 ' /*JOBPARM SYSAFF=SC63'
        JOB_CARD_3_1 ' /*'
        JOB_CARD_4_1 ' /*'
    )
/**

```

Figure 7-5 Disaster recovery job that builds DR JCL

Figure 7-6 shows the syntax of the input control cards.

```
//ARY#DATA DD *
DISASTER_RECOVERY      (
  DB2_SUBSYSTEM         D9C1
  USER_INDICATOR        ARY
  PROFILE_NAME           'D9CGIC'
  PROFILE_CREATOR        PAOLOR6
  EXECUTION_LIB_1        ARY.V2R1M0.SARYLOAD
  GEN_TO_DATASET         PAOLOR6.IC.JCL
  JOB_CARD_1_1           '//PAOLOR62 JOB (XXX,POK),DRJOB2,CLASS=A,'
  JOB_CARD_1_2           'MSGCLASS=X'
  JOB_CARD_2_1           '/*JOBPARM SYSAFF=SC63'
  JOB_CARD_3_1           '/*'
  JOB_CARD_4_1           '/*'
  BPROF_DSN              ARY.PROFILES
  BOFFL_DSN              ARY.OFFOPTS
  BPMAP_DSN              ARY.PROFILE.MAPS
  BPCAT_DSN              ARY.PROFILE.CATS
  SBACK_DSN              ARY.SYSBACK
  SBOBJ_DSN              ARY.SYSBACK.OBJS
  SBVOL_DSN              ARY.SYSBACK.VOLS
  SBSSD_DSN              ARY.SYSBACK.SSIDS
  BREPT_DSN              ARY.BREPORT
  POBJS_DSN              ARY.OBJECTS
  PARML_DSN              ARY.V2R1M0.SARYSAMP
  PARML_MEMBER           ARY#PARM
  DB2CNTL_DSN            ARY.DB2PARMS.CONTROL
)
/*
. . . . .
```

Figure 7-6 Syntax of the step that builds the DR JCL

Building the secondary profile

The process of building the secondary disaster recovery profile should be a regularly scheduled batch job.

Note: Disaster recovery profiles can only be built using the batch build process.

When the secondary profile is built, the disaster recovery batch job that is produced simply updates the archive log and backs up and rebuilds the BSDS. The build should be scheduled as frequently as we want to establish a point of consistence for recovery. This job should be scheduled frequently so that archive logs are to the recovery list as soon as they are created.

7.4 Recovery procedures

The procedures in this scenario are to recover using image copies and archive logs at a remote site in the case of disaster. We concentrate on the approach described in 7.2.3, “Continuous archive log transfer” on page 154.

7.4.1 Establish the environment

This section lists 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 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.

Note: These steps have most likely been completed as part of restoring the operating system and libraries. They are restated here only as a reminder.

7.4.2 Contents of the recovery PDS

Table 7-1 shows the contents of the PDS populated at the local site by System Backup and Restore Services. We discuss the use of each member in detail.

Table 7-1 Summary of recovery steps to rebuild the subsystem at the remote site

Member name	Description	Comments
D9C1#JC1	Recovery JCL to RESTORE the subsystem at the remote site.	<p>Our job contains the following steps:</p> <ul style="list-style-type: none"> ▶ Step D9C1DELC - an IDCAMS step to DELETE all the DB2 data sets. This includes the BSDS and LOG data sets. It uses member D9C1DELC. It deletes these data sets with the NOSCRATCH option. ▶ Step D9C1ALLC - an IDCAMS step to define the DB2 catalog and directory spaces, the boot strap data sets, the active log,s and all user-defined VCAT spaces. It uses member D9C1ALLC. ▶ Step D9C1CATL - an IDCAMS step that verifies that all IC data sets are cataloged. ▶ Step D9C1RBSR - invokes a Recovery Expert program, ARY@YRBS, to rebuild the BSDS into 4080 byte records. It uses member D9C1BSDS. ▶ Step D9C1CPBS - an IDCAMS step that performs a REPRO to insert the BSDS data created in the previous step, D9C1RBSR, into the two BSDS data sets. ▶ Step D9C1CRCR - invokes DSNJU003 to create a CRCR in the BSDS. It uses member D9C1CRCR. This CRCR is the time of the system level backup taken before at the local site. ▶ Step D9C1PRNT - invokes DSNJU004 to print the contents of the BSDS for member D9C1. ▶ Step D9C2RBSR - invokes a Recovery Expert program, ARY@YRBS, to rebuild the BSDS at the DR site from the 4,080-byte records. It uses member D9C2BSDS. This is for DB2 member D9C2. ▶ Step D9C2CPBS - an IDCAMS step that performs a REPRO to insert the BSDS data created in the previous step, D9C2RBSR, into the two BSDS data sets for the second member, D9C2. ▶ Step D9C2CRCR - invokes DSNJU003 to create a CRCR in the BSDS. It uses member D9C1CRCR for the second data sharing group member, D9C2. ▶ Step D9C2PRNT - invokes DSNJU004 to print the contents of the BSDS for member D9C2. ▶ Step D9C2YCPL - invokes the Recovery Expert program, ARy@YCPL, to copy the archives logs from tape to disk. It used member D9C1CPL
D9C1#JC2	Restarts all the DB2 spaces in RW mode.	This job is optional. This Disaster Recovery job can be run after the DB2 Subsystem has been started and the DB2 catalog and directory have been successfully restored. Although optional, we recommend running this job. It uses member D9C1STRF.
D9C1#JC3	This job restores the DB2 data sets in the correct order.	This job uses DSNUTILB and invokes the IBM DB2 stand-alone recovery utility, RECOVER, to restore the DB2 data sets in the correct order. It first restores SYSUTILX from the logs, then DBD01, then SYSCOPY, and so on.

Member name	Description	Comments
D9C1ALLC	Defines the VSAM clusters for the DB2 catalog and directory, BSDS, active logs, and user-defined VCAT spaces.	This member is used by job D9C1#JC1.
D9C1BSDS	Contains a copy of the bootstrap data set in 80-byte records.	This member is used by job D9C1#JC1.
D9C1CATL	Contains control cards for cataloging the image copies at the recovery site.	This member is used by job D9C1#JC1.
D9C1CPYL	Contains control cards for copying the archive logs from the tape to the recovery site.	This member is used by job D9C1#JC1. It is a list of all the archive logs that will be copied to disk.
D9C1CRCR	Contains the control card for creating the conditional restart.	This member is called twice in job D9C1#JC1. It is called once for each DB2 member.
D9C1DELA	Contains control cards to delete the archive logs (with the NOSCRATCH option) from the recovery site z/OS catalog. Those logs are on tape and will be copied to disk, then cataloged.	This member is used by job D9C1#JC1.
D9C1DELC	Contains control cards to uncatalog DB2 spaces at the recovery site, before restoring them from tape.	This member is used by job D9C1#JC1.
D9C1STRF	Contains control cards for starting application table spaces in the DB2 subsystem with RW access.	This member is used by job D9C1#JC2.
D9C2BSDS	Contains a copy of the bootstrap data set in 80-byte records.	This member is used by job D9C1#JC2. It is for the second member in our data sharing group, D9C2.

7.4.3 Run the job in member ssid#JCL1

The job contained in member ssid#JC1 contains several steps. Those steps prepare the DB2 subsystem or data sharing group to be restarted so that data recovery can be performed.

1. Issues IDCAMS DELETE to delete all index space and table space data sets from the MVS catalog. A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-7.

```

/** ** ** ** **
/**
/** Disaster Recovery Delete Noscratch DB2 Datasets
/**
/** Return Codes: 0 - Successful
/**
/** ** ** **
/**
//D9C1DELCL EXEC PGM=IDCAMS,REGION=006M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD DISP=SHR,DSN=PA0L0R6.IC.JCL(D9C1DELCL)

DELETE ('DB9CL.D9C1.BSDS01')
DELETE ('DB9CL.D9C1.BSDS02')
DELETE ('DB9CL.D9C1.LOGCOPY1.DS01')
DELETE ('DB9CL.D9C1.LOGCOPY1.DS02')
DELETE ('DB9CL.D9C1.LOGCOPY1.DS03')
DELETE ('DB9CL.D9C1.LOGCOPY2.DS01')
DELETE ('DB9CL.D9C1.LOGCOPY2.DS02')
DELETE ('DB9CL.D9C1.LOGCOPY2.DS03')
DELETE ('DB9CL.D9C2.BSDS01')
DELETE ('DB9CL.D9C2.BSDS02')
DELETE ('DB9CL.D9C2.LOGCOPY1.DS01')
DELETE ('DB9CL.D9C2.LOGCOPY1.DS02')
DELETE ('DB9CL.D9C2.LOGCOPY1.DS03')
DELETE ('DB9CL.D9C2.LOGCOPY2.DS01')
DELETE ('DB9CL.D9C2.LOGCOPY2.DS02')
DELETE ('DB9CL.D9C2.LOGCOPY2.DS03')
DELETE ('DB9CD.DSNDBC.DSNDB01.DBD01.I0001.A001')
DELETE ('DB9CD.DSNDBC.DSNDB01.DSNLLX01.I0001.A001')

...

DELETE ('DB9CD.DSNDBC.DLCDB.DRRICOPY.I0001.A001')
DELETE ('DB9CD.DSNDBC.DLCDB.ARCHIVES.I0001.A001')
DELETE ('DB9CD.DSNDBC.DSNDB06.TABS1EBP.I0001.A001')
DELETE ('DB9CD.DSNDBC.ADBDCH.ADBCKPTX.I0001.A001')
DELETE ('DB9CD.DSNDBC.ADBDCH.ADBCHKX1.I0001.A001')
DELETE ('DB9CD.DSNDBC.ADBDCH.ADBHLDX1.I0001.A001')
SET MAXCC = 0

```

Figure 7-7 The first step of DR job, ssid#JC1, and example of ssidDELC input

2. Recreate the DB2 catalog VSAM files. All VSAM and non-VSAM catalog files, log files, BSDS, and user VCAT-defined objects are created with the proper allocations. A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-8.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
/** Disaster Recovery Allocate DB2 Datasets
/** 1. DB2 Catalog and Directory Spaces
/** 2. Boot Strap Datasets
/** 3. Active Logs
/** 4. User Defined VCAT Spaces
/**
/** Return Codes: 0 - Successful
/**
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
//D9C1ALLC EXEC PGM=IDCAMS,REGION=006M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD DISP=SHR,DSN=PAOLOR6.IC.JCL(D9C1ALLC)

DEFINE CLUSTER
( NAME ('DB9CL.D9C1.BSDS01')
REUSE
RECORDSIZE(4089 4089)
FREESPACE(0 20)
KEYS(4 0)
CONTROLINTERVALSIZE(04096)
STORAGECLASS(DB9CLOG1)
MANAGEMENTCLASS(MCDB22)
VOLUMES(SBOX5E)
TRACKS(00000078,00000002)
SHAREOPTIONS(2 3) )
DATA
( NAME ('DB9CL.D9C1.BSDS01.DATA') )
INDEX
( NAME ('DB9CL.D9C1.BSDS01.INDEX') )
...

```

Figure 7-8 The second step of DR job, ssid#JC1, to allocate the DB2 data sets

3. Catalog all the image copies from the last n number of days (as specified in the DR profile). A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-9.

Note: For data sharing environments, the next substeps are performed for each group member.

```

/** * * * * *
/**
/** Disaster Recovery Catalog Image Copies
/** 1. Make sure all image copies needed at DR are cataloged
/**
/** Return Codes: 0 - Successful
/**
/** * * * * *
/**
//D9C1CATL EXEC PGM=IDCAMS,REGION=006M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD DISP=SHR,DSN=PAOLOR6.IC.JCL(D9C1CATL)
/**

```

Figure 7-9 The third step of the ssid#JC1 DR job - recover

4. Rebuild the BSDS from the 80-byte record file, placing it back into 4,089-byte records. A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-10.

```

/** * * * * *
/**
/** Disaster Recovery Rebuild Boot Strap Dataset into 4080 byte recs
/**
/** Return Codes: 0 - Successful
/**
/** * * * * *
/**
//D9C1RBSR EXEC PGM=ARY@YRBS,REGION=006M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//BSDS#IN DD DISP=SHR,DSN=PAOLOR6.IC.JCL(D9C1BSDS)
//BSDS#OUT DD DSN=&BSDS,DISP=(NEW,PASS,DELETE),
// UNIT=3390,SPACE=(TRK,(10,10),RLSE),
// DCB=(RECFM=V,LRECL=4093)

```

Figure 7-10 Fourth step of ssid#JC1 job - rebuild the BSDS

5. Restores the BSDS by placing the 4,089-byte records into a VSAM file. A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-11.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
/** Disaster Recovery Copy Boot Strap Dataset into VSAM datasets
/**
/** Return Codes: 0 - Successful
/**
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
//D9C1CPBS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//BSDSI DD DSN=&BSDS,DISP=(OLD,DELETE,DELETE)
//BSDS1 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS01
//BSDS2 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS02
//SYSIN DD *
        REPRO INFILE(BSDSI) OUTFILE(BSDS1) REUSE
        REPRO INFILE(BSDSI) OUTFILE(BSDS2) REUSE

```

Figure 7-11 The fifth step uses REPRO to update the BSDS with the information from the previous step

6. Create a conditional restart. A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-12.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
/** Disaster Recovery Create Conditional Restart Control Record
/**
/** Return Codes: 0 - Successful
/**
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
//D9C1CRCR EXEC PGM=DSNJU003,REGION=006M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
// DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS01
//SYSUT2 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS02
//SYSIN DD DISP=SHR,DSN=PAOL0R6.IC.JCL(D9C1CRCR)
..
CRESTART CREATE,ENDLRN=C20EE3B28480,FORWARD=YES,BACKOUT=YES

```

Figure 7-12 The sixth step creates the CRCR record for DB2 to restart properly for recovery

7. Print the contents of the BSDS. A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-13.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
/** Disaster Recovery Print the Boot Strap Dataset 1
/**
/** Return Codes: 0 - Successful
/**
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
//D9C1PRNT EXEC PGM=DSNJU004,REGION=006M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
// DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=DB9CL.D9C1.BSDS01

```

Figure 7-13 The seventh step prints the BSDS information

8. Uncatalog the tape archive logs.
9. Copy the uncataloged tape archive logs to disk and catalog them. This step speeds the recoveries at the DR site because the tape logs will be copied to disk, and all recovers that apply the log will use the disk copy instead of the tape copy.

A sample of the JCL generated by System Backup and Restore Services is shown in Figure 7-14.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
/** Disaster Recovery Copy Archive Logs from Tape to DASD
/**
/** Return Codes: 0 - Successful
/**
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
//D9C1YCPL EXEC PGM=ARY@YCPL,REGION=006M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//ARY@CNTL DD DISP=SHR,DSN=PAOLOR6.IC.JCL(D9C1CPYL)

```

Figure 7-14 The last step of job ssid#JC1 copies the archives logs from tape to DASD

7.4.4 Restart the DB2 subsystems

The following steps must be completed to restart the DB2 subsystems prior to recovering the objects:

1. Reassemble the DSNZPARMs.

Before reassembling the remote site's DSNZPARMs settings:

- a. Change RESTART to DEFER.

- b. Set the site as local or recovery.
- c. Change the INSTALLSYSADM and INSTALLSYSOPR user IDs to the user ID that will be recovering the DB2 catalog.

Note: You can create a DR DSNZPARMS member at the local site with the necessary settings for the recovery site, and maintain it in the local site SDSNEXIT library for the subsystem. When the MVS catalog is restored at the recovery site, this ZPARMS member will already be on site and contain the proper settings for disaster recovery.

2. Start DB2 using the reassembled DSNZPARMS.

Note: When restoring DB2 9 subsystems using the image copy method, DB2 must be started with ACCESS(MAINT).

For all other restorations, we recommend that DB2 be started with ACCESS(MAINT).

3. Reply to conditional restart messages for each subsystem with Y.
4. Run the ssid#JC2 job to force application table spaces to start in RW mode. This job starts all application table spaces in the DB2 subsystem with read-write (RW) access. See Figure 7-15.

Note: While optional, we recommend this job. If any table space is not in RW mode, the recovery will fail.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
/** Step:
/**
/** Desc: This job is an optional Disaster Recovery job. This
/** job is to be run after the DB2 Subsystem has been
/** started and the DB2 Catalog and Directory have been
/** recovered.
/**
/** This step will start all spaces in the entire DB2
/** subsystem in RW (Read Write) mode. If any spaces
/** have an invalid status pending, the recovery of that
/** object will fail and your recovery job will be
/** terminated. Starting all spaces in RW mode will
/** ensure that the your application recovery job will
/** not fail and require a utility restart.
/**
/**
/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**
// EXEC PGM=IKJEFT1A,REGION=006M
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
// DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DISP=SHR,DSN=PAOLOR6.IC.JCL(D9C1STRF)

```

Figure 7-15 The second DR job, xxxx\$JC1, is optional

7.4.5 Run the job in member ssid#JC3

For the image copy method, we recover the DB2 objects from image copies via many steps. The first steps restore the DB2 system catalog, followed by DB2 application objects.

You may have your own application recovery jobs created that already take issues like tape stacking in the backups into account. In this case, you would only want to run the first few steps of this job, which recover the DB2 catalog and directory.

A partial sample of the JCL from member ssid#JC3 is shown in Figure 7-16.

```

/** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/** Step:      RCVR001                                         *
/**                                                    *
/** Desc:       This step will invoke the IBM Recover Utility   *
/**              for DSNDB01.SYSUTILX.                          *
/** ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**                                                    *
/**RCVR001 EXEC  PGM=DSNUTILB,REGION=006M,COND=(4,LT),        *
/**              PARM=(D9C1,)                                  *
/**                                                    *
/**STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT                     *
/**          DD DISP=SHR,DSN=DB9C9.SDSNLOAD                   *
/**SYSPRINT DD SYSOUT=*                                       *
/**SYSOUT  DD SYSOUT=*                                       *
/**UTPRINT DD SYSOUT=*                                       *
/**SYSIN   DD *                                              *
RECOVER
TABLESPACE DSNDB01.SYSUTILX
RECOVERYSITE
/**                                                    *
/** Desc:       This step will invoke the IBM Recover Utility   *
/**              for indexes on DSNDB01.SYSUTILX.               *
/** ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/**                                                    *
/**RC02001 EXEC  PGM=DSNUTILB,REGION=006M,COND=(4,LT),        *
/**              PARM=(D9C1,)                                  *
/**                                                    *
/**STEPLIB DD DISP=SHR,DSN=DB9C9.SDSNEXIT                     *
/**          DD DISP=SHR,DSN=DB9C9.SDSNLOAD                   *
/**SYSPRINT DD SYSOUT=*                                       *
/**SYSOUT  DD SYSOUT=*                                       *
/**UTPRINT DD SYSOUT=*                                       *
/**SYSIN   DD *                                              *
REBUILD INDEX
ALL TABLESPACE DSNDB01.SYSUTILX
SORTDEVT VIO
...

```

Figure 7-16 DR Job xxx#JC3 recovers DB2 data sets in the proper order

7.5 Considerations

The solution we discussed here just addressed a DB2 subsystem. If you have CICS or IMS installed as transaction a manager, 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.

System level backup based remote recovery

BACKUP SYSTEM and RESTORE SYSTEM utilities can use the dump options to propagate the volume copies from disk to tape and be used for local backup or transported to a remote site.

System Backup and Restore Services's subsystem setup facility of Recovery Expert will help you make your DB2 subsystem available for the BACKUP SYSTEM and RESTORE SYSTEM utilities.

In this chapter we describe how to use Recovery Expert to offload system level backups to tape and how to restore from this tape backup on a system level.

Recovery Expert allows you to offload your system level backups to tape. You have the option to offload using DFSMSHsm or the System Backup and Restore Services to perform an offload. You have the choice of different options for the offloads, such as the dump class, backup destination (local or backup), data set naming conventions, and so on, to meet your needs.

This chapter contains the following sections:

- ▶ Offloading to tape at the local site
- ▶ Building Disaster Recovery JCL at the local site
- ▶ Restoring steps at the disaster recovery site

8.1 Offloading to tape at the local site

Recovery Expert gives you the functionality to offload your system level backups on DASD to tape. See Figure 8-1. You have two offload options, using DFSMSHsm or the System Backup and Restore Services of Recovery Expert. You have different choices for the offloads, such as the dump class backup destination (local or backup), data set naming conventions, and other options.

Note: Load libraries, SMP/E files, and other data sets that may be used by a DB2 subsystem are considered to be already at the remote site.

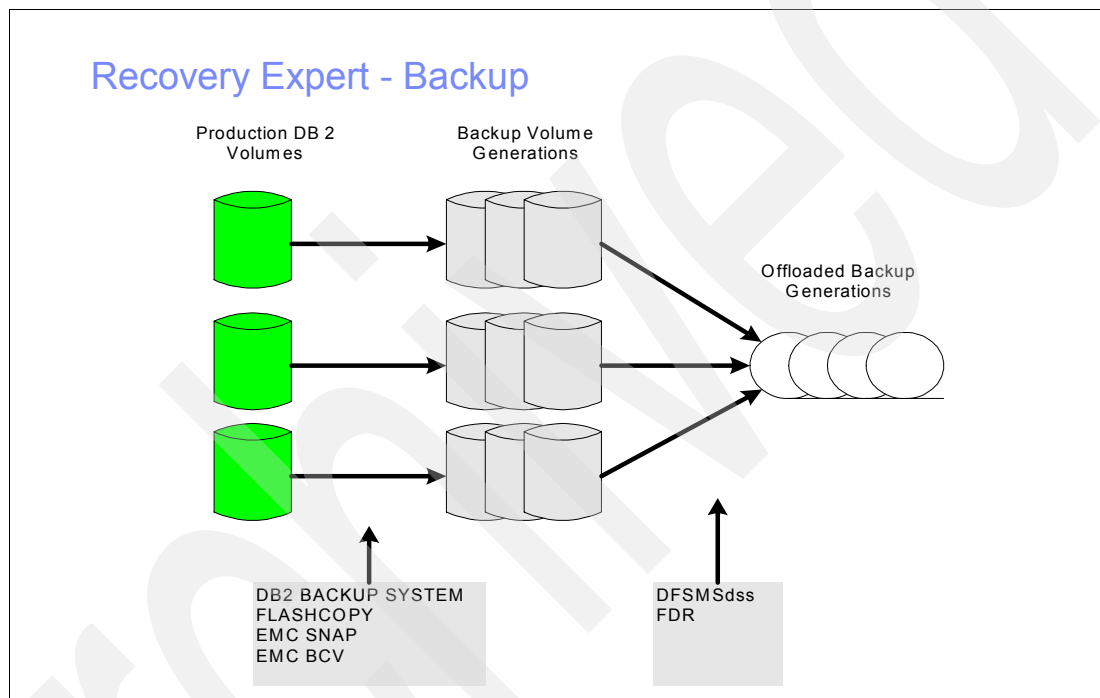


Figure 8-1 Tape dump

In the next scenario we provide an example of how to select a specific system level backup for offloading to tape. You will select the backup based upon information stored in the Recovery Expert System Backup Recovery Services repository.

There are two ways to offload your system level backup to tape:

- During your system level backup execution

This is an option that you select and that is part of your BACKUP PROFILE.

- As a stand-alone task

This option generates a task that allows you to perform an offload to tape at any time. It is dependant on a previously successful execution of a BACKUP SYSTEM.

8.1.1 Offloading to tape during a system level backup

You can select to offload your system level backup to tape as part of or during your backup. When you create your BACKUP PROFILE, choose whether you want to use this option.

In the example in Figure 8-2 a BACKUP PROFILE that was previously created and used will be updated to enable offloading from DASD to tape.

```

ARY$BPRD      ----- Backup Profile Display ----- 2008/02/22 19:40:05
Option ==>                                         Scroll ==> CSR
-----
Profile   Like *
Creator   Like *                                Row 1 of 1  >
-----

Cmd  Name                      Creator  SSID  Updt
U    D9C1                      PAOLOR6  D9C1  U
***** Bottom of Data *****

Valid Line Cmds: (B-Build, C-Creat, D-Delete, R-Rename, U-Update, V-View)

```

Figure 8-2 Selecting a backup profile for update

In Figure 8-2 the backup profile, D9C1, is selected for update. Backup profiles are reusable and editable, and are created on a per-subsystem basis. After profile setup has been run, you cannot change the SSID or method of backup in a profile, but you can change other settings, as well as redefine the source and target volumes for the offload options. See Figure 8-3.

```

ARY$BPRU      ----- Update Backup Profile ----- 2008/02/22 19:43:18
Option ==>                                         Scroll ==> CSR
-----
Creator: PAOLOR6      Name: D9C1                      User: PAOLOR6
Share Option: U (Upd,View,No)  Description: D9C1 SYSTEM LEVEL BACKUP
-----
----- Backup Options -----
DB2 Subsystem ==> D9C1                      Current Generation==> 02
Backup Method  ==> D (Bcv/Snap/Flash/Db2) Setup Needed ==> N
Backup Scope   ==> F (Full/Data)             Issue Log Suspend ==> N (Yes/No)
Backup Generations==> 02 (01 - 99)           Validate DB2 Vols ==> Y (Yes/No)
Offload Options ==> Y (Yes/No/Update)         Enable Obj Restore==> N (Yes/No)
----- Volume Mappings ----- Row 1 of 44  +
Source  Dev  Src  Target
Cmd  Volumes Type Unit Volumes Message Area
SB0X5A 3390-9 D800 SB0X6I
SB0X7I
SB0X5B 3390-9 D900 SB0X6J
SB0X7J
SB0X5C 3390-9 DA00 SB0X6K
SB0X7K
SB0X5D 3390-9 DB00 SB0X6L

Valid Line Cmds: (I-Insert, D-Delete)

```

Figure 8-3 Selecting the offload options to update the backup profile

When you initially select this option, the Offload Options window appears in order for you to specify offload options. See Figure 8-4. If you edit the profile at a later time and wish to update the offload options, enter U in this field.

```

ARY$OFF1 ----- Offload Options ----- 2008/02/22 19:44:33
Option ==>
-----
Creator: PAOLOR6      Name: D9C1                      User: PAOLOR6
Share Option: U (Upd,View,No)  Description: D9C1 SYSTEM LEVEL BACKUP
-----
Enter the Offload options to associate with this Backup profile:

Local Primary           ==> Y (Yes/No/Update)
Local Backup            ==> N (Yes/No/Update)
Recovery Site Primary   ==> Y (Yes/No/Update)
Recovery Site Backup    ==> N (Yes/No/Update)

Offload Generations     ==> 01 (1 - 99)
Delete Aged Backup files ==> Y (Yes/No)
Compress Data           ==> N (Yes/No)
Data Mover              ==> D (Dfsmsdss, Fdr, or fdrInstant)
Number of Tasks         ==> 02 (1 - 99)

Enter END command to return to the previous screen

```

Figure 8-4 Offload copy selection

The Offload Options window, shown in Figure 8-5 on page 177, allows you to define offload specifications to meet your site's needs, such as the backup destination, data set naming conventions, and which data mover you want to use for the offload.

When you build a backup profile and set the Perform Offload field to Y, ARY offloads the backup per these specifications when the job is run. Specify how many generations of backups you want to keep. ARY *rolls off* the oldest copy if required.

Note: All backup types (LP, LB, RP, and RB) are done simultaneously. If you specify four backups to offload and specify one task, four tape drives will be required. The example above will cause the backup to ask for four drives— two backup types (LP and RP) and two tasks.


```

ARY$OFF1 ----- Offload Options ----- 2008/03/09 22:46:09
Option ==>
-----
Creator: PAOLOR6      Name: D9C1                      User: PAOLOR6
Share Option: U      (Upd,View,No)      Description: D9C1 SYSTEM LEVEL BACKUP
-----
Enter the Offload options to associate with this Backup profile:

Local Primary          ==> U      (Yes/No/Update)
Local Backup           ==> N      (Yes/No/Update)
Recovery Site Primary  ==> U      (Yes/No/Update)
Recovery Site Backup   ==> N      (Yes/No/Update)

Offload Generations    ==> 02    (1 - 99)
Delete Aged Backup files ==> Y    (Yes/No)
Compress Data          ==> Y      (Yes/No)
Data Mover             ==> D      (Dfmsdss, Fdr, or fdrInstant)
Number of Tasks        ==> 02    (1 - 99)

Enter END command to return to the previous screen

```

Figure 8-5 Offload options specification for the LP and RP copies

You have the choice to update the output options for each copy type. Enter U to update the options, as shown in Figure 8-6.

```

ARY$OFF2 ----- LP Offload Options ----- 2008/03/09 23:00:19
Option ==>
-----
Creator: PAOLOR6      Name: D9C1                      User: PAOLOR6
Share Option: U      (Upd,View,No)      Description: D9C1 SYSTEM LEVEL BACKUP
-----
Update DSN Specification => Y      (Yes/No)
Unit Type                => CART    (CART/DISK/etc.)
Catalog                 => Y        (Yes/No)
Data Class               =>          (8 character class)
Storage Class            =>          (8 character class)
Management Class         =>          (8 character class)
Tape specific parameters (only needed if Unit Type is a Tape device):
Stack Backups on Tape    => Y        (Yes/No)
Tape Stack Limit         => 005      (1 - 999)
Expiration date *or*     =>          (YYYYDDD/YYDDD)
Retention period         => 0060     (4 digit number)
Maximum Tapes            => 005      (1-256)

Enter END command to return to the previous screen

```

Figure 8-6 Offload options specifications for the local copy

This Offload Options window allows you to define parameters for your backup files, such as SMS data set information, storage location, and tape options.

Each volume in the system backup is dumped (or copied to tape) in one file. The most important fields are:

- ▶ Unit Type - This is checked to be a valid unit at your site. It might be possible to enter a disk unit that is set up as a remote disk at the disaster recovery site.
- ▶ Catalog - Controls whether the dump files are cataloged at the local site.
- ▶ Stack - Controls whether Recovery Expert will stack the volume dumps onto tape.
If yes, the tape stack limit indicates how many volume dumps will be written before a new tape is requested. Recovery Expert SBRS will dynamically allocate all tapes and keep the tape mounted across all tape-stacking activities.
- ▶ Expiration date - The date at which the tape management system will place the tape volser back in the scratch pool.
- ▶ Retention period - The number of days before the TMS will place the tape back in the scratch pool.

U (update) is entered in Figure 8-7 to build the output data set names for the local copy.

```

ARY$OFF3      ----- LP Offload DSN Specification ----- 2008/03/09 23:04:09
Option  ==>

-----
Creator: PAOLOR6      Name: D9C1                      User: PAOLOR6
Share Option: U  (Upd,View,No)      Description: D9C1 SYSTEM LEVEL BACKUP
-----

Enter codes to create a dataset name specification:
Qualifier code ==>      Free form literal ==>      Show DSN ==> N

Current dataset name qualifier string:
&UID...&SSID...&DAY...&OFFLOAD1.&VOLSER

Valid dataset name specification codes are:
1. Volser          9. Day (DD)          17. Backup Type (#18.#19)
2. Vcatname        10. Julian Day (DDD)    18. Local/Recovery (L/R)
3. Subsystem ID    11. Hours (HH)          19. Primary/Backup (P/B)
4. User ID         12. Minutes (MM)        20. Job Name
5. Time (HHMMSS)   13. Seconds (SS)         21. Step Name
6. Date (YYYYDDD)  14. Timestamp            22. Profile Creator
7. Year (YYYY)     15. Random Number        23. Profile Name
8. Month (MM)      16. GDG (+1)..(+n)        24. Substring Qualifier
                                   25. Use freeform literal

Enter END command to return to the previous screen

```

Figure 8-7 Updating the output naming convention for the offload

The Offload DSN Specification window allows you to build data set names using variables that are resolved at runtime.

You need to review and update all the data set naming conventions for each copy type (LP, LB, RP, RB) that you selected for offloading.

Note that you should include the VOLSER parameter somewhere in the data set name. We also recommend including a time stamp or some sort of time/date variable. This will make the data set names used to hold the volume dumps unique across each offload and multiple offloads.

Once you have updated the backup profile you can build the system level backup job. The system level backup JOB will now have an additional step that offloads the backup to tape. as shown in Figure 8-8.

```

...
//ARY#REPT DD SYSOUT=*
//ARYSNAPO DD SYSOUT=*
//ARY#PARM DD DSN=ARY.V2R1M0.SARYSAMP(ARY#PARM),DISP=SHR
//ARYIN DD *
    BACKUP PA0LOR6.D9C1
/*
/*
/** * * * * *
/*
/* Step: ARYJOFFL
/*
/* Desc: This step will invoke the Offload job.
/*
/** * * * * *
/*
//ARYJOFFL EXEC PGM=ARY@MAIN,REGION=006M,COND=(4,LT)
/*
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
// DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//ARYBPROF DD DISP=SHR,DSN=ARY.PROFILES
//ARYBOFFL DD DISP=SHR,DSN=ARY.OFFOPTS
//ARYBPMAP DD DISP=SHR,DSN=ARY.PROFILE.MAPS
//ARYBPCAT DD DISP=SHR,DSN=ARY.PROFILE.CATS
//ARYSBACK DD DISP=SHR,DSN=ARY.SYSBACK
//ARYSBOBJ DD DISP=SHR,DSN=ARY.SYSBACK.OBJS
//ARYSBVOL DD DISP=SHR,DSN=ARY.SYSBACK.VOLS
//ARYSBSSD DD DISP=SHR,DSN=ARY.SYSBACK.SSIDS
//ARYBREPT DD DISP=SHR,DSN=ARY.BREPORT
//ARY#REPT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//ARYOUT DD SYSOUT=*
//ARYSNAPO DD SYSOUT=*
//ARY#PARM DD DSN=ARY.V2R1M0.SARYSAMP(ARY#PARM),DISP=SHR
//ARYIN DD *
    OFFLOAD PA0LOR6.D9C1
    GENERATION LAST-BACKUP

```

Figure 8-8 Additional OFFLOAD step as part of system level backup

8.1.2 Offloading to tape as a separate task

Recovery Expert gives you an additional option to offload DASD-based system level backups to tape. A batch job can be built online by selecting a system level backup from the System Restore and Offload window and run the job when required.

The example in Figure 8-9 shows how to select a DASD-based system level backup for offloading to tape.

```
ARY$MAIN      ----- Recovery Expert for DB2 -----  
Option  ==> R  
  
User: PAOLOR6 - ARY  
  
-----  
  
B. System Backup Profiles  
  
R. System Restore and Offload  
  
O. Object Recovery Profiles  
  
D. Disaster Recovery Profiles  
  
M. DB2 Subsystem Analysis and Setup  
  
S. Product Setup  
  
X. Exit  
  
Enter END command to return to ISPF.
```

Figure 8-9 Selection options for an offload - R

In Figure 8-10, a list of valid recovery points is displayed. The Restore System Display shows a list of valid system restore points from which you can choose the event that you need to restore a subsystem. V (View Reports) is entered to show the details of the system level backup.

ARY\$REST

Option ==>

----- Restore System Display -----

Scroll ==> CSR

2008/03/09 23:22:06

DB2 Subsystem ID *

Select a row from the list of recovery points displayed below.

To enter an RBA/LRSN and have a recovery point automatically selected for you, enter the RESTORE primary command followed by a DB2 subsystem ID.

Row 1 of 6 >

Cmd	Date	Time	Data Only	Mixed Data	On Disk	On Offload	Obj Rcvr	Type	Nbr Vols
V	03/06/2008	17:17:40	No	No	Yes	Yes	No	DB2	0022
	SSID: D9C1 RBA/LRSN: C20DC63434CC								
	03/04/2008	18:34:35	No	No	No	Yes	No	DB2	0022
	SSID: D9C1 RBA/LRSN: C20B53AAF74B								
	03/04/2008	15:49:16	No	No	Yes	Yes	No	DB2	0022
	SSID: D9C1 RBA/LRSN: C20B2EB5DC9F								

***** Bottom of Data *****

Valid Line Commands: (S - Select, D - Delete, V - View Report, O - Offload)

Figure 8-10 Selecting a valid recovery point for an offload - view details

The view report (Figure 8-11) gives you detailed information about the system level backup, which completed successfully.

```

ARY$BREP          ----- Backup Report Display ----- 2008/03/09 23:26:10
Option ===>                               Scroll ==> CSR

-----
                                Row 1 of 112  +>
                                Recovery Expert for DB2 z/OS
                                Backup Summary Report

Utility Executed:..... Backup
Profile Name:..... PAOLOR6.D9C1
DB2 Subsystem:..... D9C1
DB2 Version:..... 0910
Backup Type:..... IBM System Level Backup
Backup Contains:..... Object Data and Log Data
Nbr of Volumes:..... 0022
HSM Backup Token:..... C4F9C3F1C20DC63434CCF686C20DC486FBCE
Backup LRSN:..... C20DC63434CC
Last Checkpoint LRSN:..... C20DC6305D36
Backup Date:..... 03/06/2008
Backup Time:..... 17:17:40
Consistency Method:..... IBM System Level Backup
Supports Object Restore:.. No
I/O Suspend Time:..... 2008-03-06-17.17.29.040237
I/O Resume Time:..... 2008-03-06-17.17.40.684636

                                Backup Volume Detail Report

<-Source Volumes->  <-Targets->  <-----Data Types----->
Volser Ucb# Devtyp  Volser Ucb#  Obj OCat ALog ACat RLog RCat
-----
SBOX5A D800 3390-9  SBOX7I D807  No  No   Yes  No   No   Yes
SBOX5B D900 3390-9  SBOX7J D907  No  No   No   No   No   No
SBOX5C DA00 3390-9  SBOX7K DA07  No  No   Yes  No   No   No
...
SBOX5U DC02 3390-9  SBOX7C DA06  Yes Yes  No   No   No   No
SBOX5V DD02 3390-9  SBOX7D DB06  Yes No   No   No   No   No
                                Recovery Expert for DB2 z/OS
                                Volume Offload Summary Report

Utility Executed:..... Offload
Profile Name:..... PAOLOR6.D9C1
Offload Date:..... 03/06/2008
Offload Time:..... 17:17:42
Data Mover:..... DFSMSdss
Compress:..... Yes
Generation:..... 0002
Nbr Of Volumes:..... 0022

-----
SBOX5A D800 LP  PAOLOR6.D9C1.D06.OFFL0D1.SBOX7I 001  CART0B
RP  PAOLOR6.D9C1.D066.OFFL0D1.P.SBOX7I 001  CART0D
SBOX5B D900 LP  PAOLOR6.D9C1.D06.OFFL0D1.SBOX7J 001  CART0A
RP  PAOLOR6.D9C1.D066.OFFL0D1.P.SBOX7J 001  CART0C
...

```

Figure 8-11 Example of the view report output

This same information was produced when the system level backup job ran and is part of the job output. See Figure 8-12.

Edit Generated Job	Y	(Yes/No)
Build job in Dataset	PAOLOR6.SLB.JCL	
Member	OFFLOAD	Offload Job Member
Job Cards:		
==> //PAOLOR60 JOB (XXX,P0K),PAOLOR6,CLASS=A,MSGCLASS=H		
==> /*JOBPARM SYSAFF=SC63		
==> /*		
==> /*		
Press ENTER to process or PF3 to Cancel		

Figure 8-12 Window to specify JOBCARD information and output data set name for offload JCL

The Build Offload Job window, shown in Figure 8-13, lets you specify locations for the output of the offload job.

```
//ARYJOFFL EXEC PGM=ARY@MAIN,REGION=006M,COND=(4,LT)
/*
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
// DD DISP=SHR,DSN=DB9C9.SDSNEXIT
// DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//ARYBPROF DD DISP=SHR,DSN=ARY.PROFILES
//ARYBOFFL DD DISP=SHR,DSN=ARY.OFFOPTS
//ARYBPMAP DD DISP=SHR,DSN=ARY.PROFILE.MAPS
//ARYBPCAT DD DISP=SHR,DSN=ARY.PROFILE.CATS
//ARYSBACK DD DISP=SHR,DSN=ARY.SYSBACK
//ARYSBOBJ DD DISP=SHR,DSN=ARY.SYSBACK.OBJS
//ARYSBVOL DD DISP=SHR,DSN=ARY.SYSBACK.VOLS
//ARYSBSSD DD DISP=SHR,DSN=ARY.SYSBACK.SSIDS
//ARYBREPT DD DISP=SHR,DSN=ARY.BREPORT
//ARY#REPT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//ARYOUT DD SYSOUT=*
//ARYSNAPO DD SYSOUT=*
//ARY#PARM DD DSN=ARY.V2R1M0.SARYSAMP(ARY#PARM),DISP=SHR
//ARYIN DD *
OFFLOAD PAOLOR6.D9C1
GENERATION 01
DATE 02/26/2008
TIME 18:05:03
/*
```

Figure 8-13 Offload JCL example

Example 8-1 shows the offload report.

Example 8-1 Example of an offload to tape

Recovery Expert for DB2 z/OS

Volume Offload Summary Report

Utility Executed:..... Offload

Profile Name:..... PAOLOR6.D9C1
 Offload Date:..... 03/06/2008
 Offload Time:..... 17:17:42
 Data Mover:..... DFSMSdss
 Compress:..... Yes
 Generation:..... 0002
 Nbr Of Volumes:..... 0022

Recovery Expert for DB2 z/OS
 Volume Offload Detail Report

<---DB2--->				<---Tape--->	
Volser	Ucb#	Type	Offloaded to Filename	FileSeq	Volser
SBOX5A	D800	LP	PAOLOR6.D9C1.D06.OFFL0D1.SBOX7I	001	CART0B
		RP	PAOLOR6.D9C1.D066.OFFL0D1.P.SBOX7I	001	CART0D
SBOX5B	D900	LP	PAOLOR6.D9C1.D06.OFFL0D1.SBOX7J	001	CART0A
		RP	PAOLOR6.D9C1.D066.OFFL0D1.P.SBOX7J	001	CART0C
SBOX5C	DA00	LP	PAOLOR6.D9C1.D06.OFFL0D1.SBOX7K	002	CART0B

...

17:17:42 ARYS001I - Recovery Expert for DB2 z/OS Starting. Version 02.01.013
 17:17:42 ARYS162I - Parmlib used for this execution
 17:17:42 ARYS003I - Control Cards:
 17:17:42 ARYS004I - OFFLOAD PAOLOR6.D9C1
 17:17:42 ARYS004I - GENERATION LAST-BACKUP
 17:17:42 ARYS004I -
 17:17:42 ARYS123I - Backup PAOLOR6.D9C1 generation 002 was read from the repository.
 17:17:42 ARYS004I - Performing full volume offload.
 17:17:42 ARYS249I - Task 02 - Offload process starting for unit D907 (Source volser SBOX5B).
 17:17:42 ARYS249I - Task 01 - Offload process starting for unit D807 (Source volser SBOX5A).
 17:20:00 ARYS263I - Task 02 - Unit D907 offloaded to local primary dataset PAOLOR6.D9C1.D06.OFFL0D1.SBOX7J
 17:20:00 ARYS263I - Task 02 - Unit D907 offloaded to remote primary dataset
 PAOLOR6.D9C1.D066.OFFL0D1.P.SBOX7J
 17:20:00 ARYS254I - Task 02 - Offload process for unit D907 (Source volser SBOX5B) is complete.
 17:20:00 ARYS249I - Task 02 - Offload process starting for unit DB07 (Source volser SBOX5D).
 17:20:19 ARYS263I - Task 01 - Unit D807 offloaded to local primary dataset PAOLOR6.D9C1.D06.OFFL0D1.SBOX7I
 17:20:19 ARYS263I - Task 01 - Unit D807 offloaded to remote primary dataset
 PAOLOR6.D9C1.D066.OFFL0D1.P.SBOX7I
 17:20:19 ARYS254I - Task 01 - Offload process for unit D807 (Source volser SBOX5A) is complete.
 17:20:19 ARYS249I - Task 01 - Offload process starting for unit DA07 (Source volser SBOX5C).
 ...

8.2 Building Disaster Recovery JCL at the local site

In this section we describe the tasks to perform at your local site for preparing and building an off-site disaster recovery JCL using Recovery Expert.

Preparing your disaster recovery profiles at your local site is the first step to perform to eventually build JOBS to recover your subsystem at a remote site. Once the profiles have been created you can run the jobs on a continuous basis as part of your scheduler. Specify D for Disaster Recovery Profile in the menu shown in Figure 8-14.

Note: For a data sharing group, you only need to specify one profile for the group if you want to process all the members. System Backup and Restore Services will start a recovery subtask for each member in the data sharing group, including the copying of all the archive logs in one multi-tasking job for the entire data sharing group.

```

ARY$MAIN      ----- Recovery Expert for DB2 z/OS -----
Option  ==> D

User: PAOLOR6 - ARY

-----

  B. System Backup Profiles
  R. System Restore and Offload
  O. Object Recovery Profiles
  D. Disaster Recovery Profiles
  M. DB2 Subsystem Analysis and Setup
  S. Product Setup
  X. Exit

Enter END command to return to ISPF.

```

Figure 8-14 Recovery Expert main menu - using option D to create a disaster recovery profile

The accuracy of your disaster recovery depends on how you define your disaster recovery profiles. To ensure an accurate recovery, you may need to build several disaster recovery profiles and ensure that the jobs produced from the profile are executed on a regular basis. This can be achieved via your job scheduler.

Disaster recovery profiles can be renamed, viewed, and deleted in the same manner as backup object profiles. The updates to the disaster recovery profile are shown in Figure 8-15 on page 186.

We list the values specified for the fields shown in the window in the example in Figure 8-15 on page 186:

- ▶ We select the disaster recovery method for this disaster recovery profile by specifying S for system level backups. By contrast, in Chapter 7, “Traditional image copy recovery” on page 151, the image copy selection was used for the recovery method at the disaster recovery site.
- ▶ We specify C for archive logs used at DR. Archive logs will be copied and will be used at the disaster recovery site.
- ▶ For copy localsite logs, we choose B for both.

- We specify Y for force a checkpoint before archiving. This will cause ARY to issue a SET LOG LOGLOAD(0) command.

When Y is specified in this field, the recovery job only includes archive log processing. Specify this option to reduce the overhead involved in deleting and creating catalog entries for image copies when no changes have been made other than table updates. This option should only be specified for jobs after a comprehensive recovery job has been built.

- We also set Force the Active log to Archive to Y.
- The option between Y and N for “Only run Archive Log Update Process” has some performance implications.

If you specify N, Recovery Expert will build members ARYREPOC and ARYREPOD, which are used to define the VSAM Repository files, and it will create all the catalog entries for image copies.

Building members ARYREPOC and ARYREPOD is only needed once.

Once you have run with this option at least once in N, then you can change it to Y, as it will reduce the work involved in building the jobs.

```

ARY$DPRU      ---- Update Disaster Recovery Profile ---- 2008/02/26 13:12:08
Option ==>

-----
Creator: PAOLOR6      Name: D9CG                                User: PAOLOR6
Share Option: U (Upd,View,No)      Description:
DB2 Subsystem: D9C1
-----

Disaster Recovery Method      ==> S (Image copy/System backup)
Archive Logs used at DR      ==> C (Copied/1/2)
Copy Localsite Logs          ==> B (1/2/Both/Create 2 copies from 1)
Force a checkpoint before Archiving ==> Y (Yes/No)
Force the Active log to Archive ==> Y (Yes/No)
Only run Archive Log Update Process ==> N (Yes/No)
Process Datasharing Subsystems ==> A (All,Ssid,Lpar)
Archive Logs needed at DR      ==> 014 (days) and/or 000 (hours)
Copy Archive Logs to DASD      ==> 007 (days) and/or 000 (hours)
Unit for copying Archive Logs ==> CART
DR Archive Log Prefix 1      ==> D9C1.ARCHLOG1.DR
DR Archive Log Prefix 2      ==> D9C1.ARCHLOG2.DR
Image copy Options
Image Copies (or SLB) used at DR ==> R (Localsite/Recovery site)
Catalog x days of Image Copies at DR ==> 007 (0-365)

Enter END command to Update Disaster Recover Profile

```

Figure 8-15 Disaster Recovery Profile detailed definition window

Note: We recommend that you create two disaster recovery profiles. One should be a primary profile that includes forcing both a checkpoint and an active log to archive, and that has the Only run Archive Log Update Process field set to N.

The other profile to be created should have the Only run Archive Log Update Process field set to Y. This will not force a checkpoint or force the active log to archive. This job should be run periodically throughout the day to keep the recovery information up to date as much as possible and be added to your scheduler.

In Figure 8-16, the batch option of building the actual disaster recovery JCL was chosen.

```
----- Build Job for PAOLOR6.D9C1 -----

Build Online or Batch  B           (Batch)
Edit Generated Job     Y           (Yes/No)

Build job in Dataset   PAOLOR6.SLB.JCL
Member                DRJOB1

Job Cards:
==> //PAOLOR6D JOB (XXX,POK),DR,CLASS=A,MSGCLASS=X
==> /*JOBPARM SYSAFF=SC63
==> /*
==> /*

Press ENTER to process or PF3 to Cancel
```

Figure 8-16 Build disaster recovery job window - options for JOBCARD and JCL placement

You can build the disaster recovery jobs in batch. The JCL is shown in Figure 8-17.

```
----- Build Job for PAOLOR6.D9C1 -----

You have selected your job to be built in batch mode. The
batch generation JCL will be stored in dataset
PAOLOR6.SLB.JCL
Please specify in the dataset below where you want the
JCL built by the batch module to be placed.

Build job in Dataset   PAOLOR6.SLB.JCL

Jobcard data to be used on the generated job:
==> //PAOLOR62 JOB (XXX,POK),DRJOB2,CLASS=A,MSGCLASS=X
==> /*JOBPARM SYSAFF=SC63
==> /*
==> /*

Press ENTER to process or PF3 to Cancel
```

Figure 8-17 Building the disaster recovery JCL - actual DR JCL JOBCARD and placement options

Figure 8-18 shows the first step of the JOB for building the disaster recovery procedure.

```
//ARY@BULD EXEC PGM=ARY@BULD,REGION=006M
//*
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//          DD DISP=SHR,DSN=DB9C9.SDSNEXIT
//          DD DISP=SHR,DSN=DB9C9.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//ARYBPROF DD DISP=SHR,DSN=ARY.PROFILES
//ARYBOFFL DD DISP=SHR,DSN=ARY.OFFOPTS
//ARYBPMAP DD DISP=SHR,DSN=ARY.PROFILE.MAPS
//ARYBPCAT DD DISP=SHR,DSN=ARY.PROFILE.CATS
//ARYSBACK DD DISP=SHR,DSN=ARY.SYSBACK
//ARYSBOBJ DD DISP=SHR,DSN=ARY.SYSBACK.OBJS
//ARYSBVOL DD DISP=SHR,DSN=ARY.SYSBACK.VOLS
//ARYSBSSD DD DISP=SHR,DSN=ARY.SYSBACK.SSIDS
//ARYBREPT DD DISP=SHR,DSN=ARY.BREPORT
//ISPFIL DD DISP=SHR,DSN=PAOLOR6.SLB.JCL
//ISPLLOG DD SYSOUT=*,DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//ISPWK1 DD UNIT=SYSALLDA,SPACE=(CYL,(30,30)),
//          DCB=(RECFM=FB,LRECL=133,BLKSIZE=1330)
//ARYERROR DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//ARYDEBUG DD SYSOUT=*
//ARY#DATA DD *
    DISASTER_RECOVERY (
        DB2_SUBSYSTEM      D9C1
        GEN_TO_DATASET     PAOLOR6.SLB.JCL
        PROFILE_NAME       'D9C1'
        PROFILE_CREATOR    PAOLOR6
        EXECUTION_LIB_1    ARY.V2R1M0.SARYLOAD
        JOB_CARD_1_1       '//PAOLOR62 JOB (XXX,POK),DRJOB2,CLASS=A,'
        JOB_CARD_1_2       'MSGCLASS=X'
        JOB_CARD_2_1       '/*JOBPARM SYSAFF=SC63'
        JOB_CARD_3_1       '/*'
        JOB_CARD_4_1       '/*'
    )
/*
```

Figure 8-18 First step of batch JOB that will build the disaster recovery JCL

Figure 8-19 shows the second step of the JOB for building the disaster recovery procedure.

```
//ARY#DATA DD *
DISASTER_RECOVERY      (
  DB2_SUBSYSTEM         D9C1
  USER_INDICATOR        ARY
  PROFILE_NAME           'D9C1'
  PROFILE_CREATOR        PAOLOR6
  EXECUTION_LIB_1        ARY.V2R1M0.SARYLOAD
  GEN_TO_DATASET         PAOLOR6.SLB.JCL
  JOB_CARD_1_1           '//PAOLOR62 JOB (XXX,POK),DRJOB2,CLASS=A,'
  JOB_CARD_1_2           'MSGCLASS=X'
  JOB_CARD_2_1           '/*JOBPARM SYSAFF=SC63'
  JOB_CARD_3_1           '/*'
  JOB_CARD_4_1           '/*'
  BPROF_DSN              ARY.PROFILES
  BOFFL_DSN              ARY.OFFOPTS
  BPMAP_DSN              ARY.PROFILE.MAPS
  BPCAT_DSN              ARY.PROFILE.CATS
  SBACK_DSN              ARY.SYSBACK
  SBOBJ_DSN              ARY.SYSBACK.OBJS
  SBVOL_DSN              ARY.SYSBACK.VOLS
  SBSSD_DSN              ARY.SYSBACK.SSIDS
  BREPT_DSN              ARY.BREPORT
  POBJS_DSN              ARY.OBJECTS
  PARML_DSN              ARY.V2R1M0.SARYSAMP
  PARML_MEMBER           ARY#PARM
  DB2CNTL_DSN            ARY.DB2PARMS.CONTROL
)
/*
```

Figure 8-19 The second step of the batch job that builds disaster recovery JCL

Once the batch job listed in Figure 8-18 on page 188 and Figure 8-19 completes successfully, you will have the actual disaster recovery JCL built and added to a data set. The disaster recovery jobs are stored in the data set that you specified when you built the job via the ISPF interface.

Note: We recommend running the disaster recovery batch jobs on a regular basis. The disaster recovery batch jobs should be added to your job scheduler.

The JCL that was created performs the following steps for a system level backup:

1. Restores the System Backup and Restore Services repository at the disaster recovery site.
2. Restores the DB2 volumes from the most recent offloaded system backup.
3. Copies the archive logs.
4. Rebuilds the BSDS. The PDS that contains the disaster recovery jobs and control records will contain the contents of the BSDS. This eliminates the need to mount tapes at the recovery site to build the BSDS.

Note: Make sure that you sent the recovery PDS, which contains the disaster recovery JCL, off site. Place this data set on tape that you sent off-site with the rest of your off-site tape backups.

We assume that you will have a process in place to ship your DASD-to-Tape offloaded system level backups off-site as well.

Contents of the disaster recovery data set

In the example above we build disaster recovery JCL to be sent off-site. Table 8-1 on page 191 provides the sequenced summary and description of the members (names and description) for each job to be executed off-site. In our scenario we have a two-member data sharing environment that contains DB2 members D9C1 and D9C2.

Table 8-1 Summary of steps built in the job to RESTORE the subsystem at the remote site

Member name	Description	Comments
D9C1#JC1	Recovery JCL to RESTORE the subsystem at the remote site	<p>This job contains the following steps:</p> <ul style="list-style-type: none"> ► Step REPODFN - an IDCAMS step to define the VSAM files needed by Recovery Expert. It uses member ARYREPOC. It defines the following VSAM data sets: DB2PARMS, PROFILES, OFFOPTS, PROFILE.MAPS, PROFILE.CATS, SYSBACK, SYSBACK.OBJS, SYSBACK.VOLS, SYSBACK.SSIDS, and REPORT. ► Step REPOLOAD - invokes Recovery Expert program ARY#YMCR to load the repository records. It used member ARYREPOD. ► Step ARYREST - This step invokes the Recovery Expert restore system utility using the system level backup. ► Step D9C1DELC - an IDCAMS step to DELETE the BSDS and ACTIVE LOG data sets for both our data sharing members. It uses member D9C1DELC. ► Step D9C1ALLC - an IDCAMS step to DEFINE the BSDS and ACTIVE LOG data sets. It uses member D9C1ALLC. ► Step D9C1CATL - an ICAMS step to verify that all IC data sets are cataloged at the disaster recovery site. It uses member D9C1CATL. ► Step D9C1RBSR - invokes a Recovery Expert program, ARY@YRBS, to rebuild the BSDS into 4,080-byte records. It uses member D9C1BSDS. ► Step D9C1CRCR - an IDCAMS step that performs a REPRO to insert the BSDS data created in the previous step, D9C1RBSR, into the two BSDS data sets. ► Step D9C1PRNT - invokes DSNJU004 to print the contents of the BSDS for member D9C1. ► Step D9C2RBSR - invokes a Recovery Expert program, ARY@YRBS, to rebuild the BSDS into 4,080-byte records. It uses member D9C1BSDS. ► Step D9C2CPBS - an IDCAMS step that performs a REPRO to insert the BSDS data created in the previous step, D9C1RBSR, into the two BSDS data sets. ► Step D9C2CRCR - invokes DSNJU003 to create a CRCR in the BSDS. It uses member D9C1CRCR. This is for the second DB2 member D9C2. ► Step D9C1DELA - an IDCAMS step to delete the ARCHIVE data sets on tape. ► Step D9C2YCPL - invokes the Recovery Expert program, ARY@YCPL, to copy the archives logs from tape to DASD. It uses member D9C1CPYL.
D9C1#JC2	RESTORE SYSTEM with the LOGONLY option	<p>This step invokes the IBM DSNUTILB stand-alone utility to restore the system logs. DB2 must be up and you must have responded yes to the WTOR: XXXX CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT RBA XXXXXXXXXXXX. REPLY Y OR N.</p>
D9C1#JC3	Restarts all the DB2 spaces in RW mode	<p>This job is optional. This disaster recovery job can be run after the DB2 subsystem has been started and the DB2 catalog and directory have been successfully restored. Although optional, we recommend running this job. It uses member D9C1STRF.</p>
ARYREPOC	Defines clusters for Recovery Expert VSAM files	<p>This member contains all the IDCAMS DEFINE CLUSTERs for the Recovery Expert VSAM files.</p>

Member name	Description	Comments
ARYREPOD	Controls records (data) for the Recovery Expert VSAM files	This member is a data file containing the data that is loaded into the Recovery Expert VSAM files.
D9C1ALLC	Defines the VSAM clusters for the DB2 catalog and directory, BSDS, active logs, and user-defined VCAT spaces	This member is used by job D9C1#JC1.
D9C1BSDS	Contains a copy of the bootstrap data set in 80-byte records	This member is used by job D9C1#JC1.
D9C1CATL	Contains control cards for cataloging the image copies at the recovery site	This member is used by job D9C1#JC1.
D9C1CPYL	Contains control cards for copying the archive logs from the tape to the recovery site	This member is used by job D9C1#JC1.
D9C1CRCR	Contains the control card for creating the system point- in-time conditional restart record	This member is called twice in job D9C1#JC1. It is called once for each DB2 member.
D9C1DELA	Contains control cards to delete the archive logs (with the NOSCRATCH option) from the recovery site z/OS catalog (Those logs are on tape and will be copied to DASD, then cataloged.)	This member is used by job D9C1#JC1.
D9C1DELC	Contains control cards to uncatalog the active logs and BSDS at the disaster site if they exist	This member is used by job D9C1#JC1.
D9C1STRF	Contains control cards for starting application table spaces in the DB2 subsystem with RW access	This member is used by job D9C1#JC3.
D9C2BSDS	Contains a copy of the bootstrap data set in 80 byte records	This member is used by job D9C1#JC1. This is for the second DB2 member, D9C2.

8.3 Restoring steps at the disaster recovery site

In the disaster recovery example that follows, 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.

We assume that the z/OS catalog is available and has been restored at the remote site.

The following steps describe and also show real job output for a successful system level restore using Recovery Expert.

Run job ssid#JC1 - D9C1#JC2

The first job to run at the disaster recovery site is job ssid#JC1. In our example it is job D9C1#JC1. This job performs the following steps:

1. Defines the System Backup and Restore Services data repository files.
2. Loads the System Backup and Restore Services data repository files required for disaster recovery processing. See Example 8-2.

Example 8-2 Output from the system backup offload job run at the local site

```
Recovery Expert for DB2 z/OS
Backup Summary Report

Utility Executed:..... Restore
Profile Name:..... PAOLR6.D9C1
DB2 Subsystem:..... D9C1
DB2 Version:..... 0910
Restore Type:..... Offload
Restored:..... Object Data Only
Nbr of Volumes:..... 0010
Backup RBA:..... C20DC63434CC
Recovery Expert for DB2 z/OS
Backup Volume Detail Report

<--DB2 Source Volume-->
Volser Ucb# Sym# Devtyp Restored From Filename File
-----
SBOX5B D900 0000 3390-9 PAOLR6.D9C1.D066.OFFLOAD1.P.SBOX7J
...
...
SBOX5R D902 0000 3390-9 PAOLR6.D9C1.D066.OFFLOAD1.P.SBOX7H 004
SBOX5S DA02 0000 3390-9 PAOLR6.D9C1.D066.OFFLOAD1.P.SBOX7A 005
SBOX5T DB02 0000 3390-9 PAOLR6.D9C1.D066.OFFLOAD1.P.SBOX7B 005
```

3. Invokes the System Backup and Restore Services job to restore the volume backups from tape. See Example 8-3.

Example 8-3 Output from the step to restore the volumes backups from tape

```
ARYS001I - Recovery Expert for DB2 z/OS Starting. Version 02.01.013
ARYS162I - Parmlib used for this execution
ARYS003I - Control Cards:
ARYS004I - RESTORE PAOLR6.D9C1
ARYS004I - GENERATION 02
ARYS004I - DATE 03/06/2008
ARYS004I - TIME 17:17:40
ARYS004I - RECOVERY-SITE
ARYS004I - FROM-OFFLOAD
```

```

ARYS004I -
ARYS123I - Backup PAOL0R6.D9C1 generation 02 was read from the repository.
ARYS013I - Backup profile PAOL0R6.D9C1 was read from the repository.
ARYS038I - Performing profile volume map validation...
ARYS146I - Removing volser SB0X5C from this restore. It contains only log data
ARYS146I - Removing volser SB0X5K from this restore. It contains only log data
ARYS146I - Removing volser SB0X5D from this restore. It contains only log data
ARYS146I - Removing volser SB0X5E from this restore. It contains only log data
..
..
ARYS039I - Volume map validation complete.
ARYS137I - Varying volumes offline.
ARYS136I - Disconnecting user catalogs.
ARYS217I - User catalog UCAT.DB9CDAATA disconnected.
ARYS217I - User catalog UCAT.DB9CLOGS disconnected.
ARYS217I - User catalog UCAT.VSBOX01 disconnected.
ARYS137I - Varying volumes online.
ARYS004I - Restoring volumes from offloaded backup...
ARYS277I - Task 02 - Volser SB0X5B was restored from PAOL0R6.D9C1.D066.OFFL0D1.
ARYS277I - Task 01 - Volser SB0X5I was restored from PAOL0R6.D9C1.D066.OFFL0D1.
ARYS277I - Task 02 - Volser SB0X5J was restored from PAOL0R6.D9C1.D066.OFFL0D1.
..
..
ARYS277I - Task 02 - Volser SB0X5T was restored from PAOL0R6.D9C1.D066.OFFL0D1.
ARYS277I - Task 01 - Volser SB0X5U was restored from PAOL0R6.D9C1.D066.OFFL0D1.
ARYS277I - Task 02 - Volser SB0X5V was restored from PAOL0R6.D9C1.D066.OFFL0D1.
ARYS002I - Recovery Expert for DB2 z/OS complete. RC=000.

```

4. Issues IDCAMS DELETE to delete all BSDS and active log data sets. See Example 8-4.

Example 8-4 Output from the deleting the BSDS and log data sets

```

IDCAMS  SYSTEM SERVICES

      DELETE ('DB9CL.D9C1.BSDS01')
IDC0550I ENTRY (D) DB9CL.D9C1.BSDS01.DATA DELETED
IDC0550I ENTRY (I) DB9CL.D9C1.BSDS01.INDEX DELETED
IDC0550I ENTRY (C) DB9CL.D9C1.BSDS01 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DELETE ('DB9CL.D9C1.BSDS02')
IDC0550I ENTRY (D) DB9CL.D9C1.BSDS02.DATA DELETED
IDC0550I ENTRY (I) DB9CL.D9C1.BSDS02.INDEX DELETED
IDC0550I ENTRY (C) DB9CL.D9C1.BSDS02 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      DELETE ('DB9CL.D9C1.LOGCOPY1.DS01')
IDC0550I ENTRY (D) DB9CL.D9C1.LOGCOPY1.DS01.DATA DELETED
IDC0550I ENTRY (C) DB9CL.D9C1.LOGCOPY1.DS01 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
...

```

5. Define the DB2 active log and BSDS data sets with the proper allocations. See Example 8-5.

Example 8-5 Output of defining the active logs and BSDS data sets

```

IDCAMS  SYSTEM SERVICES

      DEFINE CLUSTER
          ( NAME ('DB9CL.D9C1.BSDS01')

```

```

REUSE -
RECORDSIZE(4089 4089) -
FREESPACE(0 20) -
KEYS(4 0) -
CONTROLINTERVALSIZE(04096) -
STORAGECLASS(DB9CLOG1) -
MANAGEMENTCLASS(MCDB22) -
VOLUMES(SBOX5F) -
TRACKS(00000078,00000002) -
SHAREOPTIONS(2 3) ) -
DATA -
( NAME ('DB9CL.D9C1.BSDS01.DATA') ) -
INDEX -
( NAME ('DB9CL.D9C1.BSDS01.INDEX') ) -
IDC0508I DATA ALLOCATION STATUS FOR VOLUME SBOX5E IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME SBOX5E IS 0
IDC0181I STORAGECLASS USED IS DB9CLOG1
IDC0181I MANAGEMENTCLASS USED IS MCDB22
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
...

```

6. Rebuilds the BSDS from the 80-byte record file, placing it back into 4,089-byte records.
7. Restores the BSDS by placing the 4,089-byte records into a VSAM file. See Example 8-6.

Example 8-6 IDCAMS REPRO output for the restore of the BSDS

```

IDCAMS  SYSTEM SERVICES

      REPRO INFILE(BSDSI) OUTFILE(BSDS1) REUSE
IDC0005I NUMBER OF RECORDS PROCESSED WAS 777
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      REPRO INFILE(BSDSI) OUTFILE(BSDS2) REUSE
IDC0005I NUMBER OF RECORDS PROCESSED WAS 777
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0

```

8. Creates a conditional restart. See Example 8-7.

Notice that this is a different type from the image copy DR method. It is a SYSPITR (system point in time record).

Example 8-7 Conditional restart record creation output

```

DSNJCNVB CONVERSION PROGRAM HAS RUN   DDNAME=SYSUT1
DSNJCNVB CONVERSION PROGRAM HAS RUN   DDNAME=SYSUT2
      CRESTART CREATE,SYSPITR=C20DCC61FA97,FORWARD=YES,BACKOUT=YES
DSNJ408I DSNRJFCK CHECKPOINT RBA FOUND, RBA = 0000126F8B90, TIME = 22:45:09 MAR
DSNJ411I DSNRJRCR CRESTART CREATE FOR CRCRID = 0006, DDNAME = SYSUT1
DSNJ408I DSNRJFCK CHECKPOINT RBA FOUND, RBA = 0000126F8B90, TIME = 22:45:09 MAR
DSNJ411I DSNRJRCR CRESTART CREATE FOR CRCRID = 0006, DDNAME = SYSUT2
DSNJ225I CRESTART OPERATION COMPLETED SUCCESSFULLY
DSNJ200I DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING COMPLETED SUCCESSFULLY

```

9. Prints the contents of the BSDS.
10. Uncatalogs the tape archive logs.
11. Copies the uncataloged tape archive logs to DASD and catalogs them to speed up the recovery at the DR site.

Change DSNZPARM for DR

Change your remote site DSNZPARM settings as follows:

1. Change RESTART to DEFER.
2. Set the site as local or recovery c.
3. Change the SYSADM and SYSOPR user IDs to the user ID that will be recovering the DB2 catalog.

Delete the structures in the XCF

In our example we are RESTORING two members in a data sharing environment. Deleting the LCA and LOCK structures in the coupling facility is very important.

Tip: With PTF for PK59675, Recovery Expert automatically resets the CF structures before RESTORE in data sharing.

Start DB2

Start DB2 using the new reassembled DSNZPARMs. We recommend that DB2 be started with ACCESS(MAINT). In a data sharing member, ensure that you start all the DB2 members.

Reply to conditional restart message

For each DB2 for each subsystem, reply Y to the outstanding message. See Example 8-8.

Example 8-8 Replying to CRCR outstanding message

```
448 DSNJ245I  -D9C2 CONDITIONAL RESTART RECORD INDICATES TRUNCATION
AT LRSN C20D7649C742.  REPLY Y TO CONTINUE, N TO CANCEL
R 448,Y
```

Run job ssid#JC2 (D9C1#JC2)

For the system level backup method, job #JC2 executes the IBM RESTORE SYSTEM LOGONLY utility. Since the DB2 object data sets were restored from the system level backup, this utility applies log records to all objects in the system to restore them to the most recent point available at the disaster recovery site.

Run job ssid#JC2 (D9C1#JC3)

Job #JC3 is used to force application table spaces to start in RW mode. This job starts all application table spaces in the DB2 subsystem with read-write access. While optional, we recommend this job. If any table space is not in RW mode, the recovery will fail.

ReSTART DB2 - all members

If you started DB2 with ACCESS MAINT, stop DB2 and restart it using normal access. You need to do this for each member in a data sharing environment.

Appendixes

This part contains referenced appendixes:

- ▶ Appendix A, “Frequently asked questions” on page 199
- ▶ Appendix B, “Recommended maintenance” on page 207
- ▶ Appendix C, “Storage and DB2 functions” on page 209
- ▶ Appendix D, “Contents of schema level repository” on page 247
- ▶ Appendix E, “Sample jobs” on page 263

Archived

Frequently asked questions

Recovery Expert has been available for a few years and it keeps evolving very quickly. Since Version 1 and up through the major enhancements on system level backup support available with Version 2, a number of questions about how to best utilize the tool can be asked.

In this appendix we provide a list of frequently asked questions, and provide a short answer for each, followed by a reference to a manual or book where you can access more detailed information. For those topics that are covered in more detail in this book, we provide a link to the appropriate chapter.

Table A-1 contains a list of frequently asked questions that deal with recovery topics.

Table A-1 Application development frequently asked questions

No.	Question	Answer
1.	I could not RECOVER a table space from either ISPF or GUI using a system level backup (SLB) created by Recovery Expert.	Make sure to set the Enable Obj Restore option to Y in the backup profile before creating the SLB. See 4.4, "Profiles" on page 80.
2.	Can I perform a recovery of an object using the SLB and no log apply?	No, because SLBs are fuzzy copies. In order for Recovery Expert to use the SLB, you need to specify a point that is at least RBA/LRSN+1 of the SLB. See 6.1.3, "Recover object to point in time using system level backup" on page 138.
3.	What is the purpose of the RBA capture utility? When do I need this utility?	This is used only from the ISPF interface and is useful in non-data sharing to create TIMESTAMP to RBA relationships. This utility captures/records RBA/LRSN information about user-defined time intervals. This information can be used to determine which RBA/LRSN to use to recover the SUBSYSTEM corresponding to a given time stamp (PIT). If the PIT recovery RBA/LRSN is different from that of the f system backup or image copy or quiesce point then this could be used to pick a desired point manually. See "Step 21 - Configure the RBA capture utility" on page 32.
4.	After adding a new subsystem to the product control file, I started the agent started task successfully, but I am still not seeing this new system from the GUI client. What is wrong?	If the client was brought up before you started the agent then you must close the client window and reconnect to the appropriate server to see the new system.
5.	Is partition-level restore possible with the ISPF interface?	Yes, you can select individual partitions using the "Object Recovery Profile" option. See "Add objects to object profile" on page 128.
6.	Can I perform index level RESTORE using SLB via the ISPF component?	Yes, you can create an Object Recovery profile using Recovery Expert to include this index and build a recovery job. If the index is created COPY YES, the generated job will recover the index from either an SLB or an image copy if one is found. If the index is created COPY NO, the generated job will rebuild the index.
7.	Is it possible to restore a prior version of DDL (pertaining to a table space and all its dependent objects) using the ISPF interface?	No. This feature is available only through GUI. See the previous companion IBM Redbooks publication <i>IBM DB2 Recovery Expert for z/OS User Scenarios</i> , SG24-7226.
8.	If I want to restore a table space to a specific copy, which interface should I use?	GUI. But you may also use ISPF if the specific copy that you are interested in recovering to is the LASTCOPY or you would just have to know the RBA/LRSN of the copy that you wish to restore to and enter the RBA/LRSN of the copy + 1. See 6.1.4, "Recovery object to copy (full or incremental copy or quiesce)" on page 138.

No.	Question	Answer
9.	After I have Recovery Expert installed, if I am using it for system level backups, should I no longer use DB2's BACKUP SYSTEM and RESTORE SYSTEM?	Information about SLB taken via Recovery Expert is maintained in a VSAM repository and used by Recovery Expert for subsequent RESTOREs. If you run BACKUP YSTEM outside of Recovery Expert this information is not updated and Recovery Expert will not be able to execute the RESTORE, when needed. Please note: <ul style="list-style-type: none"> ► We recommend using Recovery Expert to manage your system level backups. ► There are many benefits such as object level recovery in DB2 versions prior to DB2 9. ► Backup through Recovery Expert provides validation.
10.	Which component of DB2 Recovery Expert uses the ARYV210 CLIST?	The ARYV210 CLIST is used by the ISPF interface. It is not used by the GUI.
11.	Is there any dependency between the ARYV21 CLIST and the schema level repository?	No.
12.	Why can we not see the object profiles created through DB2 Recovery Expert (object Recovery Profile option) from the GUI, whereas we can see the object profiles created via DB2 Automation Tool?	The object profiles created through DB2 Recovery Expert (Object Recovery Profile option) are stored on VSAM repository files, and the GUI interface is not designed to work with this VSAM repository. But, the DB2 Automation Tool profiles (even though they are created via the ISPF interface) are stored in DB2 tables (DLC), and the GUI can access these DB2 tables.
13.	Can I perform disaster recovery from the GUI interface?	No. See Chapter 8, "System level backup based remote recovery" on page 173.
14.	If the agent started task is not running on the LPAR that I am logged on, can I still invoke Recovery Expert (ISPF interface) from that LPAR and create object recovery profiles and build recovery jobs?	Yes. None of the functions available from the ISPF interface require or use the RE Agent started task, the RE Server started task, or the SLR.
15.	What recovery options are available through the ISPF interface but not available through GUI?	<ul style="list-style-type: none"> ► Configuration ► System backup profiles ► Subsystem analysis and backup of an entire system (using FlashCopy, DB2, BCVs, or Snap) ► Object recovery to last quiesce point for each object included in the profile ► Displaying of RBA capture information ► Disaster recovery (remote site)
16.	What recovery options are available through GUI but not available through ISPF?	<ul style="list-style-type: none"> ► Dropped object recovery ► Log analysis ► Restore tables ► Restore objects related to DB2 plans ► Restore objects that are part of an object profile created via automation tool

No.	Question	Answer
17.	<p>How many product control files (PCF) and agent started tasks are required if I am in a 3-way data sharing group involving three LPARs?</p> <p>Follow-up question: How many more agents do I need if I have two more data sharing groups in the same three LPARs?</p>	<p>One PCF and three agents.</p> <p>None, you just need to update the product control file with all the member subsystem information and group attach name.</p>
18.	In a non-data sharing environment how many agent started tasks are needed for eight subsystems in three LPARs?	Three. It is independent of the number of subsystems.
19.	If I have only one server started task to manage all the subsystems (and data sharing groups) in my environment, then it looks like there is no difference in the contents of the AGENT configuration file. But, since I need to have multiple agent started tasks (one per LPAR), can I have this AGENT config file stored in a shared DASD and avoid duplication of efforts?	<p>Yes.</p> <p>See 3.1, "GUI environment configuration" on page 40.</p>
20.	When I restore a table space using an SLB, will the indexes pertaining to that table space be restored too?	<p>Yes in all versions of DB2.</p> <p>When using the ISPF interface, the indexes will be recovered/rebuilt depending on their definition. If they are copy, yes, they will also be restored from the SLB. If they are COPY NO, they will be rebuilt. The GUI will always rebuild the indexes.</p>
21.	How can I restore a view after it is dropped?	You cannot directly restore a view using Recovery Expert, but it will handle views for recreated tables.
22.	We have FlashCopy V1.2. Can I use Recovery Expert to restore a table space from SLB?	No.
23.	If I do not have FlashCopy available then can I still buy Recovery Expert for z/OS?	Yes. The only lost functionality is the functions that require Flash-enabled DASD. Also, if EMC DASD is installed, the Flash emulation is not required. Recovery Expert can drive the native EMC hardware functions.
24.	Can I recover LOB table spaces via the ISPF interface?	Yes.
25.	Do I have any control over which recovery option (like DB2 SYSTEM RESTORE, RECOVER TOLOGPOINT, RECOVER TOCOPY, or RESTORE) tool to use when I try to recover a table space?	<p>No, the ISPF interface will chose for you. Recovery Expert tool will find the best possible recovery/restore option.</p> <p>The GUI presents all plans and lets you chose. However, APAR PK59675 gives you this control. You can chose whether you want to consider SLBs, ICs, or both, as recovery methods as an option when an object profile recovery job is built.</p>

No.	Question	Answer
26.	How is Recovery Expert related to the Application Recovery Tool?	The two products are separate and there is no interface between the two. IBM Application Recovery Tool for IMS and DB2 Databases works with either DB2 or IMS or with both. It uses image copies for both databases and establishes synchronization points in each called a virtual image copy to which recovery may be done. It also establishes a quiesce point and notes where the point is located on the respective logs. During recovery, the user specifies the Virtual Image Copy, and the Application Recovery Tool applies the appropriate image copies and causes the log to be applied up to this point. Users can also use the Application Recovery Tool to automate the recovery of resources. The tool will generate the job control language, locate the proper image copies, and control the execution of jobs for either DB2 or IMS.
27.	When I try to restore a set of table spaces pertaining to an application using SLB, am I assured business continuity of other applications running on the same DB2 subsystem?	Yes. See "How System Backup and Restore Services performs object recovery" in Chapter 13 of <i>DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1</i> , SC19-1222.
28.	When I try to create an object level recovery using the Recovery Expert tool using the Recover to Last Copy option, the tool generates RECOVER TABLESPACE TOLOGPOINT with LRSN/RBA =, the START_RBA+1 pertaining to the LAST COPY rather than RECOVER TO LASTCOPY. Why?	The ISPF interface generates syntax that will recover the object to the image copy by generating this syntax. It does this to generate the most efficient recovery syntax it can. If there are multiple table spaces (or indexspaces) being recovered to image copies taken at the same time, the generated syntax will support multi-tasking.
29.	If we are not interested in using GUI then should I still set UPDATE IPC IPC_RBR = Y during product control file configuration in ARYSJ001?	No. See "Step 8 - Configure the Recovery Expert product control file" on page 24.
30.	Where does Recovery Expert (ISPF component) get the DB2 subsystem information, load library, and plan information from?	From the product control file (which is updated during installation or anytime there is change in the system configuration) through ARYSJ001. If the information is not provided by running ARYSJ001 then you may enter the DB2 subsystem, load library, and plan information online by selecting the M option (DB2 Subsystem Analysis and Setup) and following the panels that appear.
31.	My workstation is running Windows Vista®. Can I install Recovery Expert client and expect normal behavior?	It has not been tested thoroughly, but there are no issues reported as of this writing.
32.	Can I perform a redirected restore (similar to LUW) using Recovery Expert for z/OS system level backup to create a clone of my development subsystem on a different LPAR?	No. IBM DB2 Cloning Tool for z/OS is a vendor-independent tool that provides access to data sets on replicated volumes created with point-in-time copies of Fast Data Replication tools (FlashCopy and SnapShot) or Splits of Continuous Mirrors tools (EMC TimeFinder, IBM PPRC, HDS ShadowImage, Softek TDMF™, and Fujitsu Equivalent Copy). In addition, customers can easily create test and QA environments and refresh the data on a regular basis.

No.	Question	Answer
33.	If my restore system job abends can I restart the job automatically without any manual intervention?	You can submit the job again. If the restore is from a system backup on tape and it abends, adding the "RESTART" keyword will cause the restore to pick up from where it left off. This support was added by APAR PK59675. See also 5.2, "System level backup and restore using Recover Expert ISPF-based System Backup and Restore Services" on page 99.
34.	We have installed the Recovery Expert V2 Client/Server component only. The client generates recovery JCL for a given database correctly. The image copy step generates data set names that do not conform to our naming standards. Where is the template for the data set names, or how can I make the names generate according to our naming standards?	Currently, there is not an easy way of doing this except by manual intervention. We are working on user-controlled mechanisms (templates, parms, and so on) to be added so that customers can control data set naming.
35.	Can Recovery Expert's backup profile SETUP process verify that the target DASD units of the current backup are not in use by any other backup profile created on the current LPAR but used in a different DB2 subsystem?	Yes. But, if the target volumes are used by a different subsystem that belongs to a different LPAR (and if the DASD volumes are shared across these two LPARs), then Recovery Expert's profile SETUP process will not know that and may result in data corruption.
36.	Can I restore a dropped object via the ISPF Interface?	No.
37.	When I try to restore a table without any unique or PRIMARY KEY are there any restrictions?	When you are selecting tables for recovery, remember that any recovery plan might include undo or redo SQL. You must ensure that there is a primary, or unique, key on the tables for which you are generating SQL. Without a primary key, you may not get the desired results because of the inability to uniquely identify rows that were changed by the original SQL.
38.	Why is the point-in-time conversion job not being generated sometimes?	DB2 Recovery Expert generates and submits this batch job during recovery plan generation when you specify either a time stamp or a log RBA as the point in time for the recovery, when running against a non-data sharing subsystem. The job performs log analysis to convert from the time stamp that you specified to the corresponding log RBA, or vice versa. If you browse recovery history events or quiet times as the source for selecting the point in time, then both time stamp and log RBA information are already available (although only one or the other is displayed in the user interface). In this case, DB2 Recovery Expert automatically uses both pieces of information, and does not require the point-in-time conversion job. See "Point in time conversion job" in Chapter 7 of <i>DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1</i> , SC19-1222.
39.	I am running DB2 Version 7. Can I still use Recovery Expert to perform object level restore from SLB?	Yes. Ensure that the prerequisites listed in this book are met.
40.	What System Backup and Restore Services repository data sets are needed at the DR site?	They are only needed when using the system backup method of DR, and it needs the ones that it automatically restores there.

No.	Question	Answer
41.	If I want to perform disaster recovery using System Backup and Restore Services system level backup to restore the entire subsystem in a remote location, can I do this without the offload option on my backup profiles?	No. The offload is needed to get the backup to the remote site. You might be able to perform the offload to a remote distance at the DR site if your system is configured for such an operation. See 8.1, "Offloading to tape at the local site" on page 174.
42.	The DB2 V8 Utility Guide (Chapter 5. BACKUP SYSTEM) says that "each DB2 subsystem can have up to two copy pools, one for databases and one for logs. BACKUP SYSTEM copies the volumes that are associated with these copy pools at the time of the copy." Can I have the RUNLIB and DSNEXIT libraries on the same copy pool as DB2 logs?	We strongly recommend not having RUNLIB and DSNEXIT libraries on these copy pools, because a subsequent system restore can corrupt these libraries depending on when the backup was taken and whether there are any changes to these libraries since then.
43.	Using Recovery Expert tool, can I restore the subsystem to an SLB taken in the prior version of DB2?	Yes.
44.	We have an SLB taken while running Recovery Expert v1.1. Can I use it to restore the subsystem after I migrate Recovery Expert to V2.1?	Yes, but only with the GUI. The ISPF interface does not support restoring from system backups that it did not perform.
45.	If my product control file is on a shared DASD (shared across all my LPARs), then can I just use one product control file for my entire environment?	Yes, it makes sure to add control information for each and every subsystem (and the data sharing group information, if any) into this file.
46.	If I am only running one server started task, then my agent config file is going to have the same content across all the LPARs/subsystems (assuming that my product control file name is same across all the lpars/subsystems), so if I just copy the same configuration proc member (that is, the same name) to all my SYS1.PROCLIB and start the started task, will it work?	Yes.
47.	Will Recover Expert choose an SLB taken using the BACKUP SYSTEM utility in V8 (rather than SLB created via Recover Expert) to arrive at an optimal system recovery solution?	Yes.

No.	Question	Answer
48.	Can I recover a partition of a PBG partitioned table space?	<p>No. Recovery of a partition by growth partitions is only available for all partitions at the table space level. Partitions cannot be recovered individually.</p> <p>See "Summary of changes" in Chapter 1 of <i>DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1</i>, SC19-1222.</p>
49.	If I have four tables involved in an RI constraint and if I want to restore only one of those tables, are there any consequences?	<p>If all tables involved in the RI constraints were not present in your results (that is, they were filtered out), the generated SQL might omit any such RI records, as proper execution requires all such tables. This can lead to an empty SQLOUT DD data set, which usually contains all the SQL. This happens for recovery to a prior point in time. A recovery to current will cause no errors.</p>
50.	How do I delete a subsystem (or a data sharing group) from being managed from an Recovery Expert Server?	<p>You have to delete the product control file and recreate it and then run ARYSJ001 without that subsystem (or data sharing group) information.</p>

Recommended maintenance

In this appendix we look at recent maintenance for Recovery Expert V2.1. These APARs represents a snapshot of the current maintenance at the moment of writing. As such, the list will become incomplete or even incorrect at the time of reading. It is here to identify the level of code used during the project and provide recommendations for possible improvements.

Make sure that you contact your IBM Service Representative for the most current maintenance at the time of your installation. Also check on RETAIN® for the applicability of these APARs to your environment, as well as to verify prerequisites and post-requisites.

Table B-1 Recovery Expert V2.1 current APARs

APAR #	Area	Text	PTF and notes
PK53379	Log analysis, DB2 V7	GA maintenance.	UK30484
PK55713	ISPF interface	GA maintenance.	UK32857
PK55821	Multiple issues in RE V2.1	GA maintenance.	OPEN
PK57092 PK57163 PK58428 PK58430 PK58431 PK58433 PK58434 PK58928 PK58945	Multiple issues in RE V2.1	GA maintenance.	UK33508
PK57183	Hang	Hang situation with z/OS 1.9.	UK33601
PK61889	Delete tables	Incorrect delete objects during recover if z/OS earlier than 1.8.	OPEN
PK60078	Samplib	Corrections and documentation enhancements to various SAMPLIB members such as ARYBIND7, ARYBIND8, ARYBIND71, ARYBIND81, clist ARYV210.	OPEN

APAR #	Area	Text	PTF and notes
PK55713	Roll up	GA maintenance.	UK32857
PK62568	Recover	Recovering a dropped tablespace a S0C4 may occur when using ARYLDSN1 to invoke DSN1COPY.	UK34785
PK62974	Message	Message ARYX094S--An unexpected error occurred with token "0" may be incorrectly issued during recovery plan generation. This can occur when trying to generate recovery plans to a specific RBA or a hard coded timestamp for an object defined on a DB2 subsystem that is partially set up for datasharing. The DB2 subsystem does not belong to a datasharing group but has a group attach name defined in the users SYS1.PARMLIB(IEFSSNxx) member.	UK35022
PK59664 PK59666 PK60696 PK61814 PK62088 PK59675 PK60716 PK60786 PK60788	Multiple issues with RE V2.1	GA maintenance.	UK35495
PK61889		DSNU054I ssid DSNUGMAP - TABLESPACE 'SYSTOOLS.ARYROLES' NOT FOUND running SARYSAMP(ARYCPY1).	OPEN
PK62742		Optional indexes to improve SLR update will be added to SARYSAMP lib member ARYDSCIX for customers that are having issues with SLR update performance.	OPEN
PK62922		Optional indexes to improve SLR update will be added to SARYSAMP lib member ARYDSCIX for customers that are having issues with SLR update performance.	OPEN
PK29628		User interface does not indicate that agent tasks/jobs are not currently connected to the server. Users may experience an empty list when trying to select a DB2 subsystem.	OPEN

Pertinent DB2 APARS are listed in Table B-2.

Table B-2 DB2 Backup and Restore current APARs

APAR #	Area	Text	PTF and notes
PK23288	Check index shrlevel change	Flascopy is rejected if the target disk is a PPRC primary volume	SUG
PK51979	Restore	RESTORE SYSTEM Utility can be used to recovery DB2 to the current end of log, no log truncation required	UK31997 V9
PK49366	Restore	Fallback to a previous backup for restore and recover	OPEN
PK63692	Restore	Restore fails with system managed duplexing	UK35488 for V9 UK35487 for V8

Storage and DB2 functions

In this appendix we provide an overview of the advanced storage (disk and tape) and DB2 functions that are utilized for backup and restore of DB2 objects and DB2 subsystems. The storage functions are provided by the family of Enterprise Storage Servers and the follow-on devices.

The DB2 backup and restore functions are introduced with DB2 V8 and have been further enriched by DB2 9 for z/OS.

This appendix is a subset of the information contained in *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370, and *DB2 9 for z/OS Technical Overview*, SG24-7330. It is included here for convenience.

This appendix covers the following topics:

- ▶ FlashCopy
- ▶ DFSMS functions for DB2 point in time recovery
- ▶ DB2 BACKUP and RESTORE functions in V8
- ▶ BACKUP and RESTORE enhancements with DB2 9 for z/OS

FlashCopy

In this section we briefly introduce the FlashCopy function provided by the family of IBM Enterprise Storage Servers (ESS). 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 Metro and Global Mirroring, 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 section covers the following topics:

- ▶ Introducing FlashCopy
- ▶ How FlashCopy works
- ▶ How to invoke FlashCopy
- ▶ DFSMSDss utility
- ▶ Incremental FlashCopy
- ▶ FlashCopy consistency group

For more information about 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.

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 IBM TotalStorage Enterprise Storage Server, with minimal impact on other ESS activities.

There are currently two FlashCopy versions available: Version 1 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 ESSs.
- ▶ 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.

FlashCopy V2 can produce data set copies as well as volume copies. The DB2 9 for z/OS Recover utility can operate at the DB2 object level from a volume backup.

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 that 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 that 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 outlined in Figure C-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.

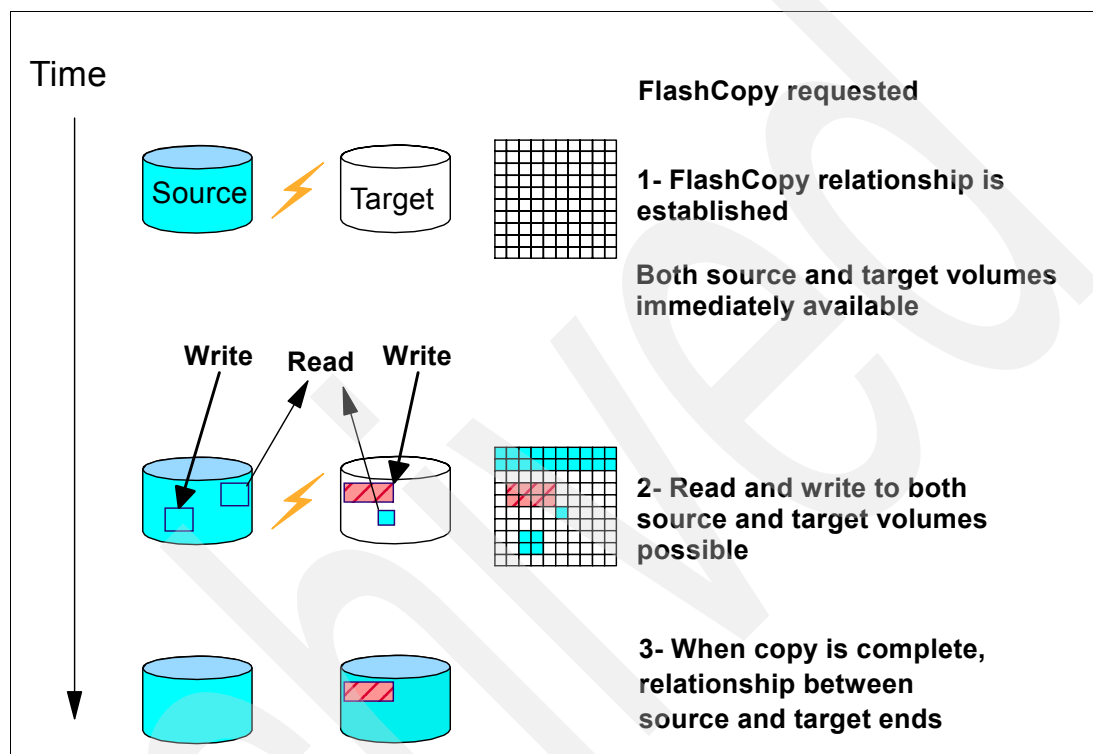


Figure C-1 FlashCopy with background copy

Phase 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.

Phase 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 that 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.

Phase 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 the ESS Copy Services configuration and use for PPRC and FlashCopy for the ESS Model 800.

Methods are described in detail in the book *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM eServer zSeries*, SG24-5680-04, and its bibliography.

DFSMSDss utility

In this section we discuss how to use the DFSMSDss utility to invoke FlashCopy functions. For more detailed information about DFSMSDss, refer to *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, the FCTOPPPRCPRIMARY 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, DFSMSDss (in this case) 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.

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

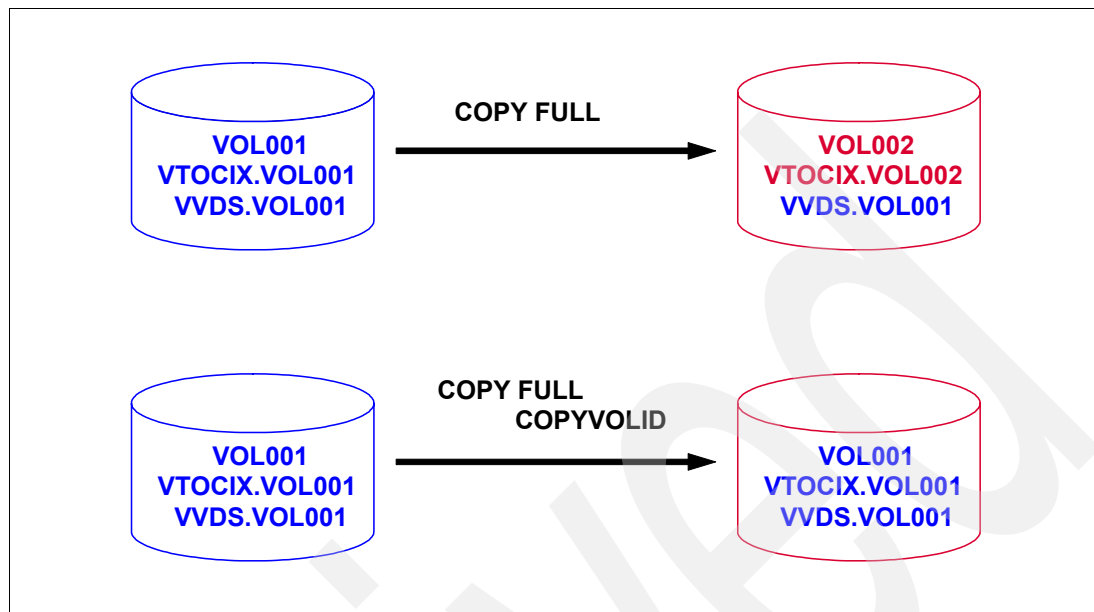


Figure C-2 DFSMSdss COPY with COPYVOLID

Example C-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: C-1 DFSMSdss COPY FULL with COPYVOLID

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

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 C-3 on page 216 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 though 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.

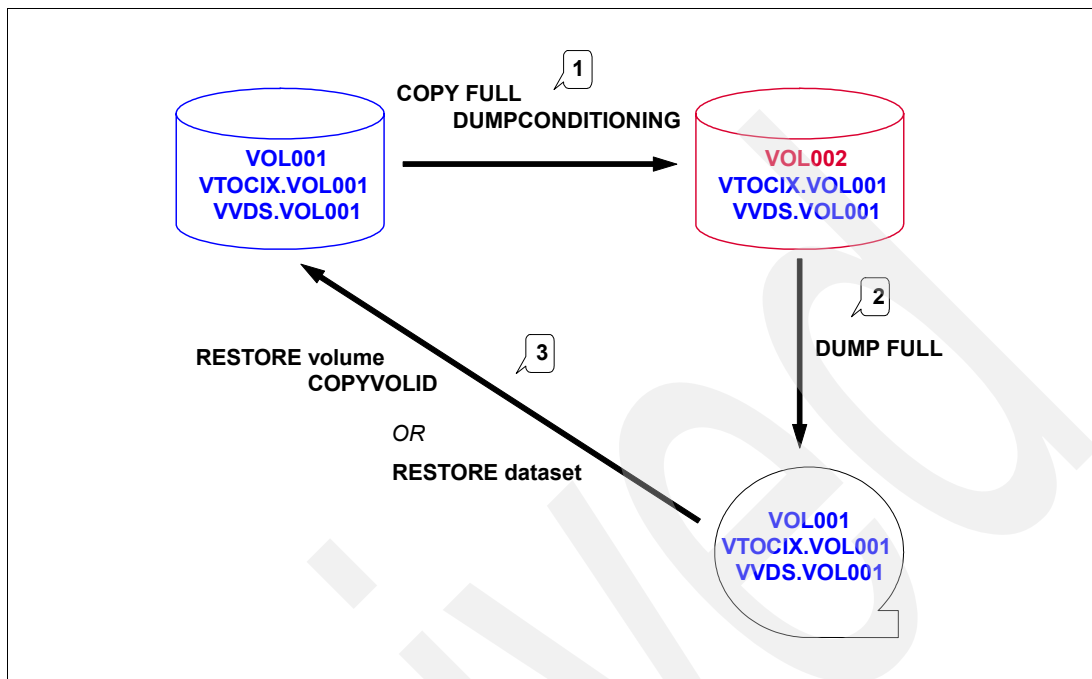


Figure C-3 DFSMSdss COPY with DUMPCONDITIONING

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

Example: C-2 Backup-restore cycle with DUMPCONDITIONING and background copy

```

//BACKUP JOB .....
//*
/* Step 1 - COPY FULL with DUMPCONDITIONING
/*
//STEP1 EXEC PGM=ADDRSSU
//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=ADDRSSU
//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=ADDRSSU
//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 C-3 on page 216, 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 that 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.

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.

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 that 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. 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 C-3 illustrates how the backup cycle (steps 1 and 2) shown in Figure C-3 on page 216 can now be implemented with the no-background copy.

Example: C-3 Backup to tape with no-background copy

```
//BACKUP JOB .....
/*
/* Step 1 - COPY FULL with DUMPCONDITIONING & FCNOCOPY
/*
//STEP1 EXEC PGM=ADDRSSU
//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
/*

```

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 the 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 C-4 on page 219 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, and 5 updated, and volB had tracks 2, 7, and 9 updated, then tracks 1, 2, 3, 5, 7, and 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.

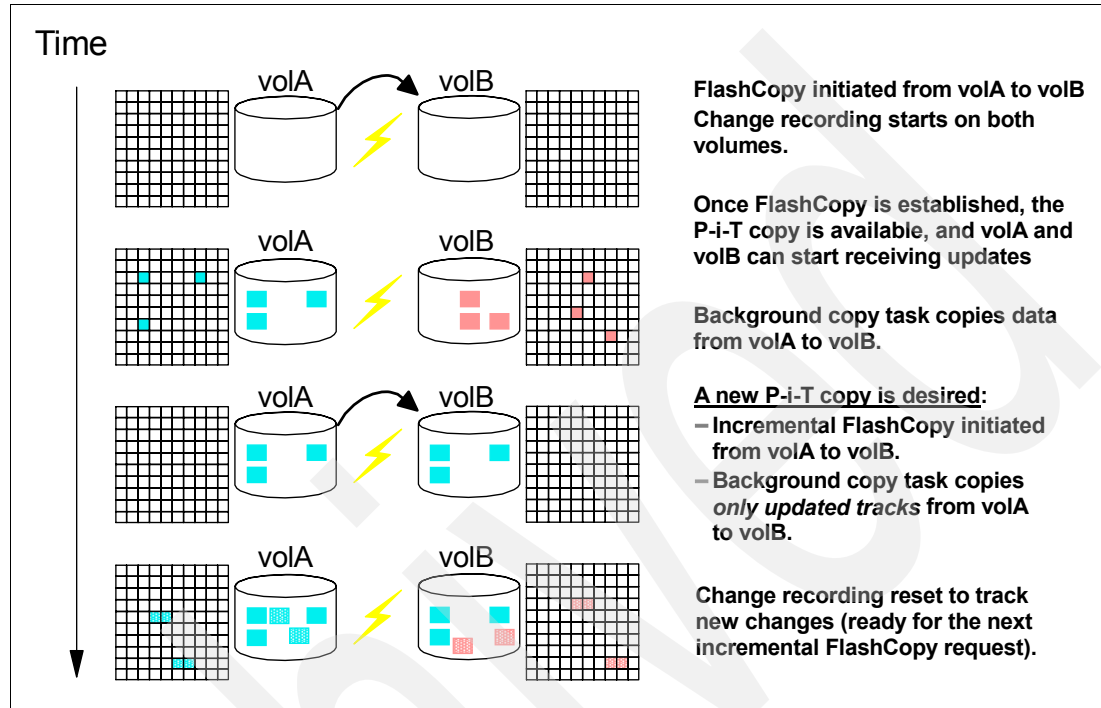


Figure C-4 Incremental FlashCopy

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

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 C-5 on page 220 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 entire 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 entire 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.

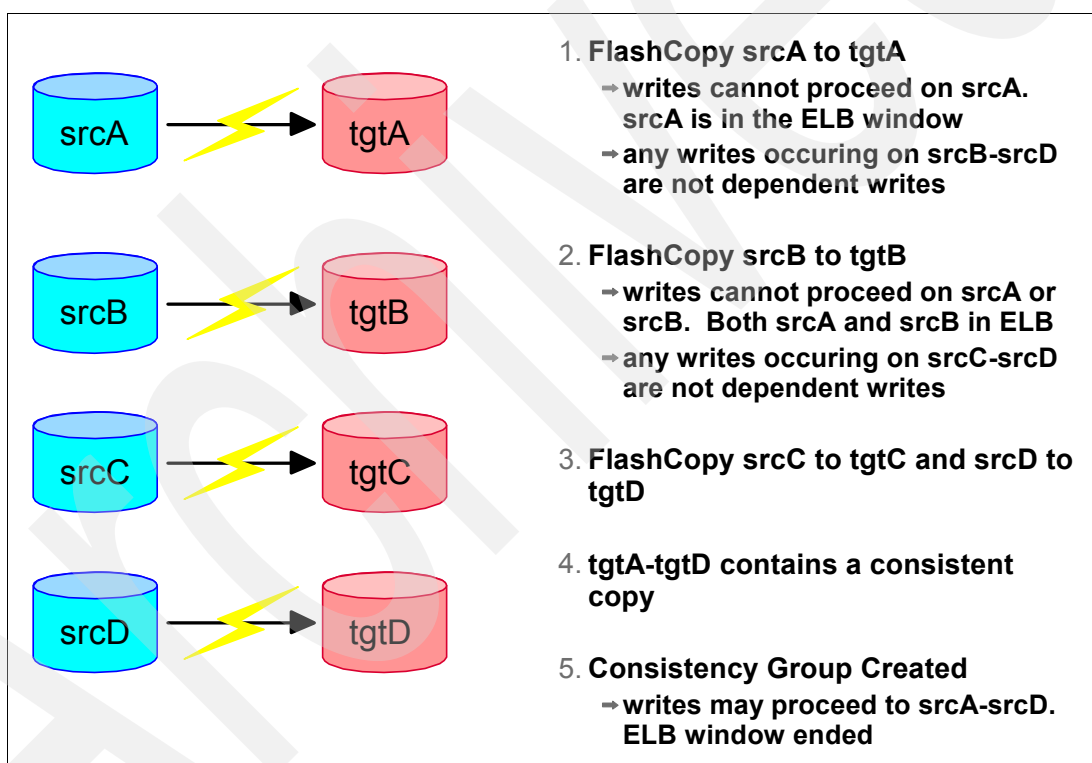


Figure C-5 FlashCopy consistency group

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 Storage Server Web Interface User's Guide*, SC26-7448.

DFSMS functions for DB2 point in time recovery

In this section we provide information about 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 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. Nondisruptive 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.

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

- ▶ *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370
- ▶ *DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, .. and More*, SG24-6079
- ▶ *DFSMSHsm Fast Replication Technical Guide*, SG24-7069

DFSMSHsm Fast Replication

DFSMSHsm was 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 later
- ▶ 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 DFSMSHsm commands support this new function, and other commands provide 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.

DFSMS copy pools

SMS was also enhanced in z/OS DFSMS V1.5 to support this function. The construct named *copy pool* and the *copy pool backup* storage group type were introduced.

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

The copy pool backup storage group type is 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 BCDS 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.

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: The following restrictions apply:

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

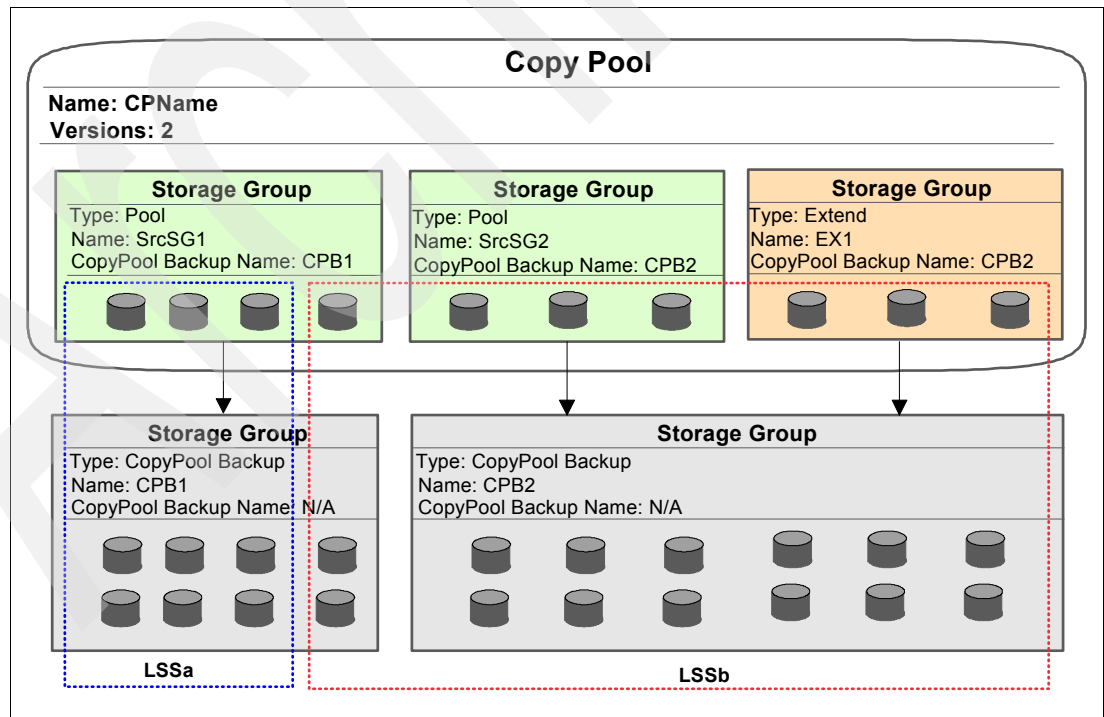


Figure C-6 Copy pool structure

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

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 from the Interactive Storage Management Facility (ISMF) panels 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, two versions of 10 backup target volumes each, or 20 backup volumes, would be allocated. This ensures that there are sufficient target volumes and will move the target volume selection process outside of the fast replication window. If you do not use the PREPARE keyword prior to taking backups, only copy pool backup volumes for the backup in progress will be allocated.

DB2 BACKUP and RESTORE functions in V8

DB2 for z/OS V8 introduced 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 that is stored by DFSMSHsm in its backup control data set (BCDS) records. The DB2 token breakout is shown in Table C-1.

Table C-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 C-4 shows how this looks from a DFSMSHsm LIST command output.

Example: C-4 DFSMSHsm LIST command output

```

F HSM,LIST CP(DSN$DB8A$DB) TERM
COPYPOOL= DSN$DB8A$DB VER=002, VALID=Y, VTOCENQ=N,
MADE ON 2008/03/03 AT 17:58:01
TKN(C)=C'DB8A & U+ - '
TKN(H)=X'C4C2F8C2BA5053E44E472848000002426090'
SGNAME=DB8A SOURCE=MHL125 TARGET=MHL213
SGNAME=DB8A SOURCE=MHL126 TARGET=MHL214

```

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

Table C-2 shows its breakout for the example.

Table C-2 DB2 token contents breakout of Example C-4 output listing

Field contents	Length	Value	Human readable format
DB2 SSID	4	C4C2F8C2	DB8A
TOD	8	BA5053E44E472848	Nov. 12th, 2003 at 22:58:01,939058

Field contents	Length	Value	Human readable format
RBA	6	000002426090	000002426090

NOVTOCENQ keyword

NOVTOCENQ is an optional parameter that DB2 uses to request DFSMSHsm to not serialize on the VTOCs of the volumes being processed. This is because DB2 has taken specific action to ensure that no data sets will be created, extended, or opened during the DFSMSHsm Fast Replication.

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 C-5 illustrates the changes.

Example: C-5 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  000000000000000000  BA245635C2B2  BA245635C2B2
2008/03/08

```

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.

DB2 PITR using Fast Replication

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

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 boot strap data set (BSDS) 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 that 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, and so on), deletion (drop table space, index, and so on), 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 that the backup is complete.

Figure C-7 illustrates what happen during the backup process.

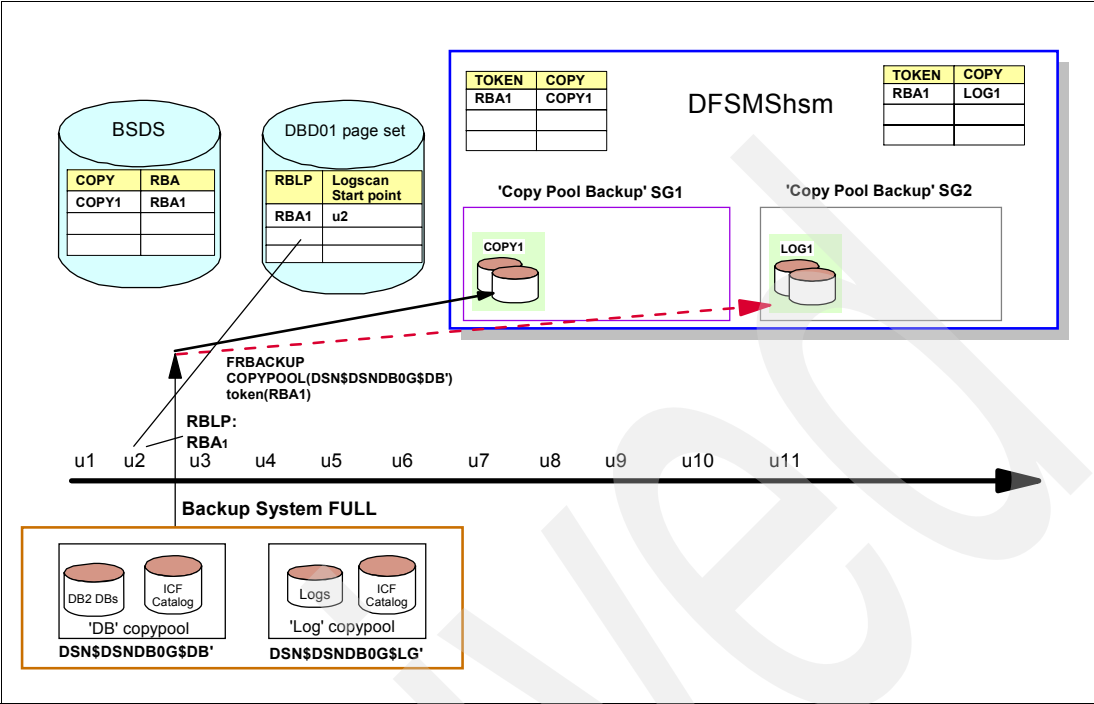


Figure C-7 BACKUP SYSTEM utility execution for DSNDB0G

One the 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 C-8 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 DSN\$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.

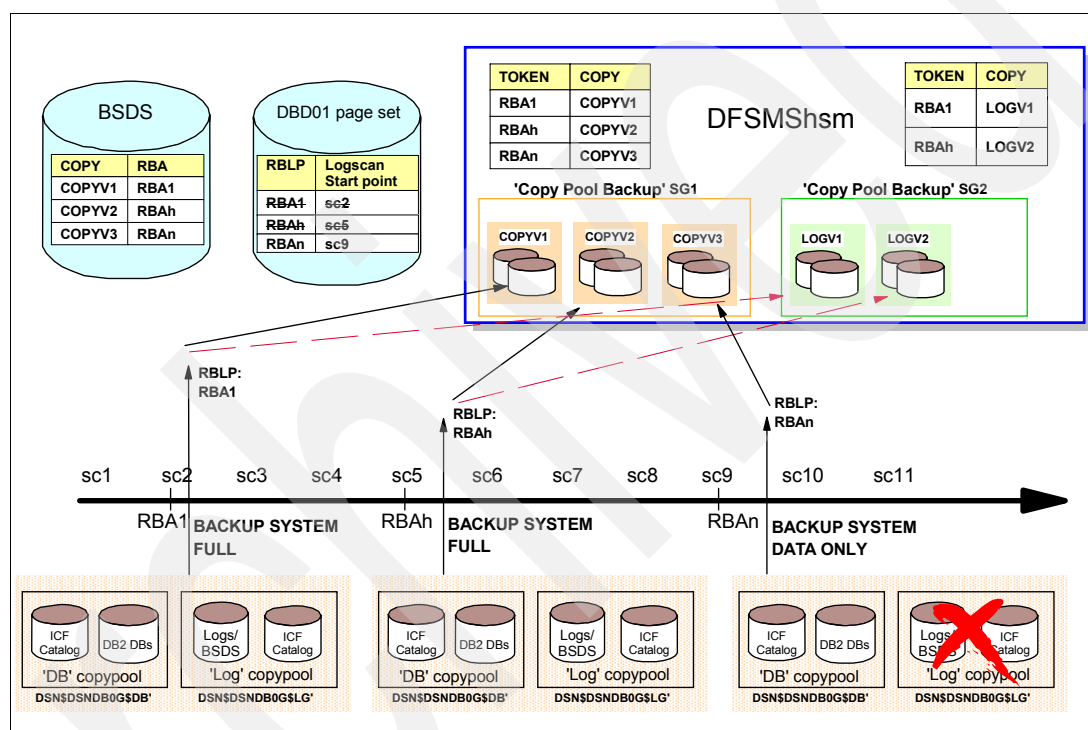


Figure C-8 Two BACKUP SYSTEM FULL, and one DATA ONLY are taken

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 C-6 shows an instance of full backup.

Example: C-6 Invoking DFSMSHsm replication with DB2 utility

```
//STEP1 EXEC DSNUPROC,SYSTEM=DB8A,UID=DIAG
//*      UTSTATS=' '
//SYSIN DD *
        BACKUP SYSTEM FULL
```

If you do not specify the FULL option in the Example C-6 on page 230, the FULL option is still implied, however, this is the default.

In Example C-7, the DATA ONLY option has been used. This will not copy the log copy pool.

Example: C-7 Invoking DFSMSHsm Fast Replication with DB2 utility (DATA ONLY option)

```
//STEP1 EXEC DSNUPROC,SYSTEM=DB8A,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 C-8 shows the database output.

Example: C-8 DFSMSHsm LIST command and output on DSN\$DB8A\$DB copy pool

F HSM,LIST CP(DSN\$DB8A\$DB) ODS(MHLRES3.DB8A.LIST)

-- DFSMSHsm CONTROL DATASET --COPY POOL--LISTING ----- AT 18:36:34 ON 08/03/03 FOR SYSTEM=SC65

COPYPOOL = DSN\$DB8A\$DB

VERSION	VALID	VTOCENQ	DATE	TIME
002	Y	N	2008/03/03	17:58:01

TOKEN(C)=C'DB8A[&U+â.ç...â-°'
 TOKEN(H)=X'C4C2F8C2BA5053E44E472848000002426090'

SGNAME	SOURCE - TARGET	SOURCE - TARGET	SOURCE - TARGET	SOURCE - TARGET
DB8A	MHL125 - MHL213	MHL126 - MHL214		

VERSION	VALID	VTOCENQ	DATE	TIME
001	Y	N	2008/03/03	17:46:29

TOKEN(C)=C'DB8A[&U+â.ç...â-°'
 TOKEN(H)=X'C4C2F8C2BA50514FB03D684A000002426090'

SGNAME	SOURCE - TARGET	SOURCE - TARGET	SOURCE - TARGET	SOURCE - TARGET
DB8A	MHL125 - MHL013	MHL126 - MHL014		

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

Example C-9 shows the log output.

Example: C-9 DFSMSHsm LIST command and output on DSN\$DB8A\$LG copy pool

F HSM,LIST CP(DSN\$DB8A\$LG) ODS(MHLRES3.DB8A.LIST)

-- DFSMSHsm CONTROL DATASET --COPY POOL--LISTING ----- AT 18:36:43 ON 08/03/03 FOR SYSTEM=SC65

COPYPOOL = DSN\$DB8A\$LG

VERSION	VALID	VTOCENQ	DATE	TIME
002	Y	N	2008/03/03	17:58:04

TOKEN(C)=C'DB8A[&U+â.ç...â-°'
 TOKEN(H)=X'C4C2F8C2BA5053E44E472848000002426090'

SGNAME	SOURCE - TARGET	SOURCE - TARGET	SOURCE - TARGET	SOURCE - TARGET
DB8ALOG1	MHL037 - MHL225			

DB8ALOG2 MHL147 - MHL226

VERSION VALID VTOCENQ DATE TIME
001 Y N 2008/03/03 17:46:32

TOKEN(C)=C'DB8A[&e|^\,Çç...â-°'

TOKEN(H)=X'C4C2F8C2BA50514FB03D684A000002426090'

SGNAME SOURCE - TARGET SOURCE - TARGET SOURCE - TARGET SOURCE - TARGET
DB8ALOG1 MHL037 - MHL025
DB8ALOG2 MHL147 - MHL026

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

Sample DB2 messages for DB2 BACKUP

In Example C-10 you see the output from an execution of the DB2 Backup utility specifying BACKUP SYSTEM FULL.

Example: C-10 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\$DB8A\$DB TOKEN = X'C4C2F8C2BA5053E44E472848000002426090'.
DSNU1614I	DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA COMPLETED SUCCESSFULLY, COPYPOOL = DSN\$DB8A\$DB TOKEN = X'C4C2F8C2BA5053E44E472848000002426090' ELAPSED TIME = 00:00:02.
DSNU1600I	DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS STARTING, COPYPOOL = DSN\$DB8A\$LG TOKEN = X'C4C2F8C2BA5053E44E472848000002426090'.
DSNU1614I	DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS COMPLETED SUCCESSFULLY, COPYPOOL = DSN\$DB8A\$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 C-11 shows the HSM output written to syslog when the FRBACKUP commands are executed during the DB2 backup execution.

Example: C-11 FRBACKUP output in syslog

ARC1801I FAST REPLICATION BACKUP IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN\$DB8A\$DB, AT 17:58:01
ARC1801I (CONT.) ON 2008/03/03
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION BACKUP OF COPY POOL
ARC1805I (CONT.) DSN\$DB8A\$DB
ARC1805I (CONT.) MHL125
ARC1805I (CONT.) MHL126
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN\$DB8A\$DB, AT 17:58:04
ARC1802I (CONT.) ON 2008/03/03, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR


```

ARC1801I (CONT.) COPY POOL DSN$DB8A$LG, AT 17:58:04
ARC1801I (CONT.) ON 2008/03/03
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION BACKUP OF COPY POOL
ARC1805I (CONT.) DSN$DB8A$LG
ARC1805I (CONT.) MHL037
ARC1805I (CONT.) MHL147
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8A$LG, AT 17:58:05
ARC1802I (CONT.) ON 2008/03/03, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000

```

Example C-12 shows the output written to the DFSMSHsm backup log data set from execution of the FRBACKUP command.

Example: C-12 FRBACKUP output in DFSMSHsm backup log data set

```

ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB8A$DB, AT 17:58:01 ON 2008/03/03
ARC0640I ARCFRTM - PAGE 0001 5695-DF175 DFSMSDSS V1R05.0 DATA SET SERVICES 2003.316 17:58
ARC0640I ARCFRTM - ADRO35I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC0640I ARCFRTM - PARALLEL
ARC0640I ARCFRTM - ADRI01I (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 - ADRI01I (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 - ADRI01I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - ADRI09I (R/I)-RI01 (01), 2003.316 17:58:03 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCFRTM - ADRO14I (SCH)-DSSU (02),
2003.316 17:58:03 ALL PREVIOUSLY SCHEDULED TASKS COMPLETED. PARALLEL MODE NOW IN EFFECT
ARC0640I ARCFRTM - ADRO50I (002)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADRO35I (002)-PRIME(86), INSTALLATION EXIT ALTERED DEBUG(FRMSG) OPTION TO UNSPECIFIED
ARC0640I ARCFRTM - ADRO16I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADRO50I (003)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION INTERFACE
ARC0640I ARCFRTM - ADRO35I (003)-PRIME(86), INSTALLATION EXIT ALTERED DEBUG(FRMSG) OPTION TO UNSPECIFIED
ARC0640I ARCFRTM - ADRO16I (003)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ARC0640I ARCFRTM - ADRO06I (002)-STEND(01), 2003.316 17:58:03 EXECUTION BEGINS
ARC0640I ARCFRTM - ADRI24I (002)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADRI80I (002)-TOMI (02), VOLUME MHL125 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADRO06I (002)-STEND(02), 2003.316 17:58:03 EXECUTION ENDS
ARC0640I ARCFRTM - ADRI01I (002)-CLTSK(01), 2003.316 17:58:03 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADRO06I (003)-STEND(01), 2003.316 17:58:03 EXECUTION BEGINS
ARC0640I ARCFRTM - ADRI24I (003)-DDTFP(01), TARGET VTOC BEGINNING AT 000003:0000 AND ENDING AT 000008:0014 IS OVERLAID
ARC0640I ARCFRTM - ADRI80I (003)-TOMI (02), VOLUME MHL126 WAS COPIED USING A FAST REPLICATION FUNCTION
ARC0640I ARCFRTM - ADRO06I (003)-STEND(02), 2003.316 17:58:03 EXECUTION ENDS
ARC0640I ARCFRTM - ADRI01I (003)-CLTSK(01), 2003.316 17:58:03 TASK COMPLETED WITH RETURN CODE 0000
ARC0640I ARCFRTM - ADRI01I (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$DB8A$DB
ARC1805I (CONT.) MHL125
ARC1805I (CONT.) MHL126
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB8A$DB, AT 17:58:04 ON 2008/03/03, FUNCTION RC=0000,
MAXIMUM VOLUME RC=0000
ARC1801I FAST REPLICATION BACKUP IS STARTING FOR COPY POOL DSN$DB8A$LG, AT 17:58:04 ON 2008/03/03
ARC0640I ARCFRTM - PAGE 0001 5695-DF175 DFSMSDSS V1R05.0 DATA SET SERVICES 2003.316 17:58
ARC0640I ARCFRTM - ADRO35I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC0640I ARCFRTM - PARALLEL
ARC0640I ARCFRTM - ADRI01I (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 - ADRI01I (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 - ADRI01I (R/I)-RI01 (01), TASKID 003 HAS BEEN ASSIGNED TO COMMAND 'COPY '
ARC0640I ARCFRTM - ADRI09I (R/I)-RI01 (01), 2003.316 17:58:05 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC0640I ARCFRTM - ADRO14I (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$DB8A$LG
ARC1805I (CONT.) MHL037
ARC1805I (CONT.) MHL147
ARC1802I FAST REPLICATION BACKUP HAS COMPLETED FOR COPY POOL DSN$DB8A$LG, AT 17:58:05 ON 2008/03/03, FUNCTION RC=0000,
MAXIMUM VOLUME RC=0000

```

DB2 Restore using Fast Replication backups

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

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, including the most current one (PK51979). 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 or the end of the log. 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 full system (including the logs) 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 that 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:

1. Stop DB2. If data sharing, stop all members.
2. 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.
3. If data sharing, delete all coupling facility structures.
4. Start DB2. If data sharing, start all members of the data sharing group.
5. DB2 will enter into system recover-pending mode, ACCESS(MAINT), and will bypass recovery except for indoubt URs.
6. Execute the RESTORE SYSTEM utility. The utility must be completed on the original submitting member.
7. Restores the most recent *database copy pool* version that was taken prior to the log truncation point.
8. Performs the log apply function.
9. 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 later.
 - Backs up data with another volume dump solution and uses SET LOG SUSPEND/RESUME.
 - Performs the log apply function only.
10. Stops DB2 to reset system recovery-pending status. If data sharing, stop all members.
11. Displays and terminates any active utilities.
12. Displays restricted objects and recovers objects in RECP status and rebuilds objects in RBDP status.

The SET LOG SUSPEND/RESUME commands can be used with other volume backup solutions if BACKUP SYSTEM is not 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 than 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 LOGONLY is possible.

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

Figure C-9 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.

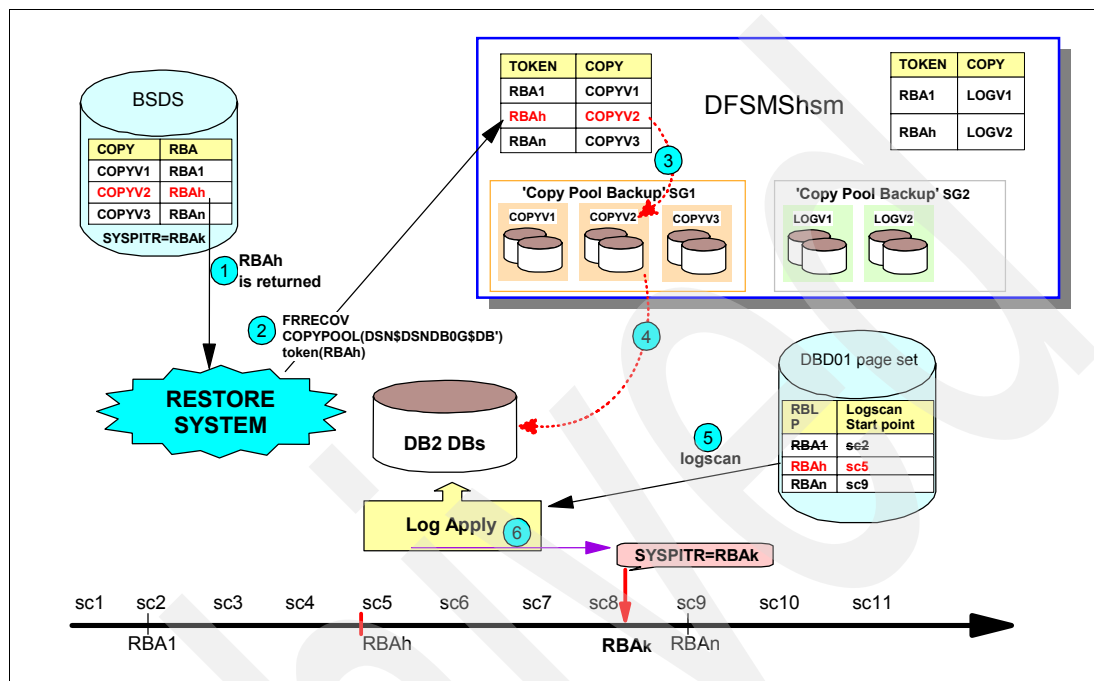


Figure C-9 Recovering to an arbitrary point in time using the RESTORE SYSTEM utility

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 C-13 shows the JCL and control statements required for the RESTORE SYSTEM.

Example: C-13 Sample JCL of RESTORE SYSTEM utility

```
//STEP1 EXEC DSNUPROC,SYSTEM=DB8A,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 C-14.

Example: C-14 Sample JCL of RESTORE SYSTEM with LOGONLY option

```
//STEP1 EXEC DSNUPROC,SYSTEM=DB8A,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:

1. Stop DB2.
2. 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.
3. Start DB2.

After running RESTORE SYSTEM

Complete the following steps after running the RESTORE SYSTEM utility:

1. Stop DB2 to reset system RECOVER-pending status.
2. Use the DISPLAY UTILITY command to see whether any utilities are running. If other utilities are running, use the TERM UTILITY command to end them.
3. 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 *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 C-15 shows an example of the output that you should see from a successful execution of the RESTORE SYSTEM utility.

Example: C-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'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 C-16.

Example: C-16 FRRECOVER messages in DFSMSHsm log

```
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR
ARC1801I (CONT.) COPY POOL DSN$DB8A$DB, AT 16:38:50
ARC1801I (CONT.) ON 2008/03/11
ARC1805I THE FOLLOWING 00002 VOLUME(S) WERE
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST
ARC1805I (CONT.) REPLICATION RECOVERY OF COPY POOL
ARC1805I (CONT.) DSN$DB8A$DB
ARC1805I (CONT.) MHL125
ARC1805I (CONT.) MHL126
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR
ARC1802I (CONT.) COPY POOL DSN$DB8A$DB, AT 16:38:51
ARC1802I (CONT.) ON 2008/03/11, FUNCTION RC=0000,
ARC1802I (CONT.) MAXIMUM VOLUME RC=0000
```

Several sample scenarios are explained in 5.1, “System level backup using DB2 and Recovery Expert GUI to perform the restore” on page 84.

BACKUP and RESTORE enhancements with DB2 9 for z/OS

As we have seen in “DB2 BACKUP and RESTORE functions in V8” on page 224, BACKUP and RESTORE SYSTEM utilities were added in DB2 V8 and use disk volume FlashCopy backups and copy pool z/OS DFSMSHsm V1R5 constructs. In DB2 9 for z/OS these utilities are enhanced to use new functions available with z/OS V1R8 DFSMSHsm:

- ▶ **Object-level recovery**
This enhancement allows *individual table spaces or index spaces* to be recovered (the V8 support was for the entire system).
- ▶ **Dump to tape**
Maintaining multiple copies of all the data on disk can be expensive. DB2 9 for z/OS also allows for the backup to be implemented directly to tape.
- ▶ **Incremental FlashCopy**
The physical copying to disk in the background can be improved by the use of incremental FlashCopy.

Note that, even if incremental FlashCopy is used, the dump to tape is always a full dump. Incremental FlashCopy has no benefit when dumping to tape except that the dumping to tape might begin earlier because less data needs to be copied to disk before writing to tape can be started.

Object-level recovery

The backups taken by the BACKUP SYSTEM utility (referred to here as system level backups) in V8 are used for recovery of the entire DB2 system. DB2 9 for z/OS has an enhancement to allow a DB2 object to be recovered from the system level backups. Recovery is via the RECOVER utility, which is now capable of using system level backups for the restore phase in addition to image copies. In order to allow RECOVER to consider using system level backups there is a DSNZPARM option that can be set from panel DSNTIP6. This is the first option on the panel, as shown in Figure C-10 on page 240. Having enabled the use of system level backups, they will be automatically considered in object level recoveries.

Note: You need to alter your indexes to the COPY YES attribute to enable RECOVER from system level backups.

Unless the installation is also planning to offload the system level volume backups to tape (see “Tape support for BACKUP SYSTEM” on page 241), there are no syntax changes to RECOVER. The user will specify the table spaces or indexes in a list, individually, or via a LISTDEF, and the RECOVER utility will determine the most recent recovery base. In addition to the image copies recorded in the SYSCOPY (or the log for some catalog or directory spaces), the system level volume backups are also examined.

If the last copy before the recovery point is a system level volume backup, then the specific data sets will be extracted and restored via DFSMSHsm.

Note that if RECOVER cannot successfully restore from this system level backup, then the utility will terminate, assuming that the OPTION EVENT(ITEMERROR,SKIP) was not specified. This is in contrast to RECOVER having a problem locating an image copy on tape or disk, when an alternative such as the *local backup* or an earlier image copy is tried. If the system level volume backup is not usable for any reason, then the user can bypass it by using the RESTOREBEFORE option on RECOVER.

Restriction: If a data set has moved to a different volume or was deleted between the time of the BACKUP SYSTEM utility and the time of the RECOVER utility, then object level recovery is not possible unless an image copy is available.

This restriction does not apply to system level backups created using DB2 Recovery Expert for z/OS because the tool maintains this information in one of the SBRS repository data sets for using at the time of recovery.

Allowing system level volume backups to be used by RECOVER means that conventional image copies need be taken on a potentially much-reduced frequency. For example, if there is a system backup daily, then running image copies daily may not be needed, unless for special reasons like DR. Due to the restriction noted above, we do not recommend that image copies are dispensed with entirely. Image copies will still be required following LOAD REPLACE and REORG LOG NO, as enforced by DB2.

Note: Running BACKUP SYSTEM will now result in Real Time Statistics columns for COPY to be updated because system level backups may now be used in object-level recovery.

The installation panel DSNTIP6 is shown in Figure C-10.

Option 1 determines whether the system level backup and restore functions should be active. We discuss the next three options in the next sections.

```
DSNTIP6          INSTALL DB2 - DB2 UTILITIES PARAMETERS
===>

Enter system-level backup options for RESTORE SYSTEM and RECOVER below:
1 SYSTEM-LEVEL BACKUPS ===> NO          As a recovery base: NO or YES
2 RESTORE/RECOVER      ===> NO          From dump: NO or YES
3 DUMP CLASS NAME      ===>             For RESTORE/RECOVER from dump
4 MAXIMUM TAPE UNITS   ===> NOLIMIT     For RESTORE SYSTEM: NOLIMIT or 1-255

Enter other DB2 Utilities options below:
5 TEMP DS UNIT NAME    ===> SYSDA       Device for temporary utility data sets
6 UTILITY CACHE OPTION ===> NO          3990 storage for DB2 utility IO
7 STATISTICS HISTORY   ===> NONE        Default for collection of stats history
8 STATISTICS ROLLUP     ===> NO          Allow statistics aggregation: NO or YES
9 STATISTICS CLUSTERING===> ENHANCED    For RUNSTATS (ENHANCED or STANDARD)
10 UTILITY TIMEOUT      ===> 6           Utility wait time multiplier
11 UT SORT DS ALLOCATION===> NO          Predictable sort disk space allocation
12 IGNORE SORTNUM STMT  ===> NO          Ignore SORTNUM keyword in UT stmt

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure C-10 Panel: DSNTIP6

Tape support for BACKUP SYSTEM

The BACKUP SYSTEM utility has been enhanced to allow a FlashCopy to go to tape. See Figure C-11 for the syntax diagram with the new options DUMP and DUMPONLY of the BACKUP SYSTEM utility control statement.

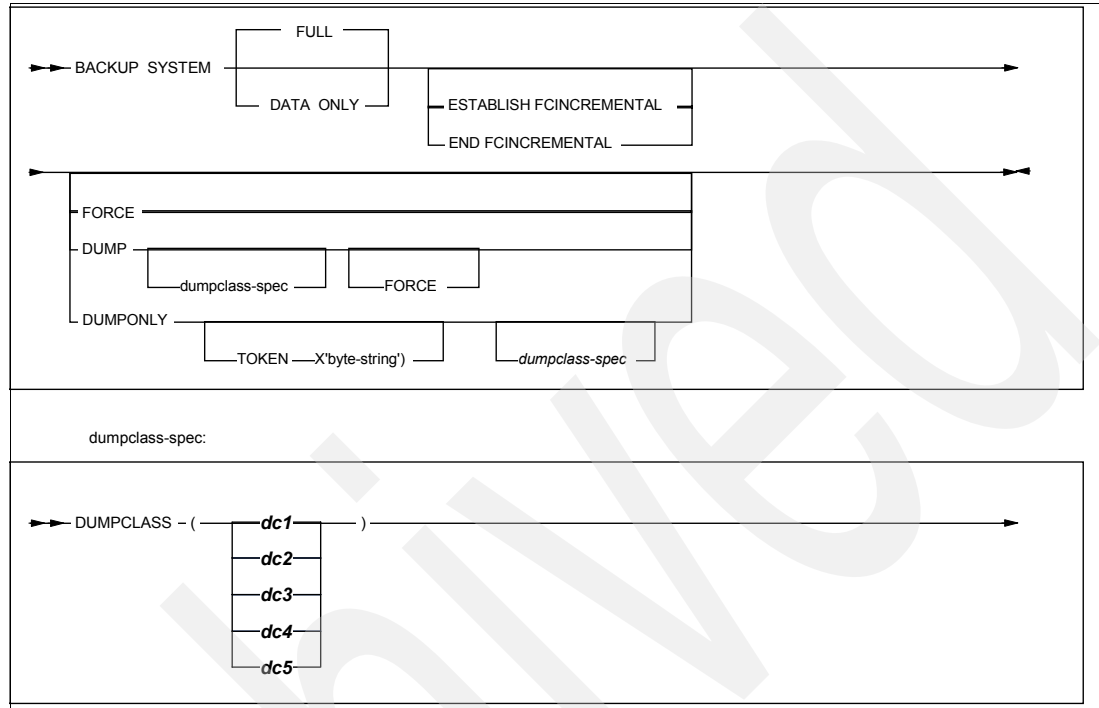


Figure C-11 BACKUP SYSTEM syntax

DUMP indicates that you want to create a fast replication copy of the database copy pool and the log copy pool on disk and then initiates a dump to tape of the fast replication copy. The dump to tape begins after DB2 successfully establishes relationships for the fast replication copy.

DUMPONLY indicates that you want to create a dump on tape of an existing fast replication copy (that is currently residing on the disk) of the database copy pool and the log copy pool. You can also use this option to resume a dump process that has failed.

The output of the DUMP or DUMPONLY is directed to what are called DFSMShsm *dump classes*. Each dump class specifies the unit types to which the data will be directed.

This DB2 enhancement was designed to allow for the data to go to tape. However, an SMS dump class is not restricted to a tape medium.

When the data class is tape, you can specify *in SMS* to what degree volumes will be stacked on tapes. There will be a trade-off in tape capacity usage and performance to be considered since the more volumes are stacked on a tape, the longer it will take to search.

DB2 allows up to five dump classes to be specified. Each specification results in a complete copy of the copy pools being backed up being directed to each dump class. Any specification in the utility control cards of dump classes will override the dump classes defined in the COPYPOOL definitions.

By having multiple dump classes, it could be possible, for example, to send one copy to remote devices for DR purposes, while keeping another copy for local operational reasons.

Important: Having data on tape may be advantageous for moving the data off site for DR situations. It may also allow for longer retention of data locally. It does not, however, compare with the speed of a FlashCopy during restore. Full system restores from tape will be lengthy processes.

Although the logical FlashCopy is extremely fast, the physical copying of data does take time and can lead to a significant load on the I/O subsystem. See , “Incremental FlashCopy” on page 243, for one way of reducing I/O load during BACKUP SYSTEM.

The choice of DUMP or DUMPONLY is another way to reduce impact when offloading to tape. When the DUMP option is selected, the copying to tape is started ahead of the physical completion of the FlashCopy. This means that for a period of time there is additional load on the I/O subsystem (that is, the disk-to-disk copy associated with the FlashCopy, in addition to the disk-to-tape copy). The DUMPONLY option can instead be used to copy a previous backup to tape at a later time, thereby not causing such a big peak in I/O activity.

Tip: To minimize I/O activity peaks, perform a BACKUP SYSTEM without DUMP and follow it with a BACKUP SYSTEM with DUMPONLY.

Since the copying to tape is relatively slow, a new FORCE keyword has been added to BACKUP SYSTEM. This allows a new backup to be started even though a previous DUMP has not finished. FORCE should only be used if it is more important to take a new backup than to let the offloading to tape of the previous dump to tape complete.

Note: The LIST COPYPOOL with DUMPVOLS option can be used to verify the status of DUMP and DUMPONLY.

There is a *no copy* option set in SMS that allows for copies to go to a dump class, without the FlashCopy versions going to disk. COPY NO is set on the SMS COPYPOOL PANEL. It is the *number of recoverable disk versions* that is set to zero for COPY NO. Note that this option could never provide for quick restores, and the backup could be very lengthy as well.

The RESTORE SYSTEM utility has been improved to use the system level backups that have been dumped to tape. Note that RESTORE only handles the *database* COPYPOOL and then applies the log records from the active logs. If you want to restore the log copy pool from your BACKUP SYSTEM FULL, you need to restore it natively.

The panel in Figure C-10 on page 240 shows three parameters to control the RESTORE. All of these can be overridden by the utility control statements.

Option 2 determines whether the FlashCopy version from a dump class is to be used rather than one on disk. This parameter applies also to RECOVER. The default NO may be preferred by customers who do not want to use the potentially slower recovery method.

Option 3 determines the default dump class to restore from. This also applies to RECOVER.

Option 4 determines the maximum number of tape units to be used by RESTORE. The default is not to limit the number of drives, but this can be capped at anything from 1 to 255. It can be overridden by TAPEUNITS keyword in the RESTORE SYSTEM utility statement.

Note: The reason for option 2 is to allow the tape version to be specified, rather than the significantly faster disk FlashCopy version, to support disaster recovery. The tapes could be sent to the disaster recovery location and registered in HSM.

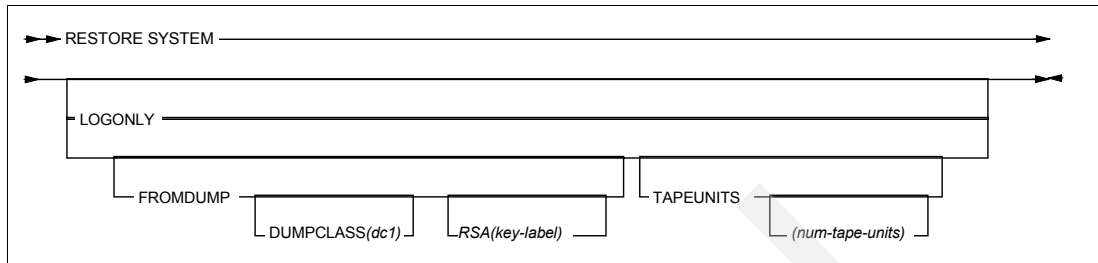


Figure C-12 *RESTORE SYSTEM* syntax

In summary, while tape backups can be useful, the time to restore from such backups should never be underestimated. It is expected that installations with high-availability requirements will restore via disk FlashCopy in all but catastrophic situations.

Incremental FlashCopy

In order to reduce I/O activity, support is being added to allow incremental FlashCopy. In addition to z/OS Version 1 Release 8 DFSMSHsm, this function requires APAR OA17314 for z/OS. Note that incremental FlashCopy does not reduce the need for disk volumes, unlike an incremental image copy potentially does.

In order for FlashCopy incrementals to be used, the relationship must first be established. This can be done either by HSM commands or by using the new options on the BACKUP SYSTEM utility. See Figure C-11 on page 241. ESTABLISH FCINCREMENTAL and END FCINCREMENTAL are added to specify a persistent or last incremental FlashCopy relationship is to be established. Similarly, they can be ended via an HSM command or the utility. All tracks on the source volume are considered to be changed when the relationship is established, so all tracks are copied. Subsequent incremental FlashCopies will copy only the tracks that have changed on the source volume since the last copy was taken. The previous content on the volumes will be replaced by the current contents.

Do not confuse the adjective *incremental* as used with image copies with the one used with FlashCopy. With COPY you can merge incremental and previous copies. With FlashCopy the incremental changes are immediately applied to the previous full copy.

Note: A disk volume can have only one incremental relationship. This means that if a copy pool has more than one disk generation, then the other generations will be full backups.

DFSMSHsm Fast Replication associates a defined number of versions to a copy pool. This specifies the maximum number of versions that can be maintained in parallel. Each invocation of the FRBACKUP command creates a new version in the copy pool (with version numbers assigned automatically in the range of 1 to 999). If more versions are created than the copy pool can support in parallel, then the oldest version will be deleted and replaced by the newest version. This is sometimes referred to as *roll-off* processing.

Since version numbers are always increased (with the maximum being 999 and then restarting at 1), a relative addressing of the versions in the copy pool is more appropriately provided by generations. They follow the same concept as Generation Data Groups (GDGs). Generation 0 always refers to the most current version in the copy pool. The second most current version is referred to as generation 1, and so forth.

Figure C-13 shows the different times to take the copies. V1 is a full backup. V2 is the first incremental, and therefore all pages are copied. This takes as long as the full backup. V3 is an incremental, and this is much quicker. Similarly, V4 and V5 are incrementals with a shorter duration than a full backup. The backup that withdraws the relationship is still an incremental. T6 and T7 are full backups.

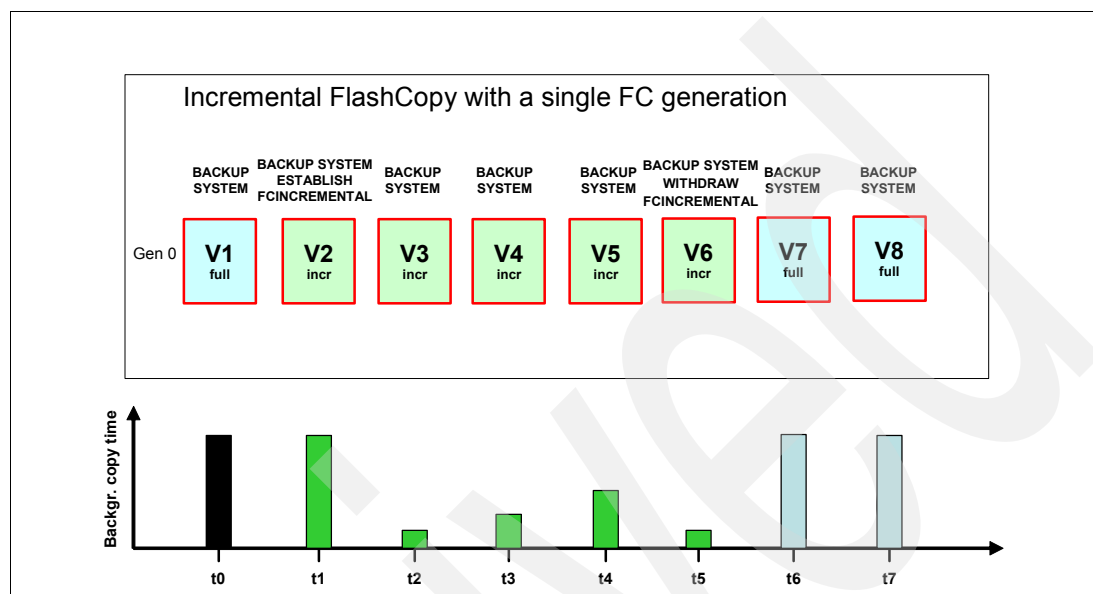


Figure C-13 Incremental FlashCopy with single version

Figure C-14 shows what happens when the copy pool is set up to have two versions. So for each source volume we have two sets of target volumes that we can copy to. However, we cannot be using incremental FlashCopy to both of them, as we are restricted to a single incremental relationship per volume.

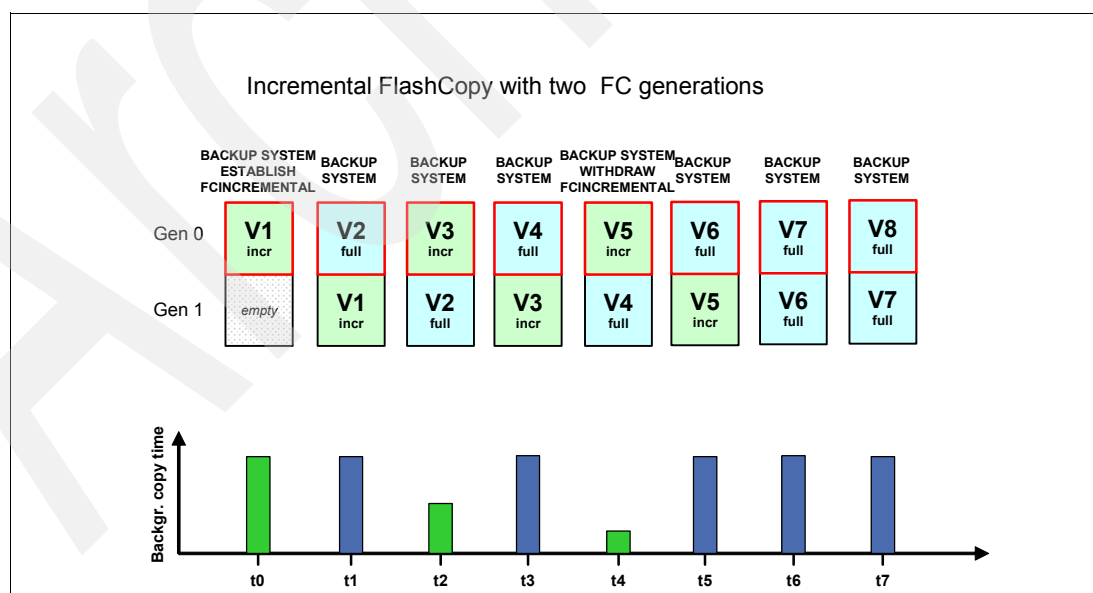


Figure C-14 Incremental FlashCopy with two versions

V1 is the backup that establishes the incremental, so it copies all the data. V2 cannot be an incremental, as it is going to the other set of volumes and is a full backup. V3 will be an incremental, and hence shows a shorter time. The next backup V4 will be a full backup. Every

other backup is a full in this setup, alternating with the incremental. So V5 will be a incremental. However, it withdraws the relationship, and all subsequent backups are full.

Clearly, the incremental reduces the copy time. It also reduces the workload on the I/O subsystem. When there are multiple versions allowed for a copy pool, the benefit diminishes. If we had five volumes, then only 20% of backups could be incremental. In a two-generation example it maybe possible to take two backups a day—one in the quiet time of the day (the full) and one at a busier time of the day (incremental).

RECOVER utility enhancements for point-in-time recovery

Performance can be considerably improved by enabling the fast log apply function on the DB2 subsystem. The RECOVER utility automatically uses the fast log apply process during the LOGAPPLY phase if the fast log apply function has been enabled. When the recovery completes, all storage used by the fast log apply function is returned to the DBM1 address space. Both copies of the log can be used, and the buffer default for DB2 9 for z/OS has been increased from 100 MB to 500 MB. However, we recommend that you do *not* increase the number of concurrent recovery jobs per member.

The RECOVER utility recovers data to the current state or to a previous point in time by restoring a copy and then applying log records. In the versions previous to DB2 9 for z/OS, a point-in-time recovery could cause a data inconsistency problem since the point recovered to was not a consistent point. This is because there was no process to back out in-flight units of recovery (UR). As a result, it was recommended to take QUIESCE points to eliminate this problem. However, running the QUIESCE utility yielded the following problems:

- ▶ It reduced access to read-only on the objects that are being quiesced. This created an immediate impedance to applications that were running on high-volume systems.
- ▶ Frequently, running this utility produced unwanted overhead on production systems.

In reality, the executions of many point-in-time recoveries are performed to unplanned points in time. This introduces manual intervention since the data that is left in an inconsistent state must be repaired. This error-prone method proves to be quite time consuming and requires a much deeper knowledge of DB2.

DB2 9 for z/OS takes considerable measures to reduce the time that is required for manual interventions to perform such an operation. The following steps are improved during the RECOVER to a point in time:

1. Automatically detect the uncommitted transactions that are running at the point-in-time recovery point.
2. Roll back changes on the object to be recovered to ensure data consistency after the point-in-time recovery. No fast log apply function is used. Here are the particulars:
 - Changes made on the recovered objects by URs that are INFLIGHT, INABORT, or POSTPONED ABORT during the recovery time are rolled back.
 - URs that are INDOUBT during the recovery point are treated as INABORT, and their changes on the recovered objects are also rolled back.
 - INCOMMIT URs are treated as committed, and no rollback occurs.
 - If a UR changes multiple objects in its life span:
 - Only changes made by it on the objects that are being recovered are rolled back.
 - Changes made by it on other objects are not rolled back.
3. Leave the recovered objects in a consistent state from a transaction point of view.

Note: Consistency is not guaranteed when recovering to a specified image copy (TOCOPY, TOLASTCOPY, and TOLASTFULLCOPY of RECOVER options).

DB2 objects can now be recovered to *any* previous point in time with full data integrity. In addition, these improvements allow you to avoid running the QUIESCE utility regularly, so you can reduce the disruption to DB2 users and applications. Recovery to a point in time with consistency is available only in new-function mode to avoid issues with coexistence environments.

Important: You must include all associated objects in the same RECOVER utility job to ensure data consistency from the application point of view. If the object or objects are not specified in the same list, the utility sets the appropriate prohibitive pending state. For example, if the parent table space is recovered to a point in time, but its dependent table space is not recovered in the same list, the dependent table space is placed in a CHKP (check pending) state.

The RECOVER utility automatically uses the fast log apply process during the LOGAPPLY phase if the fast log apply function has been enabled. After the recovery completes, all storage that is used by the fast log apply function is returned to the DBM1 address space.

In addition, the option RESTOREBEFORE has been added to the RECOVER utility. This option allows you to search for an image copy, concurrent copy, or system level backup with an relative byte address (RBA) or log record sequence number (LRSN) value earlier than the specified X'byte-string' value. We recommend that you exercise the use of this feature. For example, if you know that you have a broken image copy, you can direct the RECOVER utility to a previous full or incremental copy and perform a LOGAPPLY from that point onward.

Contents of schema level repository

In this appendix, we provide an overview of the schema level repository (SLR), a fundamental component of the Recovery Expert tool, its contents, and its management.

While no function covered in this book requires the SLR, most functions described in the companion book *IBM DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226, do.

We have added this appendix to show the small changes that have occurred to SLR since V1.

Overview

The schema level repository is a set of (currently) 53 DB2 tables (as shown in Table D-1 on page 251) that serves as an archive to hold object definitions and alterations to object definitions. The default database name for the SLR is SYSTOOLS and the default schema name is SYSTOOLS. You can customize these defaults. You have to run the ARYDDLA, ARYDDL8, ARYDDL81, and ARYDDL82 members in the ARY.V2R1M0.SARYSAMP library to define the SLR objects in the appropriate DB2 subsystem.

During the installation, the SLR is initially loaded with the current data extracted from the DB2 catalog tables by the ARYSJ002 job in the ARY.V2R1M0.SARYSAMP library. The SLR is critical to the recovery of dropped objects and point-in-time recovery of existing objects, and needs to be periodically updated with the relevant catalog information. Data Definition Language (DDL) changes over a period of time are needed when a recovery action is requested to a point in time such as the image copy that was taken for an earlier DDL version of the object.

Recovery of dropped objects is not possible if the SLR does not contain the object definitions, the alterations to object definitions (including the fact that the object was dropped), and the image copy information. You need to schedule updates of the SLR to ensure proper recovery of dropped objects. Ideally, update the SLR whenever object definitions are created or changed, and when image copies and quiesce points are taken.

Important: Schedule an SLR update every few hours or daily. If you perform important application object definition updates at scheduled intervals, then update the SLR immediately after the object definition updates.

If the SLR does not contain object definitions, alterations to object definitions, and image copy information in it, then the recovery of the undropped object to a point in time might not synchronize the data and the DDL at the required moment in time.

When you update the SLR, it creates a row in one of the SLR tables with a creation time that corresponds to the creation time of the object. This entry in the SLR has a creation time and a blank value for the end time of this object definition level. When you alter this object, such as adding a column and running the SLR update again, a new entry is created for this object in the SLR with a new creation time corresponding to when the SLR update is performed (not the exact time when the object is created) and a blank value for the end time of this object definition level. The previous entry (which now represents the earlier object definition level of this object) is now given an end time that is one microsecond less than the creation time of the new entry. Figure D-1 on page 249 shows an example of multiple object definition levels for an object as recorded in the SLR.

Note: Establish each DB2 subsystem in the SLR by tailoring the ARYSJ001 for that subsystem or data sharing group and running it.

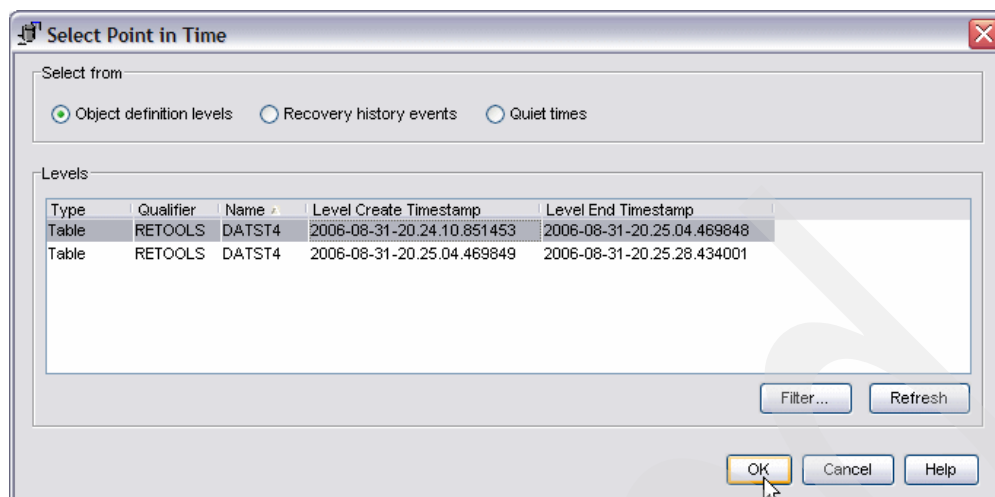


Figure D-1 Object definition level entries in the SLR

Recovery Expert provides a sample library member ARYSJ002, which contains the job control language (JCL) to update the SLR. The initial run of job ARYSJ002 might take multiple hours to copy the contents of the DB2 system catalog. Runtimes vary depending on the DB2 system catalog size.

To update the schema level repository:

1. Edit the ARY.V2R1M0.SARYSAMP member ARYSJ002.
2. Add the appropriate job card to ARYSJ002.
3. Change #SARYLOAD to the appropriate IBM DB2 Recovery Expert for Z/OS product libraries.
4. Change #SDSNLOAD to the appropriate DB2 load library.
5. Change #SSID to the target DB2 subsystem ID.
6. Change the DB2PARMS data set name to the product Virtual Storage Access Method (VSAM) control data set that is created by the sample JCL provided in member ARYSJ000.
7. Modify the CAPTURE PROFILES statement as instructed in the sample member ARYSJ002. This step is dependent of whether DB2 Automation Tool is installed.
8. Run ARYSJ002.

Attention: The SLR update job must end with a return code of zero. If a non-zero return code occurs, review the job output for errors. Correct the problem and resubmit the JCL.

The Recovery Expert GUI client options tab (Figure D-2) must identify the SLR created and updated previously, to store the quiet times.

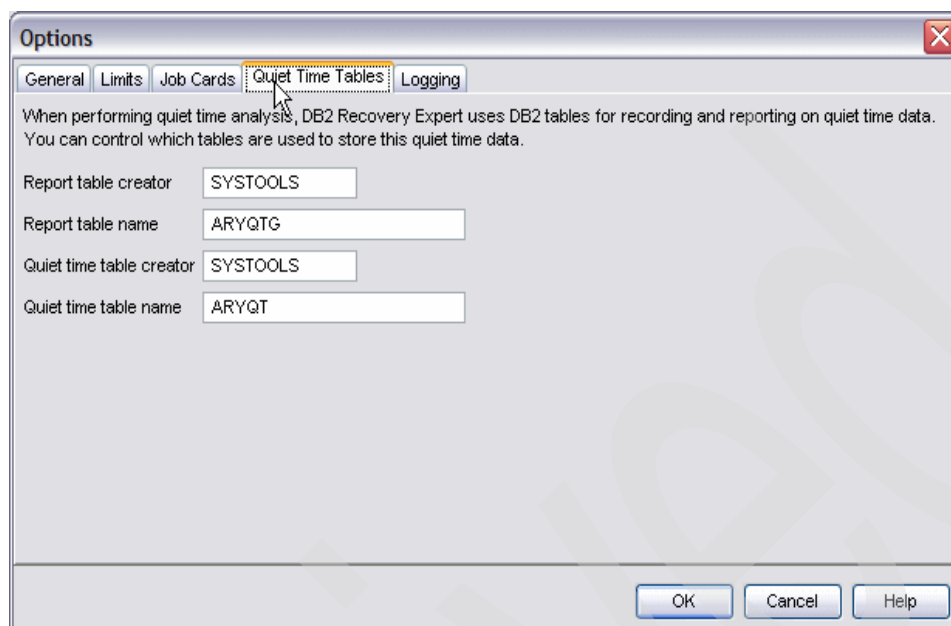


Figure D-2 SLR table details for quiet times

The quiet times are then displayed in the quiet time window, as shown in Figure D-3.

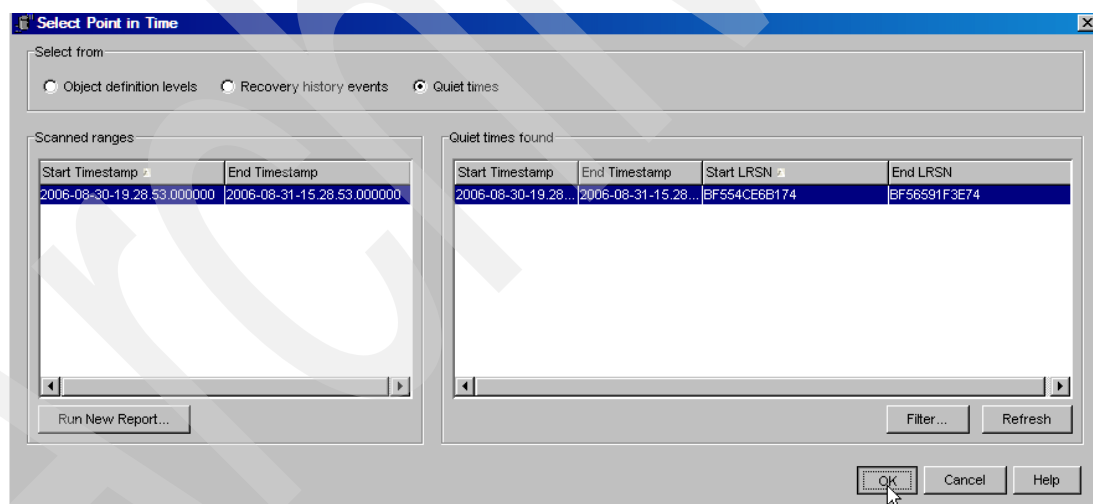


Figure D-3 Quiet times

SLR tables

Table D-1 on page 251 shows the list of SLR tables in the SYSTOOLS (default) database, and the corresponding DB2 catalog tables from which it populates these tables. A brief description of the SLR tables is included. The SLR tables have a number of indexes and views on them that are not shown here. The SLR tables also have additional columns as compared to their corresponding DB2 catalog tables.

Not all DB2 catalog tables have an equivalent SLR table, and not all SLR tables have an equivalent DB2 catalog table. Ten SLR tables have no equivalent DB2 catalog tables because they contain information about the execution of the Recovery Expert for z/OS environment. These tables are ARYQT, ARYQTG, ARYSPEC_DATA, ARYSPEC_DIR, ARYVIEWSR, ARYVRLOG, ARYPROFILES, ARYOBJECTS, DR_ARCHIVE_LOGS00, and DR_IMAGE_COPY_CATL.

Table D-1 SLR tables in comparison with DB2 catalog tables

SLR tables (SYSTOOLS)	Corresponding DB2 tables (SYSIBM)	Description
ARYAUXRELS	SYSAUXRELS	Contains one row for each auxiliary table created for a large object (LOB) column. A base table space that is partitioned must have one auxiliary table for each partition of each LOB column.
ARYCHECKS	SYSCHECKS	Contains one row for each check constraint.
ARYCHECKS2	SYSCHECKS2	Contains one row for each table check constraint for catalog tables created in or after Version 7. Check that constraints for catalog tables created before Version 7 are not included in this table.
ARYCOLAUTH	SYSCOLAUTH	Records the UPDATE or REFERENCES privileges that are held by users on individual columns of a table or view.
ARYCOLUMNS	SYSCOLUMNS	Contains one row for every column of each table and view.
ARYCOPY	SYSCOPY	Contains information required for recovery.
ARYDATABASE	SYSDATABASE	Contains one row for each database, except for database DSNDDB01.
ARYDATATYPES	SYSDATATYPES	Contains one row for each distinct type defined to the system.
ARYDBAUTH	SYSDBAUTH	Records the privileges that are held by users over databases.
ARYDBRM	SYSDBRM	Contains one row for each DBRM of each application plan.
ARYFIELDS	SYSFIELDS	Contains one row for every column that has a field procedure.
ARYFOREIGNKEYS	SYSFOREIGNKEYS	Contains one row for every column of every foreign key.
ARYINDEXES	SYSINDEXES	Contains one row for every index.
ARYINDEXPART	SYSINDEXPART	Contains one row for each non-partitioned secondary index (NPSI) and one row for each partition of a partitioning index or a data-partitioned secondary index (DPSI).
ARYKEYS	SYSKEYS	Contains one row for each column of an index key.
ARYOBJECTS	No equivalent	Is used to archive DB2 Automation Tool object profiles.
ARYPACKAGE	SYSPACKAGE	Contains a row for every package.
ARYPACKAUTH	SYSPACKAUTH	Records the privileges that are held by users over packages.
ARYPACKDEP	SYSPACKDEP	Records the dependencies of packages on local tables, views, synonyms, table spaces, indexes, aliases, functions, and stored procedures.
ARYPACKLIST	SYSPACKLIST	Contains one or more rows for every local application plan bound with a package list. Each row represents a unique entry in the plan's package list.

SLR tables (SYSTOOLS)	Corresponding DB2 tables (SYSIBM)	Description
ARYPARMS	SYSPARMS	Contains a row for each parameter of a routine or multiple rows for table parameters (one for each column of the table).
ARYPKSYSTEM	SYSPKSYSTEM	Contains zero or more rows for every package. Each row for a given package represents one or more connections to an environment in which the package can be run.
ARYPLAN	SYSPLAN	Contains one row for each application plan.
ARYPLANAUTH	SYSPLANAUTH	Records the privileges that are held by users over application plans.
ARYPLANDEP	SYSPLANDEP	Records the dependencies of plans on tables, views, aliases, synonyms, table spaces, indexes, functions, and stored procedures.
ARYPLSYSTEM	SYSPLSYSTEM	Contains zero or more rows for every plan. Each row for a given plan represents one or more connections to an environment in which the plan can be used.
ARYPROFILES	No equivalent	Is used to archive DB2 Automation Tool object profiles.
ARYQT	No equivalent	Contains the quiet times detected after running log analysis.
ARYQTG	No equivalent	Contains the group ID that relates multiple quiet times that are detected after running log analysis.
ARYRELS	SYSRELS	Contains one row for every referential constraint.
ARYRESAUTH	SYSRESAUTH	Records CREATE IN and PACKADM ON privileges for collections, USAGE privileges for distinct types, and USE privileges for buffer pools, storage groups, and table spaces.
ARYROUTINEAUTH	SYSROUTINEAUTH	Records the privileges that are held by users on routines. A routine can be a user-defined function, cast function, or stored procedure.
ARYROUTINES	SYSROUTINES	Contains a row for every routine. A routine can be a user-defined function, cast function, or stored procedure.
ARYSCHEMAAUTH	SYSSCHEMAAUTH	Contains one or more rows for each user who is granted a privilege on a particular schema in the database.
ARYSEQUENCEAUTH	SYSSEQUENCEAUTH	Records the privileges that are held by users over sequences.
ARYSEQUENCES	SYSSEQUENCES	Contains one row for each identity column or user-defined sequence.
ARYSPEC_DATA	No equivalent	Contains the detailed data associated with each specification.
ARYSPEC_DIR	No equivalent	Contains one row for each saved specification.
ARYSTOGROUP	SYSSTOGROUP	Contains one row for each storage group.
ARYSYNONYMS	SYSSYNONYMS	Contains one row for each synonym of a table or view.
ARYTABAUTH	SYSTABAUTH	Records the privileges that users hold on tables and views.
ARYTABLEPART	SYSTABLEPART	Contains one row for each non-partitioned table space and one row for each partition of a partitioned table space.
ARYTABLES	SYSTABLES	Contains one row for each table, view, or alias.

SLR tables (SYSTOOLS)	Corresponding DB2 tables (SYSIBM)	Description
ARYTABLESPACE	SYSTABLESPACE	Contains one row for each table space.
ARYTRIGGERS	SYSTRIGGERS	Contains one row for each trigger.
ARYUSERAUTH	SYSUSERAUTH	Records the system privileges that are held by users.
ARYVIEWDEP	SYSVIEWDEP	Records the dependencies of views on tables, functions, and other views.
ARYVIEWS	SYSVIEWS	Contains one or more rows for each view, materialized query table, or user-defined Structured Query Language (SQL) function.
ARYVIEWSR	No equivalent	Contains user-defined function (UDF) and procedure-related information. The ARYVIEWSR table is loaded by the SLR update utility (ADHSJ002) with SQL text for SQ- based routines. In the DB2 catalog, this text is stored in SYSVIEWS table.
ARYVOLUMES	SYSVOLUMES	Contains one row for each volume of each storage group.
ARYVRLOG	No equivalent	Records utility execution statistics in multiple entries (one for each SLR table updated) for execution of the SLR update utility. Recovery Expert for z/OS also records some internal data in this table.
DR_ARCHIVE_LOGS00	No equivalent	
DR_IMAGE_COPY_CATL	No equivalent	

SLR maintenance considerations

As mentioned previously, the SLR is critical to the ability of Recovery Expert to recover dropped objects, and to synchronize data and DDL when an object is recovered to a prior point in time. To perform this function effectively, keep the SLR up to date with information about creation, alteration, and dropping of objects, and creation of recoverable resources such as full and incremental image copies, and also quiesce points.

The SLR is loaded initially with the contents of the DB2 catalog tables when Recovery Expert is installed using the ARYSJ002 job in the ARY.V2R1M0.SARYSAMP library. As mentioned previously, this process can take a long time, depending on the size of the DB2 catalog. Subsequently, it is the user's responsibility to ensure that the SLR is updated at appropriate frequent intervals. We recommend that you perform these updates at least daily and when important application changes occur in your DB2 environment.

Besides updating the SLR at frequent intervals, it is also important to manage the SLR database, which includes performing the following tasks:

- Take backups of DB2 objects of the SLR at frequent intervals. Recovery Expert cannot be used to recover the SLR database.
- Monitor and tune the SLR database including RUNSTATS and REORG, as appropriate.
- Prune¹ the contents of the SLR when its entries become outdated:
 - Entries that belong to objects that have been dropped and are no longer required

¹ Recovery Expert for z/OS does not provide a graphical user interface (GUI) or batch facility to prune the contents of the SLR.

- Entries that belong to existing objects but are no longer required, such as entries related to image copies, quiesce points, quiet times, and object definition levels
- Entries that seem to belong to the same object, but in fact belong to different objects that accidentally shared the same name

For example, this can occur when you create a table PROD with certain columns, perform updates, take image copies, and drop it some days later. A few days later, you create another table PROD with different attributes, and perform updates, take image copies, and so on. The SLR has information about both these tables and assumes that they are different versions of the same object. This can cause unpredictable results when performing a point-in-time recovery. You must prune the entries of the first PROD table because, in practice, it does not represent an older version of the current PROD table.

Note: The use of a date or time stamp is not likely to properly prune or delete data from the SLR tables. The time stamp that is entered in the tables does not span all the SLR tables and is the time stamp when the SLR update was run.

Important: Before you delete any entries in any of the SLR tables, first run a dummy delete (SELECT statement with the same predicates as the DELETE statement) to ensure that *only* the required rows are deleted.

Example D-1 through Example D-12 on page 261 provide examples of SQL that you can run to prune the SLR tables depending on the circumstances. An advisory is provided with each sample.

Note: Example D-12 on page 261 shows the deletion of entries in the ARYSPEC_DATA table. Rows also get deleted from this table when the delete function for a selected specification is started from the Recovery Expert for z/OS GUI client.

Example: D-1 When delete authority is removed for objects in DB2 for z/OS

```

//*****
//* TABLES: ARYDBAUTH, ARYRESAUTH, ARYSCHEMAAUTH and ARYUSERAUTH
//*
//* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED
//* ON YOUR SPECIFIC ENVIRONMENT AND REQUIREMENTS
//*
//* => CHANGE
//* 'DATABASE NAME' TO THE NAME OF THE DATABASE FOR THE AUTHORITY
//* 'GRANTED BY' TO THE GRANTOR OF THE AUTHORITY
//* 'PERSON GRANTED AUTH' TO THE PERSON GRANTED AUTHORITY
//* 'TABLESPACE NAME, BPOOL NAME, ETC' TO THE APPROPRIATE NAME
//* 'NAME OF SCHEMA' TO THE NAME OF THE SCHEMA
//*
//*****

DELETE FROM SYSTOOLS.ARYDBAUTH WHERE NAME = 'DATABASE NAME'
AND GRANTOR = 'GRANTED BY'
AND GRANTEE = 'PERSON GRANTED AUTH';

DELETE FROM SYSTOOLS.ARYRESAUTH WHERE
NAME = 'TABLESPACE NAME, BPOOL NAME, ETC'
AND GRANTOR = 'GRANTED BY'
AND GRANTEE = 'PERSON GRANTED AUTH'

```

```

AND QUALIFIER = 'DATABASE NAME' ;

DELETE FROM SYSTOOLS.ARYSCHEMAAUTH WHERE
  SCHEMANAME = 'NAME OF SCHEMA'
  AND GRANTOR = 'GRANTED BY'
  AND GRANTEE = 'PERSON GRANTED AUTH';

DELETE FROM SYSTOOLS.ARYUSERAUTH  WHERE
  AND GRANTOR = 'GRANTED BY'
  AND GRANTEE = 'PERSON GRANTED AUTH';

```

Example: D-2 When image copies are deleted for objects in DB2 for z/OS

```

/*****
/*  FOLLOWING SQL PERFORMS A DELETE FROM ARYCOPY WHICH CONTAINS SYSCOPY
/*  INFORMATION.
/*
/*  YOU SHOULD UNDERSTAND AND EVALUATE CAREFULLY HOW LONG YOU WILL NEED TO KEEP
/*  THIS INFORMATION SINCE IT IS A KEY FACTOR IN THE ABILITY OF THE RECOVERY
/*  EXPERT TOOL TO RETRIEVE AND RECREATE OBJECTS (OBJECT LEVEL DEFINITION)
/*
/*  ADDITIONAL COLUMNS SUCH AS ICTIME CAN BE USED TO FURTHER LIMIT
/*  YOUR DELETION BASED ON YOUR INDIVIDUAL REQUIREMENTS
*****/
/*  TABLE: ARYCOPY
/*
/*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR SPECIFIC
/*  REQUIREMENTS
/*
/*  => CHANGE
/*      'DATABASE NAME' TO THE DATABASE NAME
/*      'TABLESPACE NAME' TO THE TABLESPACE NAME
/*      'IMAGE DATE' TO THE SPECIFIC IMAGE COPY DATE
/*
/*      YOU COULD ALSO PERFORM A RANGE OF DATES BY CHANGING THE SQL STATEMENT
/*      TO USE THE BETWEEN STATEMENT
*****/

DELETE FROM SYSTOOLS.ARYCOPY WHERE DBNAME = 'DATABASE NAME'
AND TSNAME = 'TABLESPACE NAME' AND ICDATE = 'IMAGE DATE';

```

Example: D-3 When objects are dropped in DB2 for z/OS

```

/*****
/*  WHEN A TABLE, TABLESPACE, DATABASE IS DROPPED AND
/*  1. YOU NO LONGER WISH TO RECOVER / RESTORE THE OBJECT
/*  2. BEFORE YOU CREATE ANOTHER OBJECT WITH THE SAME NAME
/*
/*  YOU SHOULD UNDERSTAND AND EVALUATE CAREFULLY HOW LONG YOU WILL NEED TO
/*  KEEP THIS INFORMATION SINCE IT IS A KEY FACTOR IN THE ABILITY OF THE
/*  RECOVERY EXPERT TOOL TO RECOVER OR RECREATE OBJECTS
/*
*****/
/*  TABLES: ARYCOLAUTH, ARYCOLUMNS, ARYDATABASE, ARYINDEXES, ARYINDEXPART,
/*            ARYKEYS, ARYSYNONYMS, ARYTABAUTH, ARYTABLEPART, ARYTABLES,
/*            ARYTABLESPACE, ARYVIEWDEP, ARYVIEWS
/*
/*  WE RECOMMEND THAT YOU SHOULD DELETE FROM ALL TABLES
/*
/*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR

```

```

/** SPECIFIC ENVIRONMENT AND REQUIREMENTS
/**
/** => CHANGE
/**      'TABLE NAME' TO THE TABLE OR VIEW NAME
/**      'TABLE CREATOR' TO THE CREATOR OF THE TABLE OR VIEW
/**      'GRANTED BY' TO THE GRANTOR OF THE COLUMN AUTHORITY
/**      'PERSON GRANTED AUTH' TO THE PERSON GRANTED COLUMN AUTHORITY
/**      'DATABASE NAME' TO THE DATABASE NAME
/**      'DATABASE CREATOR' TO THE CREATOR OF THE DATABASE NAME
/**      'INDEX NAME' TO THE NAME OF THE INDEX
/**      'INDEX SPACE NAME' TO THE SPACE NAME FOR THE INDEX
/**      'INDEX CREATOR' TO THE CREATOR OF THE INDEX
/**      'SYNONYM NAME' TO THE NAME OF THE SYNONYM
/**      'SYNONYM CREATOR' TO THE NAME OF THE CREATOR OF THE SYNONYM
/**      'TABLESPACE NAME' TO THE NAME OF THE TABLESPACE
/**      'TABLE(T) OR VIEW(V) NAME' TO CORRECT TYPE NAME (TYPICALLY T
/**              OR V, BUT CAN BE SEVERAL OTHERS SO REVIEW CAREFULLY
/**      'TABLESPACE CREATOR' TO THE NAME OF THE CREATOR OF TABLESPACE
/**      'VIEW NAME' TO THE NAME OF THE VIEW
/**      'VIEW CREATOR' TO THE CREATOR OF THE VIEW
/**
/** *****
DELETE FROM SYSTOOLS.ARYCOLAUTH WHERE TNAME = 'TABLE NAME'
      AND CREATOR = 'TABLE CREATOR' AND GRANTOR = 'GRANTED BY'
      AND GRANTEE = 'PERSON GRANTED AUTH';

DELETE FROM SYSTOOLS.ARYCOLUMNS WHERE TBNAME = 'TABLE NAME'
      AND TBCREATOR = 'TABLE CREATOR' ;

DELETE FROM SYSTOOLS.ARYDATABASE WHERE NAME = 'DATABASE NAME'
      AND CREATOR = 'DATABASE CREATOR';

DELETE FROM SYSTOOLS.ARYINDEXES WHERE NAME = 'INDEX NAME'
      AND TBCREATOR = 'TABLE CREATOR' AND TBNAME = 'TABLE NAME'
      AND DBNAME = 'DATABASE NAME' AND
      INDEXSPACE = 'INDEX SPACE NAME' ;

DELETE FROM SYSTOOLS.ARYINDEXPART WHERE IXNAME = 'INDEX SPACE NAME'
      AND IXCREATOR = 'INDEX CREATOR';

DELETE FROM SYSTOOLS.ARYKEYS WHERE IXNAME = 'INDEX NAME'
      AND IXCREATOR = 'INDEX CREATOR';

DELETE FROM SYSTOOLS.ARYSYNONYMS WHERE NAME = 'SYNONYM NAME'
      AND CREATOR = 'SYNONYM CREATOR'
      AND TBNAME = 'TABLE NAME'
      AND TBCREATOR = 'TABLE CREATOR';

DELETE FROM SYSTOOLS.ARYTABAUTH WHERE TTNAME = 'TABLE NAME'
      AND TCREATOR = 'TABLE CREATOR' AND GRANTOR = 'GRANTED BY'
      AND GRANTEE = 'PERSON GRANTED AUTH';
-- NOTE GRANTEE CAN BE PUBLIC OR PUBLIC* AS WELL AS A SPECIFIC NAME

DELETE FROM SYSTOOLS.ARYTABLEPART WHERE DBNAME = 'DATABASE NAME'
      AND TSNAME = 'TABLESPACE NAME'
      AND IXNAME = 'INDEX NAME'
      AND IXCREATOR = 'INDEX CREATOR' ;
-- NOTE IXNAME AND IXCREATOR MAY BE NULL IF THERE IS NO PARTITIONING
DELETE FROM SYSTOOLS.ARYTABLES WHERE NAME = 'TABLE NAME'
      AND CREATOR = 'TABLE CREATOR' AND DBNAME = 'DATABASE NAME'

```



```

AND TSNAME = 'TABLESPACE NAME' AND
TYPE = 'TABLE(T) OR VIEW(V) NAME' ;

DELETE FROM SYSTOOLS.ARYTABLESPACE WHERE NAME = 'TABLESPACE NAME'
AND DBNAME = 'DATABASE NAME'
AND CREATOR = 'TABLESPACE CREATOR' ;

DELETE FROM SYSTOOLS.ARYVIEWDEP WHERE BNAME = 'TABLE NAME'
AND BCREATOR = 'TABLE CREATOR'
AND DNAME = 'VIEW NAME'
AND DCREATOR = 'VIEW CREATOR' ;

DELETE FROM SYSTOOLS.ARYVIEWS WHERE NAME = 'VIEW NAME'
AND CREATOR = 'VIEW CREATOR' ;

```

Example: D-4 When quiet time details are removed

```

//*****
/* TABLES: ARYQT, ARYQTG
/*
/* THE ARYQT AND ARYQTG TABLES ARE 'LINKED' BY GROUPID.
/* YOU SHOULD THEREFORE DELETE FROM BOTH TABLES
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC QUIET TIME REQUESTS WHILE USING RECOVERY EXPERT
/*
/* => CHANGE THE 'N' FOR GROUPID TO AN APPROPRIATE GROUPID NUMBER
/*
//*****
DELETE FROM SYSTOOLS.ARYQT WHERE GROUPID = N ;

DELETE FROM SYSTOOLS.ARYQTG WHERE GROUPID = N ;

```

Example: D-5 When stogroup or volume changes occur in DB2 for z/OS

```

//*****
/* TABLES: ARYSTOGROUP, ARYVOLUMES
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC ENVIRONMENT AND REQUIREMENTS
/*
/* => CHANGE
/* 'STOGROUP NAME' TO THE NAME OF THE STOGROUP
/* 'CREATOR OF STOGROUP' TO THE NAME OF THE CREATOR OF STOGROUP
/* 'VCAT NAME' TO THE NAME OF THE HIGH LEVEL QUALIFIER / VCAT
/* 'VOLUME ID' TO THE NAME OF THE VOLUME OR * FOR SMS MANAGED
/*
//*****
DELETE FROM SYSTOOLS.ARYSTOGROUP WHERE NAME = 'STOGROUP NAME'
AND CREATOR = 'CREATOR OF STOGROUP'
AND VCATNAME = 'VCAT NAME';

DELETE FROM SYSTOOLS.ARYVOLUMES WHERE SGNAME = 'STOGROUP NAME'
AND SGCREATOR = 'CREATOR OF STOGROUP'
AND VOLID = 'VOLUME ID';

```

Example: D-6 Deleting rows in SLR table SRYVRLOG

```
//*****
/* SYSTOOLS TABLE:ARYVRLOG
/*
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC ENVIRONMENT
/*
/* => CHANGE
/* 'TIMESTAMP ASSOCIATED WITH ROW TO DELETE' TO TIMESTAMP DESIRED
/* 'JOB NAME OF SLR JOB RUN' TO JOB NAME ASSOCIATED WITH TIMESTAMP
/*
//*****

DELETE FROM SYSTOOLS.ARYVRLOG WHERE
    RUNDTS = 'TIMESTAMP ASSOCIATED WITH ROW TO DELETE'
    AND JOBNAME = 'JOB NAME OF SLR JOB RUN' ;

/*
/* DELETE FROM SYSTOOLS.ARYVRLOG WHERE
/*     RUNDTS = '2006-09-05-11.06.48.032638'
/*     AND JOBNAME = 'W8SLR2' ;
```

Example: D-7 Deleting rows in ARYAUXRELS SLR table

```
//*****
/* TABLE: ARYAUXRELS
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC ENVIRONMENT
/*
/* => CHANGE
/* 'TOWNER' TO THE AUXILLARY TABLE OWNER NAME
/* 'TABLENAME' TO THE AUXILLARY TABLE NAME
/* 'DBNAME' TO THE DATABASE NAME WHICH CONTAINS THE AUX TABLE
/* 'TSNAME' TO THE TABLESPACE NAME WHICH CONTAINS THE AUX TABLE
/*
//*****

DELETE FROM SYSTOOLS.ARYAUXRELS WHERE
    AXTBOWNER = 'TOWNER'
    AND AXTBNAME = 'TABLENAME'
    AND AUXDBNAME = 'DBNAME'
    AND AXTSNAME = 'TSNAME' ;
```

Example: D-8 Deleting rows in ARYDATATYPES, ARYPARMS, ARYROUTINES, ARYVIEWSR SLR tables

```
//*****
/* TABLES: ARYDATATYPES, ARYPARMS, ARYROUTINES, ARYVIEWSR
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC ENVIRONMENT
/*
/* => CHANGE
/* 'DATA TYPE OWNER' TO THE OWNER OF THE FUNCTION / DATA TYPE
/* 'FUNCTION OR DATA TYPE NAME' TO THE NAME OF FUNCTION / DATA TYPE
/* 'CREATOR OF THE FUNCTION OR DATA TYPE' TO CREATOR NAME
//*****

DELETE FROM SYSTOOLS.ARYDATATYPES WHERE
    OWNER = 'DATA TYPE OWNER'
    AND NAME = 'FUNCTION OR DATA TYPE NAME'
```

```

        AND   CREATEDBY = 'CREATOR OF THE FUNCTION OR DATA TYPE';

DELETE FROM SYSTOOLS.ARYPARMS WHERE
    OWNER = 'DATA TYPE OWNER'
    AND   NAME = 'FUNCTION OR DATA TYPE NAME' ;

DELETE FROM SYSTOOLS.ARYROUTINES WHERE
    OWNER = 'DATA TYPE OWNER'
    AND   NAME = 'FUNCTION OR DATA TYPE NAME'
    AND   CREATEDBY = 'CREATOR OF THE FUNCTION OR DATA TYPE';

DELETE FROM SYSTOOLS.ARYVIEWSR   WHERE
    CREATOR = 'CREATOR OF THE FUNCTION OR DATA TYPE'
    AND   NAME = 'FUNCTION OR DATA TYPE NAME' ;

```

Example: D-9 Deleting row in ARYFIELDS SLR table

```

/*****
/* TABLES: ARYFIELDS
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC ENVIRONMENT
/*
/* => CHANGE
/*      'TABLE CREATOR' TO THE NAME OF THE TABLE CREATOR
/*      'TABLE NAME' TO THE NAME OF THE TABLE
/*      'FIELD PROC' TO THE NAME OF THE FIELD PROC
/*
*****/

DELETE FROM SYSTOOLS.ARYFIELDS WHERE
    TBCreator = 'TABLE CREATOR'
    AND   TBNAME = 'TABLE NAME'
    AND   FLDPROC = 'FIELD PROC' ;

```

Example: D-10 Deleting DBRM, PLAN, and PACKAGES information in SLR tables

```

/*****
/* TABLES: (ALL DBRM, PLAN AND PACKAGES)
/*      ARYDBRM, ARYPACKAGE, ARYPACKAUTH, ARYPACKDEP, ARYPACKLIST, ARYPKSYSTEM,
/*      ARYPLAN, ARYPLANAUTH, ARYPLANDEP, ARYPLSYSTEM
/*
/* THESE TABLES CONTAIN INFORMATION ABOUT DBRM'S, PLANS AND
/* PACKAGES; WHEN YOU DELETE FROM ONE OF THE TABLES IT IS
/* YOU SHOULD DELETE FROM ALL THESE SYSTOOLS TABLES
/*
/* FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/* SPECIFIC ENVIRONMENT
/*
/* => CHANGE
/*      'DBRM NAME' TO DBRM NAME
/*      'PDS WHERE DBRM RESIDES' TO NAME OF PDS WHERE DBRM RESIDES
/*      'PLAN NAME' TO THE NAME OF THE PLAN
/*      'PLAN CREATOR' TO THE NAME OF CREATOR OF THE PLAN
/*      'PACKAGE NAME' TO THE NAME OF PACKAGE
/*      'PACKAGE CREATOR' TO THE NAME OF CREATOR OF THE PACKAGE
/*      'PACKAGE QUALIFIER' TO THE NAME OF THE ASSIGNED PKG QUALIFIER
/*      'VERSION NAME ASSIGNED' TO THE NAME OF THE VERSION
/*      'GRANTED BY' IS THE GRANTOR OF THE AUTHORITY
/*      'PERSON GRANTED AUTH' TO THE PERSON GRANTED AUTHORITY

```

```

/**      'TABLE OR VIEW NAME' TO THE NAME OF THE TABLE OR VIEW
/**      'COLLECTION NAME' TO THE NAME OF THE COLLECTION
/**      'PLAN QUALIFIER' TO THE NAME OF THE ASSIGNED PLAN QUALIFIER
/**
/*******
DELETE FROM SYSTOOLS.ARYDBRM WHERE NAME = 'DBRM NAME'
      AND PDSNAME = 'PDS WHERE DBRM RESIDES'
      AND PLNAME = 'PLAN NAME'
      AND PLCREATOR = 'PLAN CREATOR' ;

DELETE FROM SYSTOOLS.ARYPACKAGE WHERE NAME = 'PACKAGE NAME'
      AND OWNER = 'PACKAGE CREATOR' AND QUALIFIER = 'PACKAGE QUALIFIER'
      AND VERSION = 'VERSION NAME ASSIGNED'
      AND PDSNAME = 'PDS WHERE DBRM RESIDES' ;

DELETE FROM SYSTOOLS.ARYPACKAUTH WHERE NAME = 'PACKAGE NAME'
      AND GRANTOR = 'GRANTED BY'
      AND GRANTEE = 'PERSON GRANTED AUTH'
      AND COLLIDX = 'COLLECTION NAME';

DELETE FROM SYSTOOLS.ARYPACKDEP WHERE BNAME = 'TABLE OR VIEW NAME'
      AND BQUALIFIER = 'PACKAGE QUALIFIER'
      AND DCOLLID = 'COLLECTION NAME';
      AND DOWNER = 'PACKAGE CREATOR' ;

DELETE FROM SYSTOOLS.ARYPACKLIST WHERE
      PLANNAME = 'PLAN NAME'
      AND COLLIDX = 'COLLECTION NAME';

DELETE FROM SYSTOOLS.ARYPKSYSTEM WHERE
      NAME = 'PACKAGE NAME'
      AND COLLIDX = 'COLLECTION NAME';

DELETE FROM SYSTOOLS.ARYPLAN WHERE NAME = 'PLAN NAME'
      AND CREATOR = 'PLAN CREATOR' AND QUALIFIER = 'PLAN QUALIFIER' ;

DELETE FROM SYSTOOLS.ARYPLANAUTH WHERE NAME = 'PLAN NAME'
      AND GRANTOR = 'GRANTED BY'
      AND GRANTEE = 'PERSON GRANTED AUTH';

DELETE FROM SYSTOOLS.ARYPLANDEP WHERE BNAME = 'PLAN NAME'
      AND BCREATOR = 'PLAN CREATOR' ;

DELETE FROM SYSTOOLS.ARYPLSYSTEM WHERE
      NAME = 'PLAN NAME' ;

```

Example: D-11 Deleting referential constraint information in SLR tables

```

/*******
/** TABLES: (CHECK CONSTRAINTS AND REFERENTIAL RELATIONSHIPS)
/**      ARYCHECKS, ARYCHECKS2, ARYFOREIGNKEYS, ARYRELS
/** THESE TABLES CONTAIN INFORMATION ABOUT CHECK CONSTRAINTS
/** AND REFERENTIAL INTEGRITY RELATIONSHIPS.
/** WHEN A CONSTRAINT OR REFERENTIAL RELATIONSHIP IS REMOVED
/** YOU SHOULD DELETE FROM ALL THESE SYSTOOLS TABLES
/**
/** FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/** SPECIFIC ENVIRONMENT
/**
/** => CHANGE

```

```

/**      'TABLE NAME' TO TABLE NAME WITH CONSTRAINT
/**      'OWNER NAME' TO TABLE OWNER NAME WITH THE CONSTRAINT
/**      'CONSTRAINT NAME' TO THE NAME OF THE CONSTRAINT
/**      'RI CREATOR' TO THE NAME OF THE CREATOR OF THE RI
/**      'RELATIONSHIP NAME' TO NAME OF FOREIGN KEY
/**
/*******

DELETE FROM SYSTOOLS.ARYCHECKS WHERE TBNAME = 'TABLE NAME'
      AND TBOWNER = 'OWNER NAME' AND CHECKNAME = 'CONSTRAINT NAME' ;

DELETE FROM SYSTOOLS.ARYCHECKS2 WHERE TBNAME = 'TABLE NAME'
      AND TBOWNER = 'OWNER NAME' AND CHECKNAME = 'CONSTRAINT NAME' ;

DELETE FROM SYSTOOLS.ARYFOREIGNKEYS WHERE CREATOR = 'RI CREATOR'
      AND TBNAME = 'TABLE NAME' AND RELNAME = 'RELATIONSHIP NAME';

DELETE FROM SYSTOOLS.ARYRELS WHERE CREATOR = 'RI CREATOR'
      AND TBNAME = 'TABLE NAME' AND RELNAME = 'RELATIONSHIP NAME';

```

Example: D-12 Deleting specifications in SLR tables

```

/*******
/** TABLES: ARYSPEC_DATA, ARYSPEC_DIR
/**
/** THE ARYSPEC_DATA AND ARYSPEC_DIR TABLES CONTAIN INFORMATION
/** ABOUT SAVED SPECIFICATIONS.
/** YOU SHOULD DELETE FROM BOTH TABLES
/**
/** FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
/** SPECIFIC ENVIRONMENT
/** => CHANGE
/**      'SPECOWNER' TO THE SPECIFICATION OWNER NAME
/**      'SPECNAME' TO THE SAVED SPECIFICATION NAME
/**
/*******

DELETE FROM SYSTOOLS.ARYSPEC_DATA WHERE OWNER = 'SPECOWNER'
      AND NAME = 'SPECNAME' ;

DELETE FROM SYSTOOLS.ARYSPEC_DIR WHERE OWNER = 'SPECOWNER'
      AND NAME = 'SPECNAME' ;

```

Sample jobs

In this appendix we list the detailed content of sample jobs used to define the configuration of our scenarios for some of the jobs in Chapter 3, “Our test environment” on page 39.

This appendix contains the following sample environment setup jobs:

- ▶ Control statement listings for Recovery Expert Agents 63 and 67
- ▶ Configuration files for Recovery Expert Agent 59
- ▶ Control file job listing for Recovery Expert Agent 63 and 64
- ▶ Sample job to create the repository VSAM data sets
- ▶ Sample job to create DR objects
- ▶ Sample CLISTs
- ▶ Sample PARMLIB member

Control statement listings for Recovery Expert Agents 63 and 67

The members shown in this appendix are listed in Example 3-2 on page 41.

The complete listing of control statements for DB2 subsystem member D8F1 from our PDS "SG7606.ARY.V2R1M0.SARYSAMP(D8F1)" is shown in Example E-1.

Example: E-1 Control statements for DB2 subsystem D8F1

```
*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID          = D8F1
UPDATE DB2 ZPARMS     = DSNZPAF1
UPDATE DB2 BOOTSTRAP1 = DB8FU.D8F1.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB8FU.D8F1.BSDS02
UPDATE DB2 LOADLIB1   = DB8F8.SDSNEXIT
UPDATE DB2 LOADLIB2   = DB8F8.SDSNLOAD
*UPDATE DB2 LOADLIB3   =
*UPDATE DB2 LOADLIB4   =
*UPDATE DB2 LOADLIB5   =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID          = D8F1
SET PRODUCT CFG       = NULL
SET PRODUCT VER       = NULL
*
UPDATE ARY PLAN1      = ARYPLAN1  DISPLAY DATA EXTRACT
UPDATE ARY PLAN2      = ARYPLAN2  SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3      = ARYPLAN3  RECOVERY PLAN GENERATION
UPDATE ARY PLAN4      = ARYPLAN4  JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5      = ARYPLAN5  LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1   = Y        USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2   = N        USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI   = Y        ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
```



```

* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX = SG7606.&USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS = Y          N = DB2 authorizations are
*                                not saved in schema level
*                                repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
UPDATE IPC IPC_GROUPER = N          Y = Enable Grouper-related
*                                table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
UPDATE QT GRP TOWNER = SYSTOOLS
UPDATE QT GRP TBNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
*                                XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*

```

```

* The DSN PREFIX maximum length is 21 characters. The following
* set of product controls are required and must be properly set
* to ensure proper log data recoveries. The VOLSERS statement
* value can be set to blanks, if required. A maximum of 3 volsers
* can be specified.
*
UPDATE LAS DSN PREFIX = SG7606
UPDATE LAS VOLSERS    = SBOX49,SBOX3Q,SBOX3R
UPDATE LAS DATA AUNIT = C
UPDATE LAS DATA PQTY  = 00005
UPDATE LAS DATA SQTY  = 00005
UPDATE LAS INDEX AUNIT = C
UPDATE LAS INDEX PQTY  = 00005
UPDATE LAS INDEX SQTY  = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
* Schema Level Repository Unicode data conversion information.
* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
UPDATE CCS SLR TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS   = 00037      EBCDIC
UPDATE CCS SLR DBCS   = 01200      UNICODE UT-16
UPDATE CCS SLR MIXED  = 01208      UNICODE UT-8
*
* Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS   = 00037      EBCDIC
UPDATE CCS ARY DBCS   = 01200      UNICODE UT-16
UPDATE CCS ARY MIXED  = 01208      UNICODE UT-8
*
*-----
* Sample statements to add/update default data set information
*-----
*
* File tailoring work data set allocation.
*
UPDATE FTW DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE FTW ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE FTW PQTY        = 00001      PRIMARY QTY
UPDATE FTW SQTY        = 00001      SECONDARY QTY
*UPDATE FTW SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE FTW SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE FTW SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*
* Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE ICF ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE ICF PQTY        = 00001      PRIMARY QTY
UPDATE ICF SQTY        = 00001      SECONDARY QTY
*UPDATE ICF SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE ICF SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE ICF SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL   = xxx

```

```

*UPDATE ICF EXPIREDT   = xxxxxxxx    EXPIRATION DATE
*UPDATE ICF RETPERIOD  = xxxxxxxx    RETENTION PERIOD
*UPDATE ICF FILENUM    = xxxxx       LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
  UPDATE RDA DEVICE      = SYSALLDA    DEVICE TYPE
  UPDATE RDA ALCUNIT     = C           C=CYLS, T=TRACKS
  UPDATE RDA PQTY        = 00001      PRIMARY QTY
  UPDATE RDA SQTY        = 00001      SECONDARY QTY
*UPDATE RDA SMSDC        = xxxxxxxx    SMS DATA CLASS
*UPDATE RDA SMSSC        = xxxxxxxx    SMS STORAGE CLASS
*UPDATE RDA SMSMC        = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL     = xxx
*UPDATE RDA EXPIREDT     = xxxxxxxx    EXPIRATION DATE
*UPDATE RDA RETPERIOD    = xxxxxxxx    RETENTION PERIOD
*UPDATE RDA FILENUM      = xxxxx       LABEL FILE NUMBER
*
* System Backup Recovery Services
*
* IPC_RBR = Y   - Generate recovery plans from system level backups
* IPC_RBR = N   - No recovery plans based on system backups created
*
  UPDATE IPC IPC_RBR      = Y           Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE       = xxxxxxxx
*UPDATE RBR EMC LOAD1    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD2    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD3    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD4    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD1    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD2    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
  UPDATE RBR PROF REPO   = ARY.PROFILES
  UPDATE RBR MAPS REPO   = ARY.PROFILE.MAPS
  UPDATE RBR CATS REPO   = ARY.PROFILE.CATS
  UPDATE RBR SYSBK REPO  = ARY.SYSBACK
  UPDATE RBR BKUP VOLS   = ARY.SYSBACK.VOLS
  UPDATE RBR BKUP SSID   = ARY.SYSBACK.SSIDS
  UPDATE RBR BKUP OBJS   = ARY.SYSBACK.OBJS
  UPDATE RBR REPORT REPO = ARY.BREPORT
  UPDATE RBR OFFLD REPO  = ARY.BREPORT
  UPDATE RBR PARMLIB     = SG7606.ARY.V2R1M0.SARYSAMP
  UPDATE RBR PARMLIB MBR = ARY#PARM

```

The complete listing of control statements for DB2 subsystem member D8F2 from our PDS "SG7606.ARY.V2R1M0.SARYSAMP(D8F2)" is shown in Example E-2.

Example: E-2 Control statement listing for DB2 subsystem D8F2

```

*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
  SET DB2 SSID          = D8F2
  UPDATE DB2 ZPARMS      = DSNZPAF2

```

```

UPDATE DB2 BOOTSTRAP1 = DB8FU.D8F2.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB8FU.D8F2.BSDS02
UPDATE DB2 LOADLIB1   = DB8F8.SDSNEXIT
UPDATE DB2 LOADLIB2   = DB8F8.SDSNLOAD
*UPDATE DB2 LOADLIB3   =
*UPDATE DB2 LOADLIB4   =
*UPDATE DB2 LOADLIB5   =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID          = D8F2
SET PRODUCT CFG       = NULL
SET PRODUCT VER       = NULL
*
UPDATE ARY PLAN1      = ARYPLAN1   DISPLAY DATA EXTRACT
UPDATE ARY PLAN2      = ARYPLAN2   SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3      = ARYPLAN3   RECOVERY PLAN GENERATION
UPDATE ARY PLAN4      = ARYPLAN4   JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5      = ARYPLAN5   LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1   = Y          USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2   = N          USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI   = Y          ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX = SG7606.&USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS = Y          N = DB2 authorizations are

```

```

*                                     not saved in schema level
*                                     repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
*   UPDATE IPC IPC_GROUPER = N           Y = Enable Grouper-related
*                                     table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
*   EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*   BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*   IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*   IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*   UPDATE QT GRP TOWNER = SYSTOOLS
*   UPDATE QT GRP TBNAME = ARYQTG
*   UPDATE QT GRP IXOWNER = SYSTOOLS
*   UPDATE QT GRP IXNAME = ARYQTGX
*   UPDATE QT ENTRY OWNER = SYSTOOLS
*   UPDATE QT ENTRY NAME = ARYQT
*   XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
*   UPDATE QT DATABASE = SYSTOOLS
*   UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
*   The ROWDATA VSAM data set is dynamically created by the log
*   analysis services when creating SQL from the log.
*
*   The DSN PREFIX maximum length is 21 characters. The following
*   set of product controls are required and must be properly set
*   to ensure proper log data recoveries. The VOLSERS statement
*   value can be set to blanks, if required. A maximum of 3 volsers
*   can be specified.
*
*   UPDATE LAS DSN PREFIX = SG7606
*   UPDATE LAS VOLSERS = SBOX49,SBOX3Q,SBOX3R
*   UPDATE LAS DATA AUNIT = C
*   UPDATE LAS DATA PQTY = 00005
*   UPDATE LAS DATA SQTY = 00005
*   UPDATE LAS INDEX AUNIT = C
*   UPDATE LAS INDEX PQTY = 00005
*   UPDATE LAS INDEX SQTY = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
*   Schema Level Repository Unicode data conversion information.

```

```

* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
UPDATE CCS SLR TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS   = 00037       EBCDIC
UPDATE CCS SLR DBCS   = 01200       UNICODE UT-16
UPDATE CCS SLR MIXED  = 01208       UNICODE UT-8
*
* Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS   = 00037       EBCDIC
UPDATE CCS ARY DBCS   = 01200       UNICODE UT-16
UPDATE CCS ARY MIXED  = 01208       UNICODE UT-8
*
*-----
* Sample statements to add/update default data set information
*-----
*
* File tailoring work data set allocation.
*
UPDATE FTW DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE FTW ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE FTW PQTY        = 00001       PRIMARY QTY
UPDATE FTW SQTY        = 00001       SECONDARY QTY
*UPDATE FTW SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE FTW SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE FTW SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*
* Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE ICF ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE ICF PQTY        = 00001       PRIMARY QTY
UPDATE ICF SQTY        = 00001       SECONDARY QTY
*UPDATE ICF SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE ICF SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE ICF SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL   = xxx
*UPDATE ICF EXPIREDT    = xxxxxxxx    EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxxx    RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx       LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
UPDATE RDA DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE RDA ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE RDA PQTY        = 00001       PRIMARY QTY
UPDATE RDA SQTY        = 00001       SECONDARY QTY
*UPDATE RDA SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE RDA SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE RDA SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL   = xxx
*UPDATE RDA EXPIREDT    = xxxxxxxx    EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxxx    RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx       LABEL FILE NUMBER
*
* System Backup Recovery Services
*

```

```

* IPC_RBR = Y - Generate recovery plans from system level backups
* IPC_RBR = N - No recovery plans based on system backups created
*
UPDATE IPC IPC_RBR = Y Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE = XXXXXXXX
*UPDATE RBR EMC LOAD1 = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR EMC LOAD2 = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR EMC LOAD3 = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR EMC LOAD4 = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR FDR LOAD1 = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR FDR LOAD2 = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
UPDATE RBR PROF REPO = ARY.PROFILES
UPDATE RBR MAPS REPO = ARY.PROFILE.MAPS
UPDATE RBR CATS REPO = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO = ARY.SYSBACK
UPDATE RBR BKUP VOLS = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID = ARY.SYSBACK.SSIDS
UPDATE RBR BKUP OBJS = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO = ARY.BREPORT
UPDATE RBR OFFLD REPO = ARY.BREPORT
UPDATE RBR PARMLIB = SG7606.ARY.V2R1M0.SARYSAMP
UPDATE RBR PARMLIB MBR = ARY#PARM

```

The complete listing of control statements for DB2 subsystem member D8G1 from our PDS "SG7606.ARY.V2R1M0.SARYSAMP(D8G1)" is shown in Example E-3.

Example: E-3 Control statement listing for DB2 subsystem D8G1

```

*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID = D8G1
UPDATE DB2 ZPARMS = DSNZPAG1
UPDATE DB2 BOOTSTRAP1 = DB8GU.D8G1.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB8GU.D8G1.BSDS02
UPDATE DB2 LOADLIB1 = DB8G8.SDSNEXIT
UPDATE DB2 LOADLIB2 = DB8G8.SDSNLOAD
*UPDATE DB2 LOADLIB3 =
*UPDATE DB2 LOADLIB4 =
*UPDATE DB2 LOADLIB5 =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID = D8G1
SET PRODUCT CFG = NULL
SET PRODUCT VER = NULL
*
UPDATE ARY PLAN1 = ARYPLAN1 DISPLAY DATA EXTRACT
UPDATE ARY PLAN2 = ARYPLAN2 SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3 = ARYPLAN3 RECOVERY PLAN GENERATION
UPDATE ARY PLAN4 = ARYPLAN4 JCL GENERATION AND SQL EXEC

```

```

UPDATE ARY PLAN5          = ARYPLAN5    LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY    = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1      = Y            USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2      = N            USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI     = Y            ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX    = SG7606.&USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS    = Y            RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS    = Y            N = DB2 authorizations are
*                                     not saved in schema level
*                                     repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
UPDATE IPC IPC_GROUPE    = N            Y = Enable Grouper-related
*                                     table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.

```



```

*
*
*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
UPDATE QT GRP TOWNER = SYSTOOLS
UPDATE QT GRP TNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
*
*XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*
* The DSN PREFIX maximum length is 21 characters. The following
* set of product controls are required and must be properly set
* to ensure proper log data recoveries. The VOLSERS statement
* value can be set to blanks, if required. A maximum of 3 volsers
* can be specified.
*
UPDATE LAS DSN PREFIX = SG7606
UPDATE LAS VOLSERS = SBOX49,SBOX3Q,SBOX3R
UPDATE LAS DATA AUNIT = C
UPDATE LAS DATA PQTY = 00005
UPDATE LAS DATA SQTY = 00005
UPDATE LAS INDEX AUNIT = C
UPDATE LAS INDEX PQTY = 00005
UPDATE LAS INDEX SQTY = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
* Schema Level Repository Unicode data conversion information.
* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
UPDATE CCS SLR TECHNQ = ER CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS = 00037 EBCDIC
UPDATE CCS SLR DBCS = 01200 UNICODE UT-16
UPDATE CCS SLR MIXED = 01208 UNICODE UT-8
*
* Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ = ER CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS = 00037 EBCDIC
UPDATE CCS ARY DBCS = 01200 UNICODE UT-16
UPDATE CCS ARY MIXED = 01208 UNICODE UT-8
*
*-----
* Sample statements to add/update default data set information
*-----
*

```

```

* File tailoring work data set allocation.
*
UPDATE FTW DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE FTW ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE FTW PQTY        = 00001      PRIMARY QTY
UPDATE FTW SQTY        = 00001      SECONDARY QTY
*UPDATE FTW SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE FTW SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE FTW SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*
* Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE ICF ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE ICF PQTY        = 00001      PRIMARY QTY
UPDATE ICF SQTY        = 00001      SECONDARY QTY
*UPDATE ICF SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE ICF SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE ICF SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL   = xxx
*UPDATE ICF EXPIREDT   = xxxxxxxx    EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxxx    RETENTION PERIOD
*UPDATE ICF FILENUM    = xxxxx       LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
UPDATE RDA DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE RDA ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE RDA PQTY        = 00001      PRIMARY QTY
UPDATE RDA SQTY        = 00001      SECONDARY QTY
*UPDATE RDA SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE RDA SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE RDA SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL   = xxx
*UPDATE RDA EXPIREDT   = xxxxxxxx    EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxxx    RETENTION PERIOD
*UPDATE RDA FILENUM    = xxxxx       LABEL FILE NUMBER
*
* System Backup Recovery Services
*
* IPC_RBR = Y - Generate recovery plans from system level backups
* IPC_RBR = N - No recovery plans based on system backups created
*
UPDATE IPC IPC_RBR      = Y           Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE      = xxxxxxxx
*UPDATE RBR EMC LOAD1   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD2   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD3   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD4   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD1   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD2   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
UPDATE RBR PROF REPO    = ARY.PROFILES
UPDATE RBR MAPS REPO    = ARY.PROFILE.MAPS
UPDATE RBR CATS REPO    = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO   = ARY.SYSBACK
UPDATE RBR BKUP VOLS     = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID    = ARY.SYSBACK.SSIDS

```

```

UPDATE RBR BKUP OBJS   = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO = ARY.BREPORT
UPDATE RBR OFFLD REPO  = ARY.BREPORT
UPDATE RBR PARMLIB     = SG7606.ARY.V2R1M0.SARYSAMP
UPDATE RBR PARMLIB MBR = ARY#PARM

```

The complete listing of control statements for DB2 subsystem member D8G2 from our PDS “SG7606.ARY.V2R1M0.SARYSAMP(D8G2)” is shown in Example E-4.

Example: E-4 Control statement listing for DB2 subsystem D8G2

```

*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID           = D8G2
UPDATE DB2 ZPARMS       = DSNZPAG2
UPDATE DB2 BOOTSTRAP1   = DB8GU.D8G2.BSDS01
UPDATE DB2 BOOTSTRAP2   = DB8GU.D8G2.BSDS02
UPDATE DB2 LOADLIB1     = DB8G8.SDSNEXIT
UPDATE DB2 LOADLIB2     = DB8G8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID           = D8G2
SET PRODUCT CFG        = NULL
SET PRODUCT VER        = NULL
*
UPDATE ARY PLAN1        = ARYPLAN1   DISPLAY DATA EXTRACT
UPDATE ARY PLAN2        = ARYPLAN2   SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3        = ARYPLAN3   RECOVERY PLAN GENERATION
UPDATE ARY PLAN4        = ARYPLAN4   JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5        = ARYPLAN5   LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY   = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1     = Y          USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2     = N          USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI    = Y          ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*

```

```

* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX = SG7606.&USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS = Y          N = DB2 authorizations are
*                                not saved in schema level
*                                repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
UPDATE IPC IPC_GROUPER = N          Y = Enable Grouper-related
*                                table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
UPDATE QT GRP TOWNER = SYSTOOLS
UPDATE QT GRP TBNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
*                                XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*

```

* The DSN PREFIX maximum length is 21 characters. The following
 * set of product controls are required and must be properly set
 * to ensure proper log data recoveries. The VOLSERS statement
 * value can be set to blanks, if required. A maximum of 3 volsers
 * can be specified.

```

*
UPDATE LAS DSN PREFIX = SG7606
UPDATE LAS VOLSERS    = SBOX49,SBOX3Q,SBOX3R
UPDATE LAS DATA AUNIT = C
UPDATE LAS DATA PQTY  = 00005
UPDATE LAS DATA SQTY  = 00005
UPDATE LAS INDEX AUNIT = C
UPDATE LAS INDEX PQTY  = 00005
UPDATE LAS INDEX SQTY  = 00005

```

*-----
 * Sample statements to add/update character conversion information
 *-----

*
 * Schema Level Repository Unicode data conversion information.
 * These values should not be changed. The IBM DB2 Recovery Expert
 * z/OS components require the following CCSID conversions to be
 * defined on the target systems.

```

*
UPDATE CCS SLR TECHNQ = ER      CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS   = 00037   EBCDIC
UPDATE CCS SLR DBCS   = 01200   UNICODE UT-16
UPDATE CCS SLR MIXED  = 01208   UNICODE UT-8

```

* Product output Unicode data conversion information.

```

*
UPDATE CCS ARY TECHNQ = ER      CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS   = 00037   EBCDIC
UPDATE CCS ARY DBCS   = 01200   UNICODE UT-16
UPDATE CCS ARY MIXED  = 01208   UNICODE UT-8

```

*-----
 * Sample statements to add/update default data set information
 *-----

* File tailoring work data set allocation.

```

*
UPDATE FTW DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE FTW ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE FTW PQTY        = 00001       PRIMARY QTY
UPDATE FTW SQTY        = 00001       SECONDARY QTY
*UPDATE FTW SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE FTW SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE FTW SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS

```

* Image copy output data set allocation defaults.

```

*
UPDATE ICF DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE ICF ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE ICF PQTY        = 00001       PRIMARY QTY
UPDATE ICF SQTY        = 00001       SECONDARY QTY
*UPDATE ICF SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE ICF SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE ICF SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL   = xxx

```

```

*UPDATE ICF EXPIREDT      = xxxxxxxx      EXPIRATION DATE
*UPDATE ICF RETPERIOD     = xxxxxxxx      RETENTION PERIOD
*UPDATE ICF FILENUM       = xxxxx         LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
  UPDATE RDA DEVICE        = SYSALLDA      DEVICE TYPE
  UPDATE RDA ALCUNIT       = C              C=CYLS, T=TRACKS
  UPDATE RDA PQTY          = 00001         PRIMARY QTY
  UPDATE RDA SQTY          = 00001         SECONDARY QTY
*UPDATE RDA SMSDC          = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC          = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC          = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL       = xxx
*UPDATE RDA EXPIREDT       = xxxxxxxx      EXPIRATION DATE
*UPDATE RDA RETPERIOD      = xxxxxxxx      RETENTION PERIOD
*UPDATE RDA FILENUM        = xxxxx         LABEL FILE NUMBER
*
* System Backup Recovery Services
*
* IPC_RBR = Y - Generate recovery plans from system level backups
* IPC_RBR = N - No recovery plans based on system backups created
*
  UPDATE IPC IPC_RBR        = Y              Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE         = xxxxxxxx
*UPDATE RBR EMC LOAD1      = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD2      = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD3      = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD4      = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD1      = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD2      = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
  UPDATE RBR PROF REPO      = ARY.PROFILES
  UPDATE RBR MAPS REPO      = ARY.PROFILE.MAPS
  UPDATE RBR CATS REPO      = ARY.PROFILE.CATS
  UPDATE RBR SYSBK REPO     = ARY.SYSBACK
  UPDATE RBR BKUP VOLS      = ARY.SYSBACK.VOLS
  UPDATE RBR BKUP SSID      = ARY.SYSBACK.SSIDS
  UPDATE RBR BKUP OBJS      = ARY.SYSBACK.OBJS
  UPDATE RBR REPORT REPO    = ARY.BREPORT
  UPDATE RBR OFFLD REPO     = ARY.BREPORT
  UPDATE RBR PARMLIB        = SG7606.ARY.V2R1M0.SARYSAMP
  UPDATE RBR PARMLIB MBR    = ARY#PARM

```

Configuration files for Recovery Expert Agent 59

The samples in this appendix are referred to in Example 3-4 on page 43.

The sample control statement listing from our PDS
SG7606.ARY.V2R1M0.SARYSAMP.DB8W(DB8A) to set up the product control file with DB2
subsystem DB8A information is shown in Example E-5.

Example: E-5 Control statement listing for DB2 subsystem DB8A

```
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID          = DB8A
UPDATE DB2 ZPARMS     = DSNZPARM
UPDATE DB2 BOOTSTRAP1 = DB8AU.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB8AU.BSDS02
UPDATE DB2 LOADLIB1   = DB8A8.SDSNEXIT
UPDATE DB2 LOADLIB2   = DB8A8.SDSNLOAD
*UPDATE DB2 LOADLIB3   =
*UPDATE DB2 LOADLIB4   =
*UPDATE DB2 LOADLIB5   =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID          = DB8A
SET PRODUCT CFG       = NULL
SET PRODUCT VER       = NULL
*
UPDATE ARY PLAN1      = ARYPLAN1   DISPLAY DATA EXTRACT
UPDATE ARY PLAN2      = ARYPLAN2   SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3      = ARYPLAN3   RECOVERY PLAN GENERATION
UPDATE ARY PLAN4      = ARYPLAN4   JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5      = ARYPLAN5   LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1   = Y          USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2   = N          USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI   = Y          ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX = SG7606.&USERID
```

```

*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS = Y          N = DB2 authorizations are
*                                not saved in schema level
*                                repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
UPDATE IPC IPC_GROUPER = N          Y = Enable Grouper-related
*                                table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
UPDATE QT GRP TOWNER = SYSTOOLS
UPDATE QT GRP TBNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
*                                XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*
* The DSN PREFIX maximum length is 21 characters. The following
* set of product controls are required and must be properly set
* to ensure proper log data recoveries. The VOLSERS statement
* value can be set to blanks, if required. A maximum of 3 volsers
* can be specified.
*
UPDATE LAS DSN PREFIX = SG7606

```



```

UPDATE LAS VOLSERS      =
UPDATE LAS DATA AUNIT  = C
UPDATE LAS DATA PQTY   = 00005
UPDATE LAS DATA SQTY   = 00005
UPDATE LAS INDEX AUNIT  = C
UPDATE LAS INDEX PQTY   = 00005
UPDATE LAS INDEX SQTY   = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
* Schema Level Repository Unicode data conversion information.
* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
UPDATE CCS SLR TECHNQ   = ER           CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS     = 00037        EBCDIC
UPDATE CCS SLR DBCS     = 01200        UNICODE UT-16
UPDATE CCS SLR MIXED    = 01208        UNICODE UT-8
*
* Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ   = ER           CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS     = 00037        EBCDIC
UPDATE CCS ARY DBCS     = 01200        UNICODE UT-16
UPDATE CCS ARY MIXED    = 01208        UNICODE UT-8
*
*-----
* Sample statements to add/update default data set information
*-----
*
* File tailoring work data set allocation.
*
UPDATE FTW DEVICE       = SYSALLDA     DEVICE TYPE
UPDATE FTW ALCUNIT      = C            C=CYLS, T=TRACKS
UPDATE FTW PQTY         = 00001        PRIMARY QTY
UPDATE FTW SQTY         = 00001        SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx     SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx     SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx     SMS MANAGEMENT CLASS
*
* Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE       = SYSALLDA     DEVICE TYPE
UPDATE ICF ALCUNIT      = C            C=CYLS, T=TRACKS
UPDATE ICF PQTY         = 00001        PRIMARY QTY
UPDATE ICF SQTY         = 00001        SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx     SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx     SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx     SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT     = xxxxxxxx     EXPIRATION DATE
*UPDATE ICF RETPERIOD    = xxxxxxxx     RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx         LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
UPDATE RDA DEVICE       = SYSDA        DEVICE TYPE

```

```

UPDATE RDA ALCUNIT      = C           C=CYLS, T=TRACKS
UPDATE RDA PQTY         = 00001       PRIMARY QTY
UPDATE RDA SQTY         = 00001       SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx    SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx    SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx    SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxxx    EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxxx    RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx        LABEL FILE NUMBER
*
* System Backup Recovery Services
*
* IPC_RBR = Y   - Generate recovery plans from system level backups
* IPC_RBR = N   - No recovery plans based on system backups created
*
UPDATE IPC IPC_RBR      = Y           Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE      = xxxxxxxx
*UPDATE RBR EMC LOAD1   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD2   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD3   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD4   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD1   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD2   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
UPDATE RBR PROF REPO    = ARY.PROFILES
UPDATE RBR MAPS REPO    = ARY.PROFILE.MAPS
UPDATE RBR CATS REPO    = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO   = ARY.SYSBACK
UPDATE RBR BKUP VOLS    = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID    = ARY.SYSBACK.SSIDS
UPDATE RBR BKUP OBJS    = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO  = ARY.BREPORT
UPDATE RBR OFFLD REPO   = ARY.OFFOPTS
UPDATE RBR PARMLIB      = SG7606.ARY.V2R1M0.SARYSAMP.DB8W
UPDATE RBR PARMLIB MBR  = ARY#PARM

```

The sample control statement listing from our PDS
SG7606.ARY.V2R1M0.SARYSAMP.DB8W(DB8W) to set up the product control file with DB2
subsystem DB8W information is shown in Example E-6.

Example: E-6 Control statement listing for DB2 subsystem DB8W

```

*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID          = DB8W
UPDATE DB2 ZPARMS      = DSNZPARM
UPDATE DB2 BOOTSTRAP1  = DB8WU.DB2.BSDS01
UPDATE DB2 BOOTSTRAP2  = DB8WU.DB2.BSDS02
UPDATE DB2 LOADLIB1    = DB8W8.SDSNEXIT
UPDATE DB2 LOADLIB2    = DB8W8.SDSNLOAD
*UPDATE DB2 LOADLIB3   =
*UPDATE DB2 LOADLIB4   =

```

```

*UPDATE DB2 LOADLIB5      =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID              = DB8W
SET PRODUCT CFG           = NULL
SET PRODUCT VER           = NULL
*
UPDATE ARY PLAN1          = ARYPLAN1   DISPLAY DATA EXTRACT
UPDATE ARY PLAN2          = ARYPLAN2   SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3          = ARYPLAN3   RECOVERY PLAN GENERATION
UPDATE ARY PLAN4          = ARYPLAN4   JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5          = ARYPLAN5   LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY     = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1       = Y          USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2       = N          USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI      = Y          ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX     = SG7606.&USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS     = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS     = Y          N = DB2 authorizations are
*                                     not saved in schema level
*                                     repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.

```

```

*-----
*
* UPDATE IPC IPC_GROUPER = N          Y = Enable Grouper-related
*                                     table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                                     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
* UPDATE QT GRP TOWNER = SYSTOOLS
* UPDATE QT GRP TBNAME = ARYQTG
* UPDATE QT GRP IXOWNER = SYSTOOLS
* UPDATE QT GRP IXNAME = ARYQTGX
* UPDATE QT ENTRY OWNER = SYSTOOLS
* UPDATE QT ENTRY NAME = ARYQT
*                                     XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
* UPDATE QT DATABASE = SYSTOOLS
* UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*
* The DSN PREFIX maximum length is 21 characters. The following
* set of product controls are required and must be properly set
* to ensure proper log data recoveries. The VOLSERS statement
* value can be set to blanks, if required. A maximum of 3 volsers
* can be specified.
*
* UPDATE LAS DSN PREFIX = SG7606
* UPDATE LAS VOLSERS =
* UPDATE LAS DATA AUNIT = C
* UPDATE LAS DATA PQTY = 00005
* UPDATE LAS DATA SQTY = 00005
* UPDATE LAS INDEX AUNIT = C
* UPDATE LAS INDEX PQTY = 00005
* UPDATE LAS INDEX SQTY = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
* Schema Level Repository Unicode data conversion information.
* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
* UPDATE CCS SLR TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
* UPDATE CCS SLR SBCS = 00037        EBCDIC

```

```

UPDATE CCS SLR DBCS      = 01200      UNICODE UT-16
UPDATE CCS SLR MIXED     = 01208      UNICODE UT-8
*
*   Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ    = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS      = 00037      EBCDIC
UPDATE CCS ARY DBCS      = 01200      UNICODE UT-16
UPDATE CCS ARY MIXED     = 01208      UNICODE UT-8
*
*-----
* Sample statements to add/update default data set information
*-----
*
*   File tailoring work data set allocation.
*
UPDATE FTW DEVICE        = SYSALLDA    DEVICE TYPE
UPDATE FTW ALCUNIT       = C           C=CYLS, T=TRACKS
UPDATE FTW PQTY          = 00001      PRIMARY QTY
UPDATE FTW SQTY          = 00001      SECONDARY QTY
*UPDATE FTW SMSDC        = xxxxxxxx   SMS DATA CLASS
*UPDATE FTW SMSSC        = xxxxxxxx   SMS STORAGE CLASS
*UPDATE FTW SMSMC        = xxxxxxxx   SMS MANAGEMENT CLASS
*
*   Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE        = SYSALLDA    DEVICE TYPE
UPDATE ICF ALCUNIT       = C           C=CYLS, T=TRACKS
UPDATE ICF PQTY          = 00001      PRIMARY QTY
UPDATE ICF SQTY          = 00001      SECONDARY QTY
*UPDATE ICF SMSDC        = xxxxxxxx   SMS DATA CLASS
*UPDATE ICF SMSSC        = xxxxxxxx   SMS STORAGE CLASS
*UPDATE ICF SMSMC        = xxxxxxxx   SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL     = xxx
*UPDATE ICF EXPIREDT     = xxxxxxxx   EXPIRATION DATE
*UPDATE ICF RETPERIOD    = xxxxxxxx   RETENTION PERIOD
*UPDATE ICF FILENUM      = xxxx       LABEL FILE NUMBER
*
*   Recovery output data set allocation defaults.
*
UPDATE RDA DEVICE        = SYSDA       DEVICE TYPE
UPDATE RDA ALCUNIT       = C           C=CYLS, T=TRACKS
UPDATE RDA PQTY          = 00001      PRIMARY QTY
UPDATE RDA SQTY          = 00001      SECONDARY QTY
*UPDATE RDA SMSDC        = xxxxxxxx   SMS DATA CLASS
*UPDATE RDA SMSSC        = xxxxxxxx   SMS STORAGE CLASS
*UPDATE RDA SMSMC        = xxxxxxxx   SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL     = xxx
*UPDATE RDA EXPIREDT     = xxxxxxxx   EXPIRATION DATE
*UPDATE RDA RETPERIOD    = xxxxxxxx   RETENTION PERIOD
*UPDATE RDA FILENUM      = xxxx       LABEL FILE NUMBER
*
*   System Backup Recovery Services
*
*   IPC_RBR = Y   -   Generate recovery plans from system level backups
*   IPC_RBR = N   -   No recovery plans based on system backups created
*
UPDATE IPC IPC_RBR       = Y           Y=generate system level backup
*
*   System Backup Recovery Services Options

```

```

*
*UPDATE RBR DEVICE      = XXXXXXXX
*UPDATE RBR EMC LOAD1   = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR EMC LOAD2   = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR EMC LOAD3   = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR EMC LOAD4   = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR FDR LOAD1   = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
*UPDATE RBR FDR LOAD2   = XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX
UPDATE RBR PROF REPO    = ARY.PROFILES
UPDATE RBR MAPS REPO    = ARY.PROFILE.MAPS
UPDATE RBR CATS REPO    = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO   = ARY.SYSBACK
UPDATE RBR BKUP VOLS    = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID    = ARY.SYSBACK.SSIDS
UPDATE RBR BKUP OBJS    = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO  = ARY.BREPORT
UPDATE RBR OFFLD REPO   = ARY.OFFOPTS
UPDATE RBR PARMLIB      = SG7606.ARY.V2R1M0.SARYSAMP.DB8W
UPDATE RBR PARMLIB MBR  = ARY#PARM
*
/*
//

```

Control file job listing for Recovery Expert Agent 63 and 64

The samples in this appendix are referred to in Example 3-5 on page 43.

A sample control statement listing from our PDS SG7606.ARY.V2R1M0.SARYSAMP.D9C1(D9C1) to set up the product control file with DB2 subsystem D9C1 information is shown in Example E-7.

Example: E-7 Control statement listing for DB2 subsystem D9C1

```

*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID          = D9C1
UPDATE DB2 ZPARMS     = DSNZPAC1
UPDATE DB2 BOOTSTRAP1 = DB9CL.D9C1.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB9CL.D9C1.BSDS02
UPDATE DB2 LOADLIB1   = DB9C9.SDSNEXIT
UPDATE DB2 LOADLIB2   = DB9C9.SDSNLOAD
*UPDATE DB2 LOADLIB3   =
*UPDATE DB2 LOADLIB4   =
*UPDATE DB2 LOADLIB5   =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID          = D9C1
SET PRODUCT CFG       = NULL
SET PRODUCT VER       = NULL
*

```

```

UPDATE ARY PLAN1      = ARYPLAN1  DISPLAY DATA EXTRACT
UPDATE ARY PLAN2      = ARYPLAN2  SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3      = ARYPLAN3  RECOVERY PLAN GENERATION
UPDATE ARY PLAN4      = ARYPLAN4  JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5      = ARYPLAN5  LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1   = Y          USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2   = N          USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI   = Y          ACTIVE LOG PRIORITY
*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX = &USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS = Y          N = DB2 authorizations are
*                                not saved in schema level
*                                repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
UPDATE IPC IPC_GROUPER = Y          Y = Enable Grouper-related
*                                table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*

```

```

* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
UPDATE QT GRP TOWNER = SYSTOOLS
UPDATE QT GRP TBNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
*
XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*
* The DSN PREFIX maximum length is 21 characters. The following
* set of product controls are required and must be properly set
* to ensure proper log data recoveries. The VOLSERS statement
* value can be set to blanks, if required. A maximum of 3 volsers
* can be specified.
*
UPDATE LAS DSN PREFIX = ARY
UPDATE LAS VOLSERS =
UPDATE LAS DATA AUNIT = C
UPDATE LAS DATA PQTY = 00005
UPDATE LAS DATA SQTY = 00005
UPDATE LAS INDEX AUNIT = C
UPDATE LAS INDEX PQTY = 00005
UPDATE LAS INDEX SQTY = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
* Schema Level Repository Unicode data conversion information.
* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
UPDATE CCS SLR TECHNQ = ER CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS = 00037 EBCDIC
UPDATE CCS SLR DBCS = 01200 UNICODE UT-16
UPDATE CCS SLR MIXED = 01208 UNICODE UT-8
*
* Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ = ER CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS = 00037 EBCDIC
UPDATE CCS ARY DBCS = 01200 UNICODE UT-16
UPDATE CCS ARY MIXED = 01208 UNICODE UT-8
*

```



```

*-----
* Sample statements to add/update default data set information
*-----
*
* File tailoring work data set allocation.
*
UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
UPDATE FTW ALCUNIT     = C              C=CYLS, T=TRACKS
UPDATE FTW PQTY        = 00001         PRIMARY QTY
UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC      = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC      = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC      = xxxxxxxx      SMS MANAGEMENT CLASS
*
* Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
UPDATE ICF ALCUNIT     = C              C=CYLS, T=TRACKS
UPDATE ICF PQTY        = 00001         PRIMARY QTY
UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC      = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC      = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC      = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL   = xxx
*UPDATE ICF EXPIREDT   = xxxxxxxx      EXPIRATION DATE
*UPDATE ICF RETPERIOD  = xxxxxxxx      RETENTION PERIOD
*UPDATE ICF FILENUM    = xxxx          LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
UPDATE RDA ALCUNIT     = C              C=CYLS, T=TRACKS
UPDATE RDA PQTY        = 00001         PRIMARY QTY
UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC      = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC      = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC      = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL   = xxx
*UPDATE RDA EXPIREDT   = xxxxxxxx      EXPIRATION DATE
*UPDATE RDA RETPERIOD  = xxxxxxxx      RETENTION PERIOD
*UPDATE RDA FILENUM    = xxxx          LABEL FILE NUMBER
*
* System Backup Recovery Services
*
* IPC_RBR = Y - Generate recovery plans from system level backups
* IPC_RBR = N - No recovery plans based on system backups created
*
UPDATE IPC IPC_RBR      = Y              Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE      = xxxxxxxx
*UPDATE RBR EMC LOAD1   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD2   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD3   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD4   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD1   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD2   = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
UPDATE RBR PROF REPO    = ARY.PROFILES
UPDATE RBR MAPS REPO    = ARY.PROFILE.MAPS

```

```

UPDATE RBR CATS REPO = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO = ARY.SYSBACK
UPDATE RBR BKUP VOLS = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID = ARY.SYSBACK.SSIDS
UPDATE RBR BKUP OBJS = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO = ARY.BREPORT
UPDATE RBR OFFLD REPO = ARY.OFFOPTS
UPDATE RBR PARMLIB = ARY.V2R1M0.SARYSAMP
UPDATE RBR PARMLIB MBR = ARY#PARM

```

A sample control statement listing from our PDS
SG7606.ARY.V2R1M0.SARYSAMP.D9C1(D9C2) to set up the product control file with DB2
subsystem D9C2 information is shown in Example E-8.

Example: E-8 Control statement listing for DB2 subsystem D9C2

```

*
*-----
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-----
*
SET DB2 SSID          = D9C2
UPDATE DB2 ZPARMS     = DSNZPAC2
UPDATE DB2 BOOTSTRAP1 = DB9CL.D9C2.BSDS01
UPDATE DB2 BOOTSTRAP2 = DB9CL.D9C2.BSDS02
UPDATE DB2 LOADLIB1   = DB9C9.SDSNEXIT
UPDATE DB2 LOADLIB2   = DB9C9.SDSNLOAD
*UPDATE DB2 LOADLIB3   =
*UPDATE DB2 LOADLIB4   =
*UPDATE DB2 LOADLIB5   =
*
*-----
* Sample statements to add/update ARY product plans
*-----
*
SET DB2 SSID          = D9C2
SET PRODUCT CFG       = NULL
SET PRODUCT VER       = NULL
*
UPDATE ARY PLAN1      = ARYPLAN1   DISPLAY DATA EXTRACT
UPDATE ARY PLAN2      = ARYPLAN2   SCHEMA LEVEL REPOSITORY LOAD
UPDATE ARY PLAN3      = ARYPLAN3   RECOVERY PLAN GENERATION
UPDATE ARY PLAN4      = ARYPLAN4   JCL GENERATION AND SQL EXEC
UPDATE ARY PLAN5      = ARYPLAN5   LOG ANALYSIS SERVICES
*
*-----
* Sample statements to add/update product message library
*-----
*
UPDATE ARY MSGLIBRARY = ARY.V2R1M0.SARYMENU
*
*-----
* Sample statements to add/update log services options
*-----
*
UPDATE ARY ARCHLOG1   = Y          USE ARCHIVE LOG 1
UPDATE ARY ARCHLOG2   = N          USE ARCHIVE LOG 2
UPDATE ARY ACTLOGPRI   = Y          ACTIVE LOG PRIORITY

```

```

*
*-----
* Sample statements to add/update data set prefix generation
*-----
*
* The DSN PREFIX maximum length is 17 characters. If NULL
* is specified then user id is used as data set prefix. Use &USERID
* in the prefix to insert user id. Example: TEST.&USERID will
* generate a data set prefix of 'TEST.MYID' where the user id is
* 'MYID'.
*
UPDATE ARY DSN PREFIX = &USERID
*
*-----
* Sample statements to turn on/off recovery options
*-----
*
UPDATE ARY RCVR AUTHS = Y          RECOVER DB2 OBJECT AUTHORIZATIONS
*
*-----
* Sample statements to add/update schema level repository data
* capture options.
*-----
*
UPDATE SLR LOAD AUTHS = Y          N = DB2 authorizations are
*                                not saved in schema level
*                                repository.
*
*-----
* Sample statements to add/update interproduct communication
* options.
*-----
*
UPDATE IPC IPC_GROUPE = Y          Y = Enable Grouper-related
*                                table recovery.
*
*-----
* Sample statements to add/update table activity quiet time
* repository names.
*-----
*
* EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
* BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
* IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
* IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
UPDATE QT GRP TBOWNER = SYSTOOLS
UPDATE QT GRP TBNAME = ARYQTG
UPDATE QT GRP IXOWNER = SYSTOOLS
UPDATE QT GRP IXNAME = ARYQTGX
UPDATE QT ENTRY OWNER = SYSTOOLS
UPDATE QT ENTRY NAME = ARYQT
*                                XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
UPDATE QT DATABASE = SYSTOOLS
UPDATE QT TABLESPACE = ARYTSQT
*
*-----
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.

```

```

*-----
*
* The ROWDATA VSAM data set is dynamically created by the log
* analysis services when creating SQL from the log.
*
* The DSN PREFIX maximum length is 21 characters. The following
* set of product controls are required and must be properly set
* to ensure proper log data recoveries. The VOLSERS statement
* value can be set to blanks, if required. A maximum of 3 volsers
* can be specified.
*
UPDATE LAS DSN PREFIX = ARY
UPDATE LAS VOLSERS    =
UPDATE LAS DATA AUNIT = C
UPDATE LAS DATA PQTY  = 00005
UPDATE LAS DATA SQTY  = 00005
UPDATE LAS INDEX AUNIT = C
UPDATE LAS INDEX PQTY  = 00005
UPDATE LAS INDEX SQTY  = 00005
*
*-----
* Sample statements to add/update character conversion information
*-----
*
* Schema Level Repository Unicode data conversion information.
* These values should not be changed. The IBM DB2 Recovery Expert
* z/OS components require the following CCSID conversions to be
* defined on the target systems.
*
UPDATE CCS SLR TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS SLR SBCS   = 00037      EBCDIC
UPDATE CCS SLR DBCS   = 01200      UNICODE UT-16
UPDATE CCS SLR MIXED  = 01208      UNICODE UT-8
*
* Product output Unicode data conversion information.
*
UPDATE CCS ARY TECHNQ = ER          CHARACTER CONVERSION TECHNIQUE
UPDATE CCS ARY SBCS   = 00037      EBCDIC
UPDATE CCS ARY DBCS   = 01200      UNICODE UT-16
UPDATE CCS ARY MIXED  = 01208      UNICODE UT-8
*
*-----
* Sample statements to add/update default data set information
*-----
*
* File tailoring work data set allocation.
*
UPDATE FTW DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE FTW ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE FTW PQTY        = 00001      PRIMARY QTY
UPDATE FTW SQTY        = 00001      SECONDARY QTY
*UPDATE FTW SMSDC      = xxxxxxxx    SMS DATA CLASS
*UPDATE FTW SMSSC      = xxxxxxxx    SMS STORAGE CLASS
*UPDATE FTW SMSMC      = xxxxxxxx    SMS MANAGEMENT CLASS
*
* Image copy output data set allocation defaults.
*
UPDATE ICF DEVICE      = SYSALLDA    DEVICE TYPE
UPDATE ICF ALCUNIT     = C           C=CYLS, T=TRACKS
UPDATE ICF PQTY        = 00001      PRIMARY QTY

```

```

UPDATE ICF SQTY          = 00001          SECONDARY QTY
*UPDATE ICF SMSDC        = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC        = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC        = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL     = xxx
*UPDATE ICF EXPIREDT     = xxxxxxxx      EXPIRATION DATE
*UPDATE ICF RETPERIOD    = xxxxxxxx      RETENTION PERIOD
*UPDATE ICF FILENUM      = xxxx          LABEL FILE NUMBER
*
* Recovery output data set allocation defaults.
*
UPDATE RDA DEVICE        = SYSALLDA      DEVICE TYPE
UPDATE RDA ALCUNIT       = C             C=CYLS, T=TRACKS
UPDATE RDA PQTY          = 00001          PRIMARY QTY
UPDATE RDA SQTY          = 00001          SECONDARY QTY
*UPDATE RDA SMSDC        = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC        = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC        = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL     = xxx
*UPDATE RDA EXPIREDT     = xxxxxxxx      EXPIRATION DATE
*UPDATE RDA RETPERIOD    = xxxxxxxx      RETENTION PERIOD
*UPDATE RDA FILENUM      = xxxx          LABEL FILE NUMBER
*
* System Backup Recovery Services
*
* IPC_RBR = Y - Generate recovery plans from system level backups
* IPC_RBR = N - No recovery plans based on system backups created
*
UPDATE IPC IPC_RBR       = Y             Y=generate system level backup
*
* System Backup Recovery Services Options
*
*UPDATE RBR DEVICE       = xxxxxxxx
*UPDATE RBR EMC LOAD1    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD2    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD3    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR EMC LOAD4    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD1    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
*UPDATE RBR FDR LOAD2    = xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
UPDATE RBR PROF REPO     = ARY.PROFILES
UPDATE RBR MAPS REPO     = ARY.PROFILE.MAPS
UPDATE RBR CATS REPO     = ARY.PROFILE.CATS
UPDATE RBR SYSBK REPO    = ARY.SYSBACK
UPDATE RBR BKUP VOLS     = ARY.SYSBACK.VOLS
UPDATE RBR BKUP SSID     = ARY.SYSBACK.SSIDS
UPDATE RBR BKUP OBJS     = ARY.SYSBACK.OBJS
UPDATE RBR REPORT REPO   = ARY.BREPORT
UPDATE RBR OFFLD REPO    = ARY.OFFOPTS
UPDATE RBR PARMLIB       = ARY.V2R1M0.SARYSAMP
UPDATE RBR PARMLIB MBR   = ARY#PARM

```

Sample job to create the repository VSAM data sets

The sample in this example is referred to in “Sample JCL from ARYREPO member” on page 47.

This is the sample job to create the repository VSAM data sets. This JCL is from the ARYREPO member of the SARSAMP installation library supplied as shown in Example E-9.

Example: E-9 JCL to create repository VSAM data sets

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUT1 DD *
DUMMY RECORD
/*
//SYSIN DD *
DELETE 'ARY.PROFILES'

SET MAXCC = 0

DEFINE CLUSTER
( NAME ('ARY.PROFILES')
  VOLUMES (SBOX3T)
  CYLINDERS (5 10)
  SHAREOPTIONS (3 3)
  INDEXED
  RECORDSIZE (1024 1024)
  KEYS (40 0)
  REUSE )
DATA ( NAME ('ARY.PROFILES.DATA'))
INDEX ( NAME ('ARY.PROFILES.INDEX'))

REPRO INFILE(SYSUT1)
OUTDATASET('ARY.PROFILES') REUSE

/*
/*
//*****
/* INVOKE IDCAMS *
//*****
//STEP2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUT1 DD *
DUMMY RECORD
/*
//SYSIN DD *
DELETE 'ARY.PROFILE.MAPS'

SET MAXCC = 0

DEFINE CLUSTER
( NAME ('ARY.PROFILE.MAPS')
  VOLUMES (SBOX3T)
  CYLINDERS (5 15)
  SHAREOPTIONS (3 3)
  INDEXED
  RECORDSIZE (1024 1024)
  KEYS (46 0)
  REUSE )
DATA ( NAME ('ARY.PROFILE.MAPS.DATA'))
INDEX ( NAME ('ARY.PROFILE.MAPS.INDEX'))

REPRO INFILE(SYSUT1)
OUTDATASET('ARY.PROFILE.MAPS') REUSE

/*
```

```

/*
/*****
/*          INVOKE IDCAMS                      *
/*****
//STEP3    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
        DELETE 'ARY.PROFILE.CATS'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.PROFILE.CATS')
              VOLUMES (SBOX3T)
              CYLINDERS (3 5)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (1024 1024)
              KEYS (40 0)
              REUSE )
            DATA ( NAME ('ARY.PROFILE.CATS.DATA'))
            INDEX ( NAME ('ARY.PROFILE.CATS.INDEX'))

        REPRO INFILE(SYSUT1)
              OUTDATASET('ARY.PROFILE.CATS') REUSE
/*
/*****
/*****
/*          INVOKE IDCAMS                      *
/*****
//STEP4    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
        DELETE 'ARY.SYSBACK'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.SYSBACK')
              VOLUMES (SBOX3T)
              CYLINDERS (5 10)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (1024 1024)
              KEYS (48 0)
              REUSE )
            DATA ( NAME ('ARY.SYSBACK.DATA'))
            INDEX ( NAME ('ARY.SYSBACK.INDEX'))

        REPRO INFILE(SYSUT1)
              OUTDATASET('ARY.SYSBACK') REUSE
/*

```

```

/*
/*****
/*          INVOKE IDCAMS                      *
/*****
//STEP5    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
      DELETE 'ARY.SYSBACK.VOLS'

      SET MAXCC = 0

      DEFINE CLUSTER
        ( NAME ('ARY.SYSBACK.VOLS')
          VOLUMES (SBOX3T)
          CYLINDERS (10 15)
          SHAREOPTIONS (3 3)
          INDEXED
          RECORDSIZE (1024 1024)
          KEYS (52 0)
          REUSE )
        DATA ( NAME ('ARY.SYSBACK.VOLS.DATA'))
        INDEX ( NAME ('ARY.SYSBACK.VOLS.INDEX'))

      REPRO INFILE(SYSUT1)
            OUTDATASET('ARY.SYSBACK.VOLS') REUSE

/*
/*****
/*****
/*          INVOKE IDCAMS                      *
/*****
//STEP6    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
      DELETE 'ARY.SYSBACK.SSIDS'

      SET MAXCC = 0

      DEFINE CLUSTER
        ( NAME ('ARY.SYSBACK.SSIDS')
          VOLUMES (SBOX3T)
          CYLINDERS (3 5)
          SHAREOPTIONS (3 3)
          INDEXED
          RECORDSIZE (1024 1024)
          KEYS (52 0)
          REUSE )
        DATA ( NAME ('ARY.SYSBACK.SSIDS.DATA')) -
        INDEX ( NAME ('ARY.SYSBACK.SSIDS.INDEX'))

      REPRO INFILE(SYSUT1)
            OUTDATASET('ARY.SYSBACK.SSIDS') REUSE

/*

```



```

//*****
//*          INVOKE IDCAMS                      *
//*****
//STEP7    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
        DELETE 'ARY.SYSBACK.OBJS'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.SYSBACK.OBJS')
              VOLUMES (SBOX3T)
              CYLINDERS (15 75)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (2620 2620)
              KEYS (30 0)
              REUSE )
            DATA ( NAME ('ARY.SYSBACK.OBJS.DATA'))
            INDEX ( NAME ('ARY.SYSBACK.OBJS.INDEX'))

        REPRO INFILE(SYSUT1)
              OUTDATASET('ARY.SYSBACK.OBJS') REUSE
/*
//*
//*****
//*          INVOKE IDCAMS                      *
//*****
//STEP8    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
        DELETE 'ARY.BREPORT'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.BREPORT')
              VOLUMES (SBOX3T)
              CYLINDERS (3 5)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (1024 1024)
              KEYS (50 0)
              REUSE )
            DATA ( NAME ('ARY.BREPORT.DATA'))
            INDEX ( NAME ('ARY.BREPORT.INDEX'))

        REPRO INFILE(SYSUT1)
              OUTDATASET('ARY.BREPORT') REUSE
/*
//*

```

```

//*****
//*          INVOKE IDCAMS                      *
//*****
//STEP9    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
        DELETE 'ARY.OFFOPTS'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.OFFOPTS')
              VOLUMES (SBOX3T)
              CYLINDERS (3 5)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (1280 1280)
              KEYS (4 0)
              REUSE )
            DATA ( NAME ('ARY.OFFOPTS.DATA'))
            INDEX ( NAME ('ARY.OFFOPTS.INDEX'))

        REPRO INFILE(SYSUT1)
              OUTDATASET('ARY.OFFOPTS') REUSE

/*
//*
//*****
//*          INVOKE IDCAMS                      *
//*****
//STEP10   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSUT1    DD  *
DUMMY RECORD
/*
//SYSIN     DD  *
        DELETE 'ARY.OBJECTS'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.OBJECTS')
              VOLUMES (SBOX3T)
              CYLINDERS (3 5)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (1300 1300)
              KEYS (62 0)
              REUSE )
            DATA ( NAME ('ARY.OBJECTS.DATA'))
            INDEX ( NAME ('ARY.OBJECTS.INDEX'))

```

```

        REPRO INFILE(SYSUT1)
            OUTDATASET('ARY.OBJECTS') REUSE
/*
//

```

This sample JCL is referred to in “Sample JCL from ARYMOVER member to create mover repository” on page 47.

A sample JCL to create the mover repository is shown in Example E-10.

Example: E-10 JCL to create mover repository

```

//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSUT1   DD   *
DUMMY RECORD
/*
//SYSIN     DD   *
        DELETE 'ARY.SG7606.MOVDATA'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('ARY.SG7606.MOVDATA')
              VOLUMES (SBOX3T)
              CYLINDERS (5 10)
              SHAREOPTIONS (3 3)
              INDEXED
              RECORDSIZE (1024 1024)
              KEYS (6 0)
              REUSE )
            DATA ( NAME ('ARY.SG7606.MOVDATA.DATA'))
            INDEX ( NAME ('ARY.SG7606.MOVDATA.INDEX'))

        REPRO INFILE(SYSUT1)
            OUTDATASET('ARY.SG7606.MOVDATA') REUSE
/*

```

This sample JCL is referred to in “Sample JCLs from ARY#RBA and ARY#RBA to create RBA capture repository” on page 47.

A sample JCL to create the VSAM data set to capture RBA information when the RBA capture utility is run is shown in Example E-11.

Example: E-11 JCL to create VSAM data set to capture RBA information

```

//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSUT1   DD   *
DUMMY RECORD
/*
//SYSIN     DD   *
        DELETE 'SG7606.SC53.ALLSSIDS.RBA'

        SET MAXCC = 0

        DEFINE CLUSTER
            ( NAME ('SG7606.SC53.ALLSSIDS.RBA')

```

```

VOLUMES (SBOX3S) -
CYLINDERS (50 50) -
SHAREOPTIONS (3 3) -
INDEXED -
RECORDSIZE (80 80) -
KEYS (21 0) -
REUSE ) -
DATA ( NAME ('SG7606.SC53.ALLSSIDS.RBA.DATA')) -
INDEX ( NAME ('SG7606.SC53.ALLSSIDS.RBA.INDEX'))

REPRO INFILE(SYSUT1) -
OUTDATASET('SG7606.SC53.ALLSSIDS.RBA') REUSE

/*

```

This sample job is referred to in “Sample JCLs from ARY#RBA and ARY#RBA to create RBA capture repository” on page 47.

A sample job for the RBA capture utility started task is shown in Example E-12. The PDS member “SG7606.ARY.V2R1M0.SARYSAMP.D8FG(RBRSSID)” shown in Example E-12. contains:

- ▶ D8F1
- ▶ D8F2
- ▶ D8G1
- ▶ D8G2

Example: E-12 Sample proc for the RBA capture utility started task

```

//ARY#RBA EXEC PGM=ARY#RBA,REGION=8M,PARM='1'
/*
//STEPLIB DD DISP=SHR,DSN=ARY.V2R1M0.SARYLOAD
//ARY#RBA DD DISP=SHR,DSN=ARY.BKRES53.SSID.RBA
//SYSIN DD DISP=SHR,DSN=SG7606.ARY.V2R1M0.SARYSAMP.D8FG(RBRSSID)
//SYSPRINT DD SYSOUT=*

```

This sample job is referred to in “Sample JCL from ARYGDG” on page 47.

Example E-13 is the sample job to create GDG bases to back up the VSAM repository data sets.

Example: E-13 Sample job to create GDG bases

```

//DEFGDG EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
DEFINE GDG (NAME('ARY.SG7606.PROFILE.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.PROFMAP.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.PROFCAT.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.SYSBACK.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.SYSBVOL.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.SYSSSID.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.BREPORT.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.OFFOPTS.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.SYSBOBJ.ARYGDGSF') SCRATCH LIMIT(10))
DEFINE GDG (NAME('ARY.SG7606.OBJECTS.ARYGDGSF') SCRATCH LIMIT(10))
/*

```

Sample job to create DR objects

This sample JCL in this example is referred to in “SBRS DR objects” on page 48.

A sample JCL to create disaster recovery objects is shown in Example E-14. This information is provided in member ARYDDLRL from the supplied SARYSAMP installation library.

Example: E-14 Sample JCL to create DR objects

```
//ARYDDL0 EXEC PGM=IKJEFT01,COND=(4,LT),DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=#SDSNLOAD
//*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(#SSID)
  RUN PROGRAM(DSNTEP2) PLAN(#DSNTEP2) -
    LIB('#SDSNRNL') PARM('/ALIGN(MID)')
END
//SYSIN DD *
--
-- Instructions:
--
-- 1) Change STGX to the desired storage group name.
-- 2) Note: PRIQTY and SECQTY are estimated values, your
--    requirements may differ.
--
CREATE TABLESPACE ARYDRCPY
  IN          SYSTOOLS
  USING       STOGROUP STGX
  PRIQTY      1000
  SECQTY      500
  ERASE       NO
  LOCKSIZE    PAGE
  BUFFERPOOL  BPO
  SEGSIZE     16
  CLOSE       NO
  CCSID       EBCDIC;

COMMIT;

CREATE TABLE SYSTOOLS.DR_IMAGE_COPY_CATL
  (DBNAME      CHAR(08) NOT NULL
  ,TSNAME      CHAR(08) NOT NULL
  ,DSNUM       INTEGER  NOT NULL
  ,START_RBA   CHAR(06) NOT NULL
  ,ICTYPE      CHAR(01) NOT NULL
  ,FILESEQNO   INTEGER  NOT NULL
  ,DEVTYPE     CHAR(08) NOT NULL
  ,DSNAME      CHAR(44) NOT NULL
  ,DSVOLSER    VARCHAR(251) NOT NULL
  ,TIMESTAMP   TIMESTAMP NOT NULL
  ,ICBACKUP    CHAR(02)  NOT NULL
  ,ICUNIT      CHAR(01)  NOT NULL)
  IN SYSTOOLS.ARYDRCPY;

COMMIT;
```

```

CREATE TYPE 2 INDEX SYSTOOLS.DR_ICOPY_CATL_IX
ON SYSTOOLS.DR_IMAGE_COPY_CATL
(DBNAME          ASC
,TSNAME          ASC
,START_RBA       ASC
,TIMESTAMP       ASC)
USING STOGROUP STGX
PRIQTY 200
SECQTY 100
ERASE NO
CLUSTER
BUFFERPOOL BPO
CLOSE NO;
COMMIT;

CREATE TABLESPACE ARYDRARC
IN SYSTOOLS
USING STOGROUP STGX
PRIQTY 100
SECQTY 100
ERASE NO
LOCKSIZE PAGE
BUFFERPOOL BPO
SEGSIZE 4
CLOSE NO
CCSID EBCDIC;

COMMIT;

CREATE TABLE SYSTOOLS.DR_ARCHIVE_LOGS00
(DB2_SSID          CHAR(04) NOT NULL
,PRODUCT_F MID     CHAR(03) NOT NULL
,ARCHLOG_NAME      CHAR(44) NOT NULL
,DR_ARCHLOG_NAME   CHAR(44) NOT NULL
,VOLUME_SEQ_NO     SMALLINT NOT NULL WITH DEFAULT 1
,COPY_IND          CHAR(01) NOT NULL
,START_TIME        CHAR(08) NOT NULL
,END_TIME          CHAR(08) NOT NULL
,START_RBA         CHAR(06) NOT NULL FOR BIT DATA
,END_RBA           CHAR(06) NOT NULL FOR BIT DATA
,START_LRSN        CHAR(06) NOT NULL FOR BIT DATA
,END_LRSN          CHAR(06) NOT NULL FOR BIT DATA
,OLD_CATALOGED     CHAR(01) NOT NULL
,NEW_CATALOGED     CHAR(01) NOT NULL
,OLD_UNIT          CHAR(08) NOT NULL
,NEW_UNIT          CHAR(08) NOT NULL
,OLD_VOLUME        CHAR(06) NOT NULL
,NEW_VOLUME        CHAR(06) NOT NULL
,PRIMARY_SPACE     INTEGER NOT NULL
,SECONDARY_SPACE   INTEGER NOT NULL
,BLOCK_SIZE        SMALLINT NOT NULL
,FILE_SEQ_NO       CHAR(03) NOT NULL)
IN SYSTOOLS.ARYDRARC;
COMMIT;

CREATE TYPE 2 UNIQUE INDEX SYSTOOLS.DR_ARCHIVE_LOG_IX
ON SYSTOOLS.DR_ARCHIVE_LOGS00
(DB2_SSID          ASC
,ARCHLOG_NAME      ASC
,START_TIME        DESC)
USING STOGROUP STGX

```

```

        PRIQTY 50
        SECQTY 50
        ERASE NO
        CLUSTER
        BUFFERPOOL BPO
        CLOSE NO;
    COMMIT;

    CREATE VIEW SYSTOOLS.DR_ICOPY_CATLV
    AS SELECT *
    FROM SYSTOOLS.DR_IMAGE_COPY_CATL;
    COMMIT;

    CREATE VIEW SYSTOOLS.DR_ARCHIVE_LOGSV
    AS SELECT *
    FROM SYSTOOLS.DR_ARCHIVE_LOGS00;
    COMMIT;

    CREATE ALIAS SYSTOOLS.DR_ICOPY_CATL
    FOR SYSTOOLS.DR_ICOPY_CATLV;
    COMMIT;

    CREATE ALIAS SYSTOOLS.DR_ARCHIVE_LOGS
    FOR SYSTOOLS.DR_ARCHIVE_LOGSV;
    COMMIT;
/*
/**

```

Sample CLISTs

The sample CLISTs are referred to in “CLIST members” on page 48.

The sample ARYV21 CLIST is shown in Example E-15.

Example: E-15 ARYV21 CLIST

```

/* PROC 0 CLISTLIB(ARY.V2R1MO.SARYSAMP) */
PROC 0 CLISTLI1(SYS1.OS390.CLIST)

    IF &SYSISPF ^= ACTIVE THEN +
    DO
        WRITE ARYV21 CLIST ERROR: ISPF NOT ACTIVE.
        EXIT CODE(20)
    END

    CONTROL NOMSG
    FREE FILE(ARYV21)
    CONTROL MSG

    ALLOC FILE(ARYV21) DA('&CLISTLI1') SHR REUSE

    ATLIB ACTIVATE APPLICATION(CLIST) FILE(ARYV21)

    ISPEXEC SELECT CMD(ARYV210) +
        NEWAPPL(ARY) PASSLIB

    CONTROL NOMSG
    ATLIB DEACTIVATE APPLICATION(CLIST)

```

```

FREE FILE(ARYV21)
CONTROL MSG

```

A sample ARYV210 CLIST is shown in Example E-16.

Example: E-16 ARYV210 CLIST

```

/*                                                    */
PROC 0 RBRVLV(ARY.V2R1M0)                            +
      RBRLOAD1(ARY.V2R1M0.SARYLOAD)                  +
      MLIB(SARYMENU)                                  +
      PLIB(SARYPENU)                                  +
      SLIB(SARYSLIB)                                  +
      LLIB(SARYLOAD)                                  +
      DB2CNTFL(ARY.DB2PARMS.CONTROL2)                +
      RBRBPROF(ARY.PROFILES)                         +
      RBRBPMAP(ARY.PROFILE.MAPS)                     +
      RBRBPCAT(ARY.PROFILE.CATS)                     +
      RBRBACK(ARY.SYSBACK)                           +
      RBRBVOL(ARY.SYSBACK.VOLS)                      +
      RBRBSSD(ARY.SYSBACK.SSIDS)                     +
      RBRBBOBJ(ARY.SYSBACK.OBJS)                     +
      RBRPOBJS(ARY.OBJECTS)                          +
      RBRBREPT(ARY.BREPORT)                          +
      RBRBOFFL(ARY.OFFOPTS)                          +
      RBRMOVER(ARY.SG7606.MOVDATA)                   +
      RBRSSRBA(SG7606.SC53.ALLSSIDS.RBA)             +
      RBRBUPRE(ARY.BKRES53)                          +
      RBRBUSUF(ARYGDGSF(+1))                         +
      EMCLOAD1()                                     +
      EMCLOAD2()                                     +
      EMCLOAD3()                                     +
      FDRLOAD1()                                     +
      FDRLOAD2()                                     +
      PARMLDSN(ARY.V2R1M0.SARYSAMP)                  +
      PARMLMBR(ARY#PARM)                             +
      ISPMLIB1()                                     +
      ISPMLIB2()                                     +
      ISPMLIB3()                                     +
      ISPTLIB1()                                     +
      ISPTLIB2()                                     +
      ISPTLIB3()                                     +
      ISPLLIB1()                                     +
      ISPLLIB2()                                     +
      ISPLLIB3()                                     +
      SCFDD()                                         +
      RBRVER(V2R1)                                   +
      USERIND(ARY)                                   +

ISPEXEC VPUT (RBRLOAD1 PARMLDSN PARMLMBR) ASIS
ISPEXEC VPUT (RBRVLV) ASIS
ISPEXEC VPUT (MLIB PLIB SLIB LLIB) ASIS
ISPEXEC VPUT (DB2CNTFL) ASIS
ISPEXEC VPUT (RBRBPROF RBRBPMAP RBRBACK RBRBVOL) ASIS
ISPEXEC VPUT (RBRBPCAT RBRBSSD RBRBREPT RBRMOVER) ASIS
ISPEXEC VPUT (RBRBOFFL RBRBBOBJ RBRPOBJS) ASIS
ISPEXEC VPUT (RBRBUPRE RBRBUSUF RBRSSRBA) ASIS
ISPEXEC VPUT (EMCLOAD1 EMCLOAD2 EMCLOAD3) ASIS
ISPEXEC VPUT (FDRLOAD1 FDRLOAD2) ASIS
ISPEXEC VPUT (ISPMLIB1 ISPMLIB2 ISPMLIB3) ASIS

```



```

ISPEXEC VPUT (ISPTLIB1 ISPTLIB2 ISPTLIB3) ASIS
ISPEXEC VPUT (ISPLLIB1 ISPLLIB2 ISPLLIB3) ASIS
ISPEXEC VPUT (SCFDD) ASIS
ISPEXEC VPUT (RBRVER USERIND) ASIS

/*CONTROL LIST SYMLIST CONLIST MSG */
CONTROL NOMSG
FREE FILE(DB2PARMS)
FREE FILE(SYSIN)
FREE FILE(SYSOUT)
IF &SCFDD GT &Z THEN +
    FREE FILE(&SCFDD)
CONTROL MSG

IF &EMCLOAD3 GT &Z && &EMCLOAD2 GT &Z && &EMCLOAD1 GT &Z THEN +
    ISPEXEC LIBDEF ISPLLIB DATASET ID('&RBRLVL..&LLIB'
                                     '&EMCLOAD1'
                                     '&EMCLOAD2'
                                     '&EMCLOAD3') UNCOND
ELSE +
    IF &EMCLOAD2 GT &Z && &EMCLOAD1 GT &Z THEN +
        ISPEXEC LIBDEF ISPLLIB DATASET ID('&RBRLVL..&LLIB'
                                           '&EMCLOAD1'
                                           '&EMCLOAD2') UNCOND
    ELSE +
        IF &EMCLOAD1 GT &Z THEN +
            ISPEXEC LIBDEF ISPLLIB DATASET ID('&RBRLVL..&LLIB'
                                              '&EMCLOAD1') UNCOND
        ELSE +
            ISPEXEC LIBDEF ISPLLIB DATASET ID('&RBRLVL..&LLIB') UNCOND

ISPEXEC LIBDEF ISPLMLIB DATASET ID('&RBRLVL..&MLIB') UNCOND
ISPEXEC LIBDEF ISPLPLIB DATASET ID('&RBRLVL..&PLIB') UNCOND
ISPEXEC LIBDEF ISPLSLIB DATASET ID('&RBRLVL..&SLIB') UNCOND

IF &SCFDD GT &Z THEN +
    ALLOC FILE(&SCFDD) DUMMY REUSE

ISPEXEC VGET (ZUSER ZPREFIX) ASIS

IF &ZPREFIX = &Z THEN +
    SET &TL = &ZUSER
ELSE +
    IF &ZPREFIX ^= &ZUSER THEN +
        SET &TL = &STR(&ZPREFIX..&ZUSER)
    ELSE +
        SET &TL = &ZUSER

CONTROL NOMSG NOFLUSH
ALLOC F(XX) DA('&TL..RBR0210.ISPTLIB') SHR
IF &LASTCC NE 0 THEN DO
    ALLOC F(XX) DA('&TL..RBR0210.ISPTLIB') SPACE(1 1) DIR(30) +
        CYLINDERS LRECL(80) BLKSIZE(800) DSORG(PO) RECFM(F B) NEW
END
FREE F(XX)
CONTROL MSG

ISPEXEC LIBDEF ISPTLIB DATASET ID('&TL..RBR0210.ISPTLIB') UNCOND
ISPEXEC LIBDEF ISPTABL DATASET ID('&TL..RBR0210.ISPTLIB') UNCOND

```

```

ARY$MAIN

ISPEXEC LIBDEF ISPLLIB
ISPEXEC LIBDEF ISPLMLIB
ISPEXEC LIBDEF ISPLPLIB
ISPEXEC LIBDEF ISPSLIB
ISPEXEC LIBDEF ISPTLIB
ISPEXEC LIBDEF ISPTABL

CONTROL NOMSG

IF &SCFDD GT &Z THEN +
  FREE FILE(&SCFDD)

CONTROL MSG

```

A sample ARYTSOC CLIST is shown in Example E-17.

Example: E-17 ARYVTSOC CLIST

```

PROC 2 TOKEN1 TOKEN2 DCB()
  ISPEXEC VGET (RBRLOAD1)
  IF &RBRLOAD1 GT &Z THEN +
    CALL '&RBRLOAD1(ARY$TSOC)' '&TOKEN1 &TOKEN2 &DCB'

```

A sample ARYCAPS CLIS is shown in Example E-18.

Example: E-18 ARYCAPS CLIST

```

ISREDIT MACRO
ISREDIT CAPS ON

```

Sample PARMLIB member

This sample PARMLIB member is referred to in “PARM library” on page 49.

A sample PARMLIB member ARY#PARM in the parmlib data set SG7606.ARY.V2R1M0.SARYSAMP is shown in Example E-19.

Example: E-19 ARY#PARM parmlib member

```

RBR_BACKUP_AND_RECOVERY_FOR_DB2 (
  PARMLIB_VERSION      210 -\* Version Number          *\
  GENERATED_JOB_REGION 6 -\* Region in Megabytes.      *\
                        -\* Valid 0 thru 256             *\
  ROUTE_ALL_ON_CONSOLE_CMDS Y -\* (Y)es/(N)o. Yes will *\
                        -\* prefix all console           *\
                        -\* commands with 'RO *ALL'      *\
                        -\* No will not.                 *\
  DASD_ALLOCATION_UNIT   SYSALDA -\* Allocation unit to be *\
                        -\* used for Dynamic             *\
                        -\* allocations.                  *\
  TEMP_DSN_ALIAS        SG7606 -\* A dataset name alias to *\
                        -\* use for creating temp        *\
                        -\* datasets.                    *\
)
-

```

```

SETUP_UTILITY (
  ABEND_ON_ERRORS          Y -\* (Y)es/(N)o - Utility will *\
                           -\* abend if any errors are *\
                           -\* encountered. *\

  USER_ABEND_RETURN_CODE  08 -\* 01-99 - Error return code *\
                           -\* if abend on errors is N or*\
                           -\* user abend code if abend *\
                           -\* on errors is Y. *\

  RELEASE_HELD_VOLUMES    Y -\* (Y)es/(N)o - Yes will *\
                           -\* release held volumes. No *\
                           -\* will produce an error and *\
                           -\* end the setup. *\

  PLACE_BKUP_VOLS_ON_HOLD Y -\* (Y)es/(N)o - Yes will *\
                           -\* cause the setup utility to*\
                           -\* place all future target *\
                           -\* volumes on hold. *\

  CLEAN_OLD_CONSIST_WINDOWS Y -\* (Y)es/(N)o - Yes will *\
                           -\* clear any non-active *\
                           -\* consistency windows. *\
                           -\* No will produce an error *\
                           -\* and end the setup. *\

  CLEAN_OLD_SNAP_SESSIONS N -\* (Y)es/(N)o - Yes will *\
                           -\* clean any non-active snap *\
                           -\* sessions from the volume. *\
                           -\* No will cause the setup *\
                           -\* utility to fail if there *\
                           -\* are old snap sessions. *\

  SYNC_ALL_BCV_GENERATIONS N -\* (Y)es/(N)o - Yes will *\
                           -\* cause the setup to *\
                           -\* establish all generations *\
                           -\* of target BCVs to their *\
                           -\* source volumes on the very*\
                           -\* first setup run of a *\
                           -\* profile. *\

  VALIDATE_DB2_VOLUME_SUBTASKS 8 -\* The number of subtasks *\
                           -\* used to get the volumes in*\
                           -\* use by DB2. *\

  VALIDATE_DB2_VOLUMES     A -\* (A)lways/(F)irst *\
                           -\* Validate DB2 source *\
                           -\* source volumes on every *\
                           -\* "SETUP" run or only the *\
                           -\* first. *\

  MAKE_READY_NOTREADY_DEVICES Y -\* (Y)es/(N)o - Yes will make*\
                           -\* not-ready devices ready. *\
                           -\* No will cause the utility *\
                           -\* to produce an error if *\
                           -\* held devices are found. *\

  MAKE_BKUP_VOLS_NOTREADY  Y -\* (Y)es/(N)o - Yes will *\
                           -\* cause setup to place *\
                           -\* future generations of *\
                           -\* target volumes in *\
                           -\* not-ready state. *\

)
-

BACKUP_UTILITY (
  ABEND_ON_ERRORS          Y -\* (Y)es/(N)o - Utility will *\
                           -\* abend if any errors are *\
                           -\* encountered. *\

  USER_ABEND_RETURN_CODE  08 -\* 01-99 - Error return code *\
                           -\* if abend on errors is N or*\

```

```

                                -\* user abend code if abend *\
                                -\* on errors is Y. *\
WAIT_FOR_VOLUME_SYNC          Y -\* (P)rompt/(Y)es/(N)o - *\
                                -\* Controls what the utility *\
                                -\* will do if any target BCVs*\
                                -\* have not synchronized to *\
                                -\* their source volumes at *\
                                -\* the time of backup. *\
                                -\* Prompt will issue a WTOR, *\
                                -\* Yes will automatically *\
                                -\* wait for sync. No will *\
                                -\* cause utility to stop an *\
                                -\* error return code. *\
PLACE_BKUP_VOLS_ON_HOLD      Y -\* (Y)es/(N)o - Yes will *\
                                -\* cause the backup utility *\
                                -\* to place backup volumes *\
                                -\* on hold. *\
RELEASE_HELD_VOLUMES         Y -\* (Y)es/(N)o - Yes will *\
                                -\* release the next *\
                                -\* generation of target vols.*\
                                -\* N will produce an error *\
                                -\* if the next generation of *\
                                -\* target volumes is held. *\
CLEAN_OLD_CONSIST_WINDOWS    Y -\* (Y)es/(N)o - Yes will *\
                                -\* clear any non-active *\
                                -\* consistency windows. *\
                                -\* No will produce an error *\
                                -\* and end the backup. *\
CLEAN_OLD_SNAP_SESSIONS      N -\* (Y)es/(N)o - Yes will *\
                                -\* clean any non-active snap *\
                                -\* sessions from the volume.*\
                                -\* No will cause the backup *\
                                -\* utility to fail if there *\
                                -\* are old snap sessions. *\
CONSIST_TIME_OUT_SECONDS     256 -\* 01-256 - Max number of *\
                                -\* seconds to suspend I/O on *\
                                -\* Standard volumes during *\
                                -\* backup. *\
BKUP_VALID_ON_CONSIST_FAIL    N -\* (Y)es/(N)o - If Yes, *\
                                -\* backup will still be *\
                                -\* registered if consistent *\
                                -\* window can not be obtained*\
                                -\* or window is closed before*\
                                -\* split/snap completed. *\
VALIDATE_DB2_VOLUMES         P -\* Prof/Always/Daily/Weekly *\
                                -\* Controls when the utility *\
                                -\* will determine the list *\
                                -\* of volumes in use by DB2. *\
                                -\* (P) will honor the profile*\
                                -\* setting. (A) for always. *\
                                -\* (D) for Daily. (W) for *\
                                -\* weekly. *\
                                -\* Only P currently supported*\
VALIDATE_DB2_VOLUMES_TIME    B -\* (B)efore/(A)fter - DB2 *\
                                -\* volume validation can be *\
                                -\* before or after the backup*\
                                -\* is taken. *\
                                -\* Only B currently supported*\
VALIDATE_DB2_VOLUME_SUBTASKS 8 -\* The number of subtasks *\
                                -\* used to get the volumes in*\

```

```

                                -\* use by DB2.                *\
WAIT_FOR_VOL_OFFLINE_SECONDS 05 -\* The nbr of seconds to wait*\
                                -\* for a volume to go offline*\
WAIT_FOR_VOL_OFFLINE_RETRIES 99 -\* The nbr of retries while *\
                                -\* waiting for a volume to go*\
                                -\* offline.                *\
MAKE_READY_NOTREADY_DEVICES Y -\* (Y)es/(N)o - Yes will make*\
                                -\* not-ready devices ready. *\
                                -\* No will cause the utility *\
                                -\* to produce an error if *\
                                -\* held devices are found. *\
MAKE_BKUP_VOLS_NOTREADY Y -\* (Y)es/(N)o - Yes will *\
                                -\* cause backup volumes to be*\
                                -\* made not-ready.        *\
RESET_COPY_PENDING_TS N -\* (Y)es/(N)o - Yes will *\
                                -\* cause all tablespaces *\
                                -\* to be reset so they are *\
                                -\* no longer in copy pending *\
                                -\* status after a system *\
                                -\* backup.                  *\
RESET_COPY_PENDING_IX N -\* (Y)es/(N)o - Yes will *\
                                -\* cause all indexspaces *\
                                -\* to be reset so they are *\
                                -\* no longer in copy pending *\
                                -\* status after a system *\
                                -\* backup.                  *\
ALLOW_SHARED_TARGET_VOLUMES N -\* (Y)es/(N)o - Yes will *\
                                -\* allow Snap or Flash target*\
                                -\* volumes to be shared among*\
                                -\* different backup profiles.*\

) -

SNAP_GLOBAL_PARAMETERS (      -\* Only for Snap Backups      *\
                                -\* "/" Indicates EMC Default *\
MAX_RETURN_CODE / -\* 4/8/12/# *\
SNAP_WAIT Y -\* (Y)es/(N)o *\
SNAP_WAIT_HOURS 00 -\* Hours on Wait Yes *\
SNAP_WAIT_MINUTES 02 -\* Minutes on Wait Yes *\
SNAP_WAIT_SECONDS 30 -\* Seconds on Wait Yes *\
MAX_ADDRSSU / -\* Maximum SNAP address space*\
MAX_TASKS1 / -\* Maximum SNAP tasks *\
MAX_TASKS2 / -\* Maximum SNAP activities *\
DEBUG_MODE / -\* All/Trace/Dump/Error/Xtra *\
DEBUG_EXTENTS / -\* Yes/No *\
TOLERATE_ENQ_FAILURES / -\* Yes/No *\
COPY_VOLUME_ID Y -\* Yes/No *\
PHASED_SNAP N -\* Use phased Snap *\
                                -\* Yes will cause each snap *\
                                -\* profile created to use *\
                                -\* the "phased snap" method.*\
                                dataset.name.of.pds -
SNAP_GROUP_PDS N -\* Wait for the background *\
WAIT_FOR_BACKGROUND_COPY -\* snaps to complete before *\
                                -\* letting the job complete *\

) -

BCV_SPLIT_PARAMETERS (      -\* Only for BCV type Backups *\
BCV_WAIT_SECONDS 06 -\* 01-99 - The nbr of seconds*\
                                -\* to wait between each check*\
                                -\* for BCV split completion *\

```

```

WAIT_RETRIES          99 -\* 01-99 The Nbr of times to *\
                        -\* check for split completion*\
                        -
)
RESTORE_UTILITY (
  ABEND_ON_ERRORS      N -\* (Y)es/(N)o - Utility will *\
                        -\* abend if any errors are *\
                        -\* encountered. *\
  USER_ABEND_RETURN_CODE 08 -\* 01-99 - Error return code *\
                        -\* if abend on errors is N or*\
                        -\* user abend code if abend *\
                        -\* on errors is Y. *\
  WAIT_FOR_VOL_OFFLINE_SECONDS 06 -\* The nbr of seconds to wait*\
                        -\* for a volume to go offline*\
  WAIT_FOR_VOL_OFFLINE_RETRIES 99 -\* The nbr of retries while *\
                        -\* waiting for a volume to go*\
                        -\* offline. *\
  WAIT_FOR_VOL_ONLINE_SECONDS 06 -\* The nbr of seconds to wait*\
                        -\* for a volume to go online *\
  WAIT_FOR_VOL_ONLINE_RETRIES 99 -\* The nbr of retries while *\
                        -\* waiting for a volume to go*\
                        -\* online. *\
  CLEAN_OLD_SNAP_SESSIONS Y -\* (Y)es/(N)o - Yes will *\
                        -\* clean any non-active snap *\
                        -\* sessions from the volume. *\
                        -\* No will cause the restore *\
                        -\* utility to fail if there *\
                        -\* are old snap sessions. *\
  FORCE_SPLIT           N -\* (Y)es/(N)o - For BCV type *\
                        -\* profiles, the current BCV *\
                        -\* generation will be split *\
                        -\* before the restore. Yes *\
                        -\* will add the force parm *\
                        -\* to the split call. *\
  PERFORM_CHECKSUM     N -\* (Y)es/(N)o - Perform check*\
                        -\* sum operation on restored *\
                        -\* volume to ensure it has *\
                        -\* not been altered since *\
                        -\* backup. *\
  DB2_UTILITY_SUITE_INSTALLED Y -\* (Y)es/(N)o - Yes means *\
                        -\* that the DB2 V8 System *\
                        -\* Restore Utility is *\
                        -\* available for recovery. *\
                        -\* No means that Recover *\
                        -\* Utility will be invoked. *\
)
OFFLOAD_UTILITY (
  ABEND_ON_ERRORS      N -\* (Y)es/(N)o - Utility will *\
                        -\* abend if any errors are *\
                        -\* encountered. *\
  USER_ABEND_RETURN_CODE 08 -\* 01-99 - Error return code *\
                        -\* if abend on errors is N or*\
                        -\* user abend code if abend *\
                        -\* on errors is Y. *\
  CLIP_PREFIX          ZZ -\* The prefix to use when *\
                        -\* clipping volsers during *\
                        -\* offload processing. *\
)

```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 312. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226
- ▶ *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370
- ▶ *DB2 9 for z/OS Technical Overview*, SG24-7330
- ▶ *DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know... and More*, SG24-6079
- ▶ *DFSMSshm Fast Replication Technical Guide*, SG24-7069
- ▶ *DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There*, SG24-7088

Other publications

These publications are also relevant as further information sources:

- ▶ *DB2 Recovery Expert for z/OS User's Guide Version 2 Release 1*, SC19-1222-00
- ▶ *DB2 Version 9.1 for z/OS Administration Guide*, SC18-9840-02
- ▶ *DB2 Version 9.1 for z/OS Utility Guide and Reference*, SC18-9855-02
- ▶ *z/OS V1R9.0 DFSMSdfp™ Storage Administration Reference*, SC26-7402-08
- ▶ *z/OS V1R9.0 MVS System Commands*, SA22-7627-16
- ▶ *Program Directory for IBM DB2 Recovery Expert for z/OS V2.1*, GI10-8764-00

Online resources

These Web sites are also relevant as further information sources:

- ▶ Recovery Expert Web site
<http://www.ibm.com/software/data/db2imstools/db2tools/db2re-zos/>

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, 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

abend 307
active log 7, 55, 152, 154, 186, 227
 data 56, 66, 154
 data set 60
 information 88
 log records 242
agent 8, 19, 27, 41, 86
 instance 8, 41
agent configuration file 41, 43–44
agent-listener-port 28
Alias, D-Dataset (A-ADD) 57, 104
allocation default 266
ALLX ALLD 233
ANTRQST macro 213
ARC0640I ARCFRTM 85, 233
Architecture 10
architecture 3, 7, 9–11
archive log 55, 153, 185–186
 copy 1 156
 copy 2 156
 fully qualified name 155
 newly created copies 159
ARY product 264
ARY#DATA DD 160, 188
ARY.BREP ORT 25, 47, 111, 160, 179, 267
ARY.DB2PARMS.CONT ROL 30, 42, 77, 102, 160, 179
ARY.OBJE CTS 29, 47, 298
ARY.OFFO PTS 25, 47, 111, 160, 179, 282
ARY.PROF ILES 25, 46, 111, 160, 179, 267
ARY.SG7606.MOVD ATA 30, 47, 299
ARY.SYSB ACK 25, 46, 111, 160, 179, 267
ARY.SYSBACK.SSID S 25, 47, 111, 160, 179, 267
ARY.V2R1 M0 25, 264
ARY.V2R1M0.SARY Load 30, 42, 77, 111, 160, 179, 300
ARY.V2R1M0.SARY Menu 25, 264
ARY.V2R1M0.SARY SAMP 25, 111, 179, 267
 library 248
 member ARYSJ002 249
ARYEMAC1 CLIST 22
ARYMOVER member 47
 sample JCL 47
ARYREPO member 47, 294
 sample JCL 47
ARYV210 CLIST 25, 47, 201
ARYV210 CLISTs
 DD name 25
automated recovery 4
Automation Tool 4–5, 12, 14, 19–20, 23, 34, 40, 126,
158, 201, 249

B

background copy 212–213, 216–218
BACKUP 6–7, 15, 51, 53–55, 174, 216–218, 224,

238–239, 241
backup xvii, 4, 11, 18, 39, 44, 51, 53–54, 84, 125–126,
153–155, 173–174, 209, 211, 215–217, 221–224, 226,
232–233, 235, 237
backup and recovery xvii, 83, 104, 153, 221, 224
backup control data set 222, 225
backup profile 6, 30, 80, 100, 134, 148, 174, 200, 309
 default value 32
 validated information 111
Backup Summary Report 112, 182
BACKUP SYSTEM 5–7, 14–15, 53, 56, 84, 139,
173–174, 201, 224–225, 227–230, 232, 234–235, 239,
241
 DATA ONLY 230–231
 FULL 225, 230, 232
BACKUP System 84, 205, 224
 backup resources 113
 I/O load 242
 job output 85
backup version 84, 221
batch job 65, 154, 179, 204
 control cars 69
 second step 189
BCDS 222, 225
BLKSIZE 160, 188
boot strap data
 set 7, 55, 85, 159, 189, 228
boot strap data set (BSDS) 85
BOTH 84, 210, 212–215, 218, 228, 234–235, 257, 261
BSDS 7, 42, 54–56, 85, 152–156, 189, 222, 224–225,
227–229, 236
 list 62, 80, 90, 154, 161
BSDS information 103, 169
BUFFERPOOL BP0 302

C

CACHE 240
CAPTURE PROFILES 249
catalog 4–5, 7–8, 23, 27, 35, 54, 57, 97, 106, 152–153,
186, 224, 227, 239, 248
 and directory 7, 61, 164, 171, 192, 224
CF 95, 120
Change Log Inventory 226, 237
CHARACTER CONVERSION Technique 26, 266
check pending 246
CHKP 246
CICS 152, 154, 172
client-listener-port 28, 41
CLIST member 31, 48
CLUSTER 69, 72, 166, 194
Cmd line
 Type D 131
 Type R 131
 Type V 131

- conditional restart 94, 96, 98, 168, 170, 195
- conditional restart control record 84, 94, 226, 234
- conditional restart control record (CRCR) 226
- configuration file 27–28, 41, 202
- Consistency Group 219–220
- control card 72, 111, 161, 184
- control data set 222, 225, 249
- control record (CRCR) 84, 163, 196, 226
- control statement 24, 42, 137, 230, 264
 - complete listing 267
 - detailed listing 43
- COPY 85, 153–155, 210, 212–214, 216–224, 228, 230–235, 238–239, 242, 255
- copy 7–8, 12, 14, 22, 24, 29, 56, 84, 125, 151, 154, 176–177, 200, 210–212, 248, 266
 - pools 56, 84, 205, 221–222
- COPY FULL
 - INDDNAME 216
 - INDYNAM 215
 - operation 214
- copy pool
 - current version 243
 - FAST REPLICATION BACKUP 234
 - FAST REPLICATION RECOVERY 99
 - new version 243
 - REPLICATION BACKUP 232
 - REPLICATION RECOVERY 238
 - RUNLIB and DSNEXIT libraries 205
 - separate ICF catalogs 225
- copy pool (CP) 56, 84, 97, 205, 221
- copy pools 56, 59
- COPYP 31, 100
- COPYPOOL 84–86, 225, 231–232, 234, 237, 241–242
- copypool 238, 243–245
- COPYTOCOPY 153, 155
- COPYVOLID parameter 214, 217
- COUNT 97, 226
- Coupling Facility 235
- CRCR 93–94, 163, 191, 226
- CRCR record 114, 168
- creation time 248
- CYCLE 215, 217

D

- DASD 53
- data consistency 154, 228, 245–246
- data set 22, 53–54, 84, 93, 126, 134, 173–174, 203, 221
 - names 177
- data set creation 228
- data sharing xvii, 5–6, 24, 39–40, 51, 54–55, 57, 84, 87, 93, 156, 185, 202, 224, 226–228, 234–236
 - environment xvii, 5, 39–40, 44, 75, 84, 93, 190, 196, 202
 - group 8, 40–42, 45, 51, 55–57, 94, 99, 117, 156, 185, 205, 224, 227, 235
 - subsystem xvii, 42, 51, 55, 57, 59, 84, 99, 165, 204, 224, 227, 235
- database xvii, 4, 8, 23, 34, 66, 84–85, 98, 126, 128–129, 204, 224–225, 248
- database copy pool 84–85, 94, 98, 224–225, 228,

- 234–236
- DATABASE Name 254
- data-partitioned secondary index (DPSI) 251
- dataset name 24, 110
 - alias 306
 - specification 178
- DATE 90, 116, 137, 152–154, 183, 227, 231–232, 255
- DB2 9 6, 21, 39, 51, 84, 129, 201, 209
 - buffer default 245
- DB2 authorization 25, 265
- DB2 BACKUP 57, 83, 232
 - System 57, 100
- DB2 Catalogue
 - VSAM clusters 192
- DB2 catalogue 54, 154, 191
 - Image copies 154
 - VSAM clusters 164
- DB2 data 4, 54, 84, 151
 - catalogue 66
 - ICF catalogs 98
 - new ICF user catalogs 62
 - non-DB2 data 100
 - set 54, 163
- DB2 data sharing 56–57
- DB2 environment 45, 51, 152, 221, 253
 - copy pools 225
- DB2 log
 - data 58
- DB2 member 94, 163, 191
- DB2 object
 - data set 56, 196
 - datasets 30
 - level 211
 - type 48
- DB2 Recovery Expert 3–4, 17, 54, 139, 151, 200, 239
 - agent 29
 - client component 27
 - configuration information 24
 - end user 20
 - GUI component 19
 - image copies 34
 - ISPF interface 32
 - load library 19
 - object 34
 - package 20
 - plan 34
 - privilege 34
 - server 19, 27
 - Services component 4
 - successful migration 32
 - System Backup 29
 - user ID authority 20
 - V1.1 11, 32
 - V2.1 1, 4
- DB2 space 163, 191
- DB2 SSID 25, 264
- DB2 Subsystem
 - DSNZPARM member 135
- DB2 subsystem 7, 15, 22, 40, 51, 54, 81, 84, 88, 127, 149, 151, 175, 203, 208–209, 248

- application table spaces 164
- displays information 7
- RBA information 37
- Sample SYSIN card 35
- SYSPITR CRCR 120
- DB2 System 106, 139, 151, 202, 224
- DB2 system
 - level backup 102, 109, 126
- DB2 table
 - space 57
- DB2 utility
 - DSNJU003 118
- DB2 V8 5, 39, 57, 84, 126, 209
 - Recovery Expert 126
- DB2 Version
 - 7 5, 18, 204
 - 8 18
 - 9 5
 - 9 product support information 18
 - 9.1 18, 139, 149, 151
- DB2 version 99, 193, 201
- DB2 z/OS 111, 182–183
 - Backup Summary Report Utility 84
 - Backup Volume Detail Report 112
 - Recovery Expert 112
 - Volume Offload Detail Report 184
- DB2PARMS DD DISP 77, 111, 160, 179
- DB9C9.SDSN Exit 85, 168, 183, 286
- DB9C9.SDSN Load 85, 160, 179, 286
- DB9CL.D9C1.BSDS 01 165, 194
- DBACRVW 20
- DBD01 85, 163, 227–230, 234–235
- DBM1 245–246
- DBRM 162, 251, 259
- DBRM Name 259
- DD DISP 85, 160, 179
- DD SYSOUT 94, 160, 179, 300
- D-Dataset Display 57, 104
- DDL 12, 22–23, 48, 248
- DDL member 34
- dependent object 5
- dependent writes 219–220
- DFSMS 66, 211, 221–222
- DFSMSdss 66, 153, 182, 184, 213–214
 - COPYVOLID 215
 - FCNOCOPY 217
 - FCWITHDRAW 217
- DFSMSHsm 66, 84, 173–174, 221–222, 224–225, 227–231, 233–235, 238, 243
- DFSMSHsm Fast Replication
 - associate 243
 - backup 227
 - function 221
 - Technical Guide 221
- DGSF 300
- directory 7, 54, 56, 61, 152–153, 192, 224, 227, 239
- disaster 81, 149, 151–154, 185, 242
- disaster recovery xvii, 7, 10, 12, 48, 51, 81–82, 149, 151–153, 155, 185, 193, 213, 217, 228, 242
 - proper settings 170

- Disaster Recovery (DR) 20, 44, 48, 81, 149, 155, 180, 201, 209
- Disaster Recovery Profile
 - Disaster Recovery Method 185
- Disaster recovery profile 101, 158, 180, 185
- disaster recovery solution 149
- disk space (DS) 240
- dropped object 4, 204, 248, 253
- DSN
 - strict naming convention 225
- DSN PREFIX maximum length 265
- DSN1COPY 13
- DSN1LOGP 227
- DSN1PRNT 227
- DSNJ139I 154–155
- DSNJCNVB 195
- DSNJU003 84, 94, 155, 163, 168, 191, 226, 234–235, 237
- DSNJU004 85, 90, 154–155, 163, 169, 191, 226
- DSNTIP6 239–240
- DSNU1602I 232
- DSNUTILB 98, 120, 163, 191
- DSNUVBBD - BACKUP (DB) 85, 225
- DSNZPARM 20, 25, 103, 196, 239
- DUMMY Record 294
- DUMP 215, 241
- dump classes 241
- DUMP FULL 216, 218
- DUMPCOND FR 85, 233
- DUMPCONDITONING 215–217
- DUMPCONDITONING parameter 215
- DUMPNLY 241

E

- ELB 219
- EMCLOAD1 GT 305
- end command 64, 101, 135, 157, 176
- ENDING AT 86, 233–234
- ENDLRN 168, 226
- ENDRBA 226
- Enterprise Storage Server 84, 209
- environment 207
 - data sharing 5, 39–40, 85, 196, 202
 - non-data sharing 5, 39–40, 45
- environment configuration 40
- ESS Copy
 - Service 210
 - Services Web user interface 211
- ESS FlashCopy 221
- EXISTS 211, 217, 225
- EXPIRATION Date 177, 267
- Extended Long Busy 219
- extended long busy (ELB) 220

F

- FACILITY class 19
- Fast Log Apply 94, 234
- fast log apply
 - function 245

- fast replication 7, 24, 84, 98, 221–222, 224, 227, 230–233, 238
- fast replication services 227
- FCNOCOPY parameter 217
- FCWITHDRAW parameter 217
- File tailoring work (FTW) 266
- FLA 234
- FlashCopy 6, 18, 51, 126, 210, 212–215, 217–221, 223, 228, 238, 242
 - background copy 211–212
 - DFSMSdss 211, 215
 - persistent relationship 213, 218
- FlashCopy relationship 210
 - source volume 212
- FlashCopy target 214
- FlashCopy V1 18, 210
- FlashCopy V2 210
- FlashCopy Version
 - 2 242
- FRBACKUP 221, 224, 232–233, 243
- FRDELETE 221
- freeze
 - FlashCopy consistency group 219
- FRRECOV 84, 221, 234
- full image copy 139
- full system backup 18, 228–230, 234
- FUNCTION RC 85, 232

G

- GDG base 31, 47, 300
- GDG dataset 30
- generated recovery plan
 - Relative cost evaluation 20
- generated SQL 206
- generations 243
- geographically distributed 9
- grants 23
- granularity 4
- graphical user interface (GUI) 1, 4, 18, 85, 253
- Grouper 23, 25, 34, 265
- GUI (graphical user interface) 4
- GUI client 8, 27, 200

H

- header page 85, 227
- high-level qualifier 19, 54, 68
- host name 41, 43–44

I

- I/O activity 219
- IBM DB2 Recovery
 - Expert 3, 18, 40, 80, 99, 266
 - Expert server 49
- ICF catalog 97, 154, 162
- ICF Catalogs Volume 97
- ICF user catalogue 7, 53–54, 100
- IDC0001I Function 194
- IDC0550I Entry 194

- IDCAMS 24, 65, 163, 191
- IDCAMS input 65
- IDCAMS step 163, 191
- Image copy 34, 81, 134, 151, 157, 185–186, 203, 239
 - Option 157, 186
 - output data 270
- image copy 239, 246
 - catalog entries 186
 - Catalog x days 157
 - DB2 objects 171
- imagecopy 138
- IMS xvii, 152, 154, 172
- incommit 154
- Incremental FlashCopy 218
- incremental FlashCopy 218, 238, 243–244
- incremental image copy 243
- indoubt URs 235
- in-flight 96, 245
- inflight 84, 154, 234
- insert 163, 191
- INSTALLATION Exit 85, 233
- installed SBRS
 - product control file 48
- integrated catalog facility (ICF) 153
- Interactive Storage Management Facility 224
- Interactive System Management Facility (ISMF) 222
- Internal Resource Lock Manager (IRLM) 4
- interproduct communication 265
- IRLM (Internal Resource Lock Manager) 4
- is defined (ID) 19, 127
- ISMF 222, 224
- ISP 216
- ISPEXEC LIBDEF 305
 - ISPTABL 305
- ISPF component 20, 39, 45, 200
- ISPF interface 1, 6, 11, 30, 81, 100, 125, 147, 189, 200, 207
 - DB2 Recovery Expert 32
 - LOB table spaces 202
 - new features 32
 - object level recovery 147
- ISPLLIB (ID) 265

J

- JCL (job control language) 4
- job ARYSJ001 32, 42
- Job Control Language (JCL) 21, 47, 57, 85, 133, 158, 183, 203, 249
- job control language (JCL) 4
- job D9C1 164, 192
- JOBPARM SYSAFF 77, 110, 160, 183

L

- LABEL FILE Number 267
- last analysis 57, 104
- launchpad 87
- LIMIT BACKOUT 149
- Line Cmds 57, 104
- line command 57, 130

- list 223, 239
- LISTDEF 239
- LOAD 22, 25, 239, 264
- LOB 251
- local site 135, 149, 153, 155, 174, 217
 - DR DSNZPARMS member 170
- location name 66, 225, 227
- log analysis 4, 13, 15, 204, 252, 265
- log data sets 7, 54, 58, 60, 154, 194
- LOG NO events 94, 234
- log RBA 204
- log record sequence number (LRSN) 98, 135, 246
- Log truncation point 84, 226, 234–237
- log truncation point 94, 226
- LOGAPPLY 245–246
- logical partition (LPAR) 8
- LOGONLY option 191, 236
- logs 7, 53–55, 84–85, 147, 151, 185, 205, 221, 224
- LPAR 8, 10, 22–23, 40–41, 45, 77, 157, 201
- LPAR (logical partition) 8
- LPAR SC59 43
 - Recovery Expert Agent 59 43
 - single non-data sharing environment 46
- LPARs
 - multiple subsystems 27
- LPARs SC53 41
 - DB2 subsystems 41
 - Recovery Expert Agent 59 49
 - sysplex environment 46
- LRSN 6, 12, 90, 96, 125, 135, 225–228, 230
- LRSN (log record sequence number) 246

M

- main features 1, 3–5
- maintenance 207
- master 65
- member ARY 31
- members D9C1 41
- messages 99, 170, 232, 237–238
- metadata structure 212
- M-Move BSDS 59, 105
- MODIFY 97
- MVS user catalogue 54, 57, 104

N

- nbr 181, 309
- next substeps 167
- NFM 5, 226, 228
- non-data sharing
 - environment 5, 39, 45–46
 - subsystem 43
- non-DB2 data 58, 100, 104
- non-partitioned secondary index (NPSI) 251
- NOVTOCENQ 226

O

- OA06196 214
- OA17314 243

- Object 5, 13, 18, 20, 75, 81, 101, 112, 126, 180, 238–239
- object definition 8, 248
- object definition level 248, 254
- object dependencies 4
- Object Profile 14, 20, 30, 81, 126, 201
- object profiles 4, 12, 14, 19, 40, 81, 251
- Object Recovery 6, 12, 126, 200
- Object-level recovery 29, 239
- object-level recovery
 - recovery base 29
- offload option 108, 174, 205
- Offload process 184
- OMVS segment 19
- Online REORG 228
- options 4–6, 8, 13–14, 30, 65, 69, 72, 83, 85, 125, 130–131, 153, 158, 173, 201, 218, 228, 237, 240, 250, 264
- ORDER 153–154, 211, 214, 218, 220, 224
- OW57347 214

P

- package 251
- parmlib member 49
- PARTITION 95–96
- partition by growth (PBG) 206
- partitioning index 251
- PDS 23, 93, 134, 158, 189, 264
- PDS member 42, 300
- performance xvii, 23, 207, 224, 241
 - fast log apply function 245
- PERMIT 19
- persistent
 - FlashCopy relationship 211, 213
- PIT (point-in-time) 4
- PITR 227
- PK23288 147, 208
- PK29628 208
- PK49366 208
- PK51979 84, 98, 208, 234
- PK53379 207
- PK55713 207–208
- PK55821 207
- PK57092 207
- PK57163 207
- PK57183 207
- PK58428 207
- PK58430 207
- PK58431 207
- PK58433 207
- PK58434 207
- PK58928 207
- PK58945 207
- PK59664 208
- PK59666 208
- PK59675 120, 196, 208
- PK60078 207
- PK60696 208
- PK60716 208
- PK60786 208
- PK60788 208

- PK61814 208
- PK61889 207–208
- PK62088 208
- PK62568 208
- PK62742 208
- PK62922 208
- PK62974 208
- PK63692 208
- plans 13, 20, 23–24, 40, 92, 252, 264
- Point-In-Time 51, 210, 218–219, 224
- point-in-time
 - recovery 221, 245
- point-in-time (PIT) 4, 210, 248
- point-in-time recovery 226, 254
 - point 245
- port number 41, 43–44
- PPRC 210, 214
- prefix generation 264
- Press PF3 128
- PRIMARY QTY 26, 266
- Print Log Map 226
- Print Log Map utility 153
- prior point in time 227, 235
- privileges 20, 23, 251
- PROC 0
 - CLISTLI1 303
 - RBRLVL 304
- PRODUCT CFG 25, 264
- product control file 24, 41–42, 44, 200, 202, 279
- product libraries 249
- PRODUCT VER 25, 264
- Profile Name
 - D9C1 107
 - field 131
- pruning SLR 253

Q

- QT OWNER/NAME 265
- QUALIFIER 255, 257
- QUIESCE 154, 227, 245–246
- quiesce point 81, 139, 147–148, 154, 253–254
- QUIET TIME
 - REPORT/CAPTURE JCL 269
- quiet time 4–5, 204, 245, 250, 254, 257, 265

R

- RACF 19, 233–234
- RACF profile 19
- RALTER Program 19
- RBA 6, 12, 88, 91, 125, 135, 155, 181, 222, 225–226, 228, 230, 246
- RBA capture utility 32, 47, 200, 299
 - Sample job 300
- RBA information 32, 47, 204, 299
- RBA/LRSN 6, 44, 98, 112, 114, 135, 137–138, 181, 200
- RBDP 94, 234–235
- RBLP 85, 227–228, 230, 234–235
- RBR Device 267
- read-write (RW) 163, 191

- Real 239
- REBUILD 136, 171, 227, 237
 - INDEX 136, 237
 - pending 237
- REBUILD INDEX 136, 237
- RECOVER 13, 25, 29, 51, 96, 136, 139, 151, 153, 162–163, 171, 200, 221–222, 224, 226–227, 234, 236–237, 239, 242, 245–246, 255, 265
- recovery xvii, 4, 12, 18, 20, 39–40, 44, 81, 83–84, 125, 149, 151, 200, 210, 217, 248
 - function 4, 8, 224, 245
 - history events 204
 - method 4, 81, 109, 154, 242
 - object xvii, 4, 12, 20, 125–126, 158, 170, 201, 239
 - objects to current 81
 - plan 13, 153, 204
 - plan generation 204
 - point 4, 47, 81, 85, 139, 154, 181, 200, 221, 248
 - process 138–139, 158, 185, 224, 235
 - scenario 126, 148, 153
 - type 13, 136, 221
- recovery action 248
- recovery base log point 227
- recovery base log point (RBLP) 85
- Recovery Expert 3–4, 17, 39, 51, 54, 125, 149, 173, 199, 247
 - agent 41
 - Agent 63 configuration file 43
 - Agent 63 reference 43
 - Agent 64 configuration file 44
 - Agent 64 reference 44
 - backup 99
 - BACKUP SYSTEM 112
 - client 203
 - component 56
 - configuration 40
 - control 24
 - control file 35
 - DB2 object 23
 - GUI 122
 - GUI client 40
 - GUI portion 99
 - ISPF components 48
 - job 120
 - JOB1 94
 - object 23
 - Object recovery 12
 - offload 24
 - perspective 58
 - privilege 34
 - product 85
 - product control 24
 - product control file 43, 46
 - program 163, 191
 - program ARy 191
 - server 7, 41, 44, 206
 - SLBRS Repository 116
 - store 80, 104
 - system backup 12
 - System Level 94, 121

- System Level Backup 100
- tool 44, 255
- v1.1 32, 205
- V2 Client/Server component 204
- V2.1 4, 20, 126, 207
- verifie 148
- VSAM file 191
- Recovery Expert for z/OS
 - agent 8, 27
 - environment configuration 40
 - GUI client 14, 27, 254
 - main features 4
- Recovery Expert Server 7, 41
- recovery option 13, 126, 201
- Recovery plan 13, 267
- recovery point 15, 113, 135, 181, 224
 - last copy 239
- recovery site 135, 151, 176
 - archive logs 154
 - DB2 spaces 164
 - image copies 164
 - necessary settings 170
- RECP 94, 234–235
- Redbooks Web site 312
 - Contact us xix
- redo
 - SQL 13, 204
- REGION 42–43, 76–77, 111, 118, 133, 160, 179
- relative byte address (RBA) 91, 137
- remote copy 149
- remote recovery 173
- Remote site 4, 7, 81, 152, 158, 173, 201, 210
- remote site
 - archive logs 162
 - DB2 knowledge 152
 - DB2 subsystem 159
 - DSNZPARMs setting 196
 - recovery 5, 149, 155
 - recovery service 7
- rename 56, 59, 131
- REORG 228, 239, 253
- REPLY Y 96, 191
- REPORT 25–26, 42, 191, 265
- report 30, 103, 113, 154–155, 183
- repository name 265
- REPRO 69, 72, 163, 191
- requirements 211, 243
- RESET 219, 226, 235, 237
- RESET_COPY_PENDING_IX 100, 309
- RESET_COPY_PENDING_TS 100, 309
- RESTART 96, 169, 191, 227
- restart 84, 88, 94, 154, 164, 168, 226, 228, 234, 237
- RESTORE 6–7, 29, 51, 53–54, 152–153, 155, 163, 181, 215–217, 227, 237–238, 240, 242, 255
- RESTORE SYSTEM 4–6, 14, 53–54, 84–85, 173, 191, 196, 201, 208, 224–226, 228, 234–238, 240
 - Example 237
 - Execution 237
 - LOGONLY 120, 234–236
 - Operation 228

- RESTOREBEFORE 239, 246
- RETENTION Period 177, 267
- RETURN Code 85, 160, 232, 249
- Row 1 57, 104, 127, 175
- RUNSTATS 240, 253
- RW mode 170, 196

S

- SAMP 42, 267
- Sample JCL 47, 249, 299
- Sample job 263
- Sample statement 264
- saved specification 252
- SBOX7I D807 112, 182
- SBOX7J D907 112, 182
- SBRS 10, 80
- SBRS CLIST 36
- SBRS repository 10, 46, 113
- SCA 85, 95, 228
- scenarios xviii, 4, 40, 46, 125–126, 234, 238, 263
- SCFDD GT 305
- SCHEMA 25, 254
- schema 8, 15, 27, 34–35, 51, 247, 265
 - name 248
- Schema Level
 - Repository 27, 43, 201, 248
- schema level 264
 - repository data 272
- schema level repository
 - data 268
- Schema Level Repository (SLR) 43
- schema level repository (SLR) 5, 8, 247–249
 - database 253
 - maintenance 253
 - objects 248
 - tables 248, 250, 254
 - update 248–249, 254
- SCRATCH Limit 300
- SDSNEXIT 19, 160, 179
- SECONDARY QTY 26, 266
- server-address 27, 35, 41, 43
- server-address space 37
- Services CLIST 48
- Services subsystem (SS) 32
- SET 25, 65, 85–86, 154, 156, 165, 186, 210, 214–215, 218, 220–223, 225, 227–228, 233
- SET LOG RESUME 227, 235
- SET LOG SUSPEND 227, 235
- SEVERAL OTHERS (SO) 254
- Share Option 108, 130, 157, 175
- Shared Communications Area (SCA) 95
- shared DASD
 - product control file 48
- SHR,LABEL (SL) 216
- Single LPAR 10, 26
- single LPAR
 - managing multiple DB2 subsystems 26
- site failures 4
- SLR 5, 8
- SLR (schema level repository) 8

- SLR table 27, 250
 - PACKAGES information 259
 - referential constraint information 260
- SLR update
 - performance 208
 - utility 253
- SMS DATA Class 266
- SMS MANAGEMENT Class 266
- SMS STORAGE Class 55, 68, 266
- SORTDEVT 171
- source volume 6, 80, 112, 182, 211, 307
 - exact size 223
 - identical copy 214
 - large number 219
 - volume serial number 214
- specifications 80, 157, 176, 261
- SSID 22, 25, 42, 57, 59, 103–104, 127, 156, 175, 225, 249
- START DATABASE 226
- START DB2 84, 196
- STARTRBA 226
- STOGROUP Name 257
- STOGROUP STGX 302
- Stop DB2 235
- storage group 40, 56, 221–222, 252, 301
 - new type 222
- Storage Management Subsystem (SMS) 84
- Structured Query Language (SQL) 253
- structures 95, 196, 218–219, 235
- subsystem xvii, 4–6, 12, 14, 18–19, 41–43, 51, 53, 84, 87, 126, 151, 173–174, 210, 217, 224, 248, 264
 - data sharing 42, 55, 60, 94, 113, 205
 - recovery 44, 51, 81, 113, 126, 135, 151, 159, 181, 217, 224, 234, 245
- suspend 227–228, 235
- synchronizing data and DDL 253
- synonyms 251
- SYSADM 20, 98, 196
- SYSCOPY 136, 163, 239, 251, 255
- SYSIN 24, 42, 77, 165, 215–218, 230, 232, 236–237
- SYSIN card 24
- SYSPIR 93–94, 195, 225–226, 237
- SYSPIR CRCR 84, 94, 98, 226, 234–236
- SYSPIR LRSN 118
- sysplex 9, 41
- SYSPRINT 42–43, 77, 94, 98, 165, 215–218
- SYSPRINT DD SYSOUT 78, 118, 166, 300
- SYSPROC library 22
- System Backup
 - local site 163
- system backup 5, 21, 24, 46, 56, 80, 100, 133, 156, 173, 200, 228, 267
- System Backup and Restore Services 53, 80, 99–100
- system backup and restore services 8
- system catalog 4–5, 8, 23, 35, 75, 171, 249
- System checkpoints 228
- SYSTEM JCL 85
- SYSTEM job 86, 230
- system level point in time recovery 226, 236
 - RESTORE SYSTEM 236

- system services 8
- SYSTEM Utility 4, 53, 84, 173, 208, 224
- system-level backup 29, 81, 139, 239

T

- table
 - creation 228
- TABLE Creator 256
- TABLE Name 256
- table space 57, 140, 170, 196, 202
- TABLESPACE Name 254
- tablespaces 128
- target volume 7, 80, 175, 204, 210, 307
 - not-yet-copied track 213
 - VTOC index names 215
- TASKID 001 85, 233
- TASKID 002 85, 233
- TCP/IP 8
- TEXT 207–208
- TIME 84, 95, 137, 152–155, 183, 210, 212–214, 218–219, 222, 224–225, 227, 230–231, 234–237, 257
- Time of Day (TOD) 225
- TIMESTAMP 96, 258
- timestamp 6, 90, 200, 254
- TOKEN 85–86, 224–225, 231, 237
- tracker site 149

U

- UCAT.DB8G Data 63
- UCAT.DB8G Log 55
- UCAT.DB9C Data 97
- UCAT.TOTD DL 57
- UDF (user-defined function) 253
- UK30484 207
- UK31997 98, 208
- UK32857 207–208
- UK33508 207
- UK33601 207
- UK34785 208
- UK35022 208
- UK35487 208
- UK35488 208
- UK35495 208
- undo 204
- UNICODE UT-16 26, 266
- UNIQUE 222, 225
- units of recovery (UR) 245
- UNIX System Services 19
- UPDATE IPC IPC_RBR 12, 24, 203, 267
- UPDATE LAS DATA AUNIT 26, 266
- UPDATE LAS DATA PQTY 26, 266
- UPDATE LAS DATA SQTY 26, 266
- UPDATE LAS INDEX AUNIT 26, 266
- UPDATE LAS INDEX PQTY 26, 266
- UPDATE LAS INDEX SQTY 26, 266
- UPDATE LAS VOLSERS 26, 266
- UPDATE RBR
 - BKUP OBJS 25, 267
 - BKUP SSID 26, 267

- BKUP VOLS 25, 267
- CATS REPO 25, 267
- Device 24
- EMC LOAD1 24, 267
- EMC LOAD2 24, 267
- EMC LOAD3 24, 267
- EMC LOAD4 24, 267
- FDR LOAD1 24, 267
- FDR LOAD2 24, 267
- MAPS REPO 25, 267
- OFFLD REPO 25, 267
- PARMLIB MBR 25, 267
- PROF REPO 25, 267
- REPORT REPO 267
- SYSBK REPO 25, 267
- uppercase 28
- USAGE 252
- User ID 20, 178
- user id 19, 131, 170, 196, 265
- user-defined function (UDF) 252–253
- Utility
 - RESTART 226

V

- Valid Primary Cmds 57, 104
- validate 103, 109, 111, 224
 - option 109
- Version 4–5, 18, 40–41, 45, 84, 111, 151, 182, 210–211, 221, 231, 235–237, 243
- version 5, 12, 19, 23, 39, 85, 98, 200, 222, 228, 236–237, 248, 254
- Version 02.01.013 112, 184
 - 17
 - 42 ARYS162I 184
 - ARYS162I 111
- Version 1.1 32
- version information 228, 230
- versioning 4
- versions 210, 242–243
- view 91, 103, 113, 127, 130–131, 202, 245–246, 251
- Virtual Storage Access Method (VSAM) 164, 249
- Volser Ucb 112, 182
- volsers 266
- VOLSERS statement 269
- volume DB8GL1 55
- volume dump 153, 215, 217, 235
- volume level backups 51, 224
- VOLUME SBOX5E
 - IDC0508I DATA ALLOCATION STATUS 195
 - IDC0509I INDEX ALLOCATION STATUS 195
- Volume SBOX5E 195
- Volume TOTDDL 75
- VSAM data 24, 265
 - set 32
- VSAM data set 32
- VSAM file 24, 104, 168, 191
 - 4089-byte records 168
- VSAM repository
 - datasets 300

- file 186, 201
- V-View Alias 57, 104
- V-Volume List 59, 105

X

- XRC 214

Z

- z/OS xvii, 3, 17, 54, 62, 65, 84, 151, 182, 209, 211, 221, 224, 227, 235, 237–238, 243
- z/OS component 266
- z/OS User 5, 18, 81, 200
 - DB2 Recovery Expert 203
 - IBM DB2 Recovery Expert 81
 - manual DB2 Recovery Expert 18
- z/OS V1R3 235
- zSeries 210, 214, 221

Archived



Optimizing Restore and Recovery Solutions with DB2 Recovery Expert for z/OS V2.1



Optimizing Restore and Recovery Solutions with DB2 Recovery Expert for z/OS V2.1



Manage and validate backup and restore system

Optimize system and object level restore from volume copies

Ensure business continuity with cost-effective recovery

IBM DB2 Recovery Expert manages the recovery of DB2 for z/OS V8 and V9 objects in a data sharing and non-data sharing environment. The major focus of IBM DB2 Recovery Expert V2.1 for z/OS (5697-N92) is on DB2 system level backup and restore. The tool provides the ability to create a consistent full system backup with no impact on DB2 availability, comprehensive backup and recovery reporting capabilities, and robust database and storage validity checking to ensure successful creation and usage of database system backups. By integrating with disk storage functions, it supports DB2 subsystem recovery from system backup, DB2 subsystem disaster recovery, and DB2 object data recovery from system backups. It also helps in selecting the most convenient recovery solution and in managing backups.

This IBM Redbooks publication documents how to use Recovery Expert V2.1 for all restore and recovery functions related to system level backup. For scenarios of using Recovery Expert functions when recovering DB2 objects from traditional image copies, see the previous companion IBM Redbooks publication *IBM DB2 Recovery Expert for z/OS User Scenarios*, SG24-7226.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7606-00

ISBN 0738485365