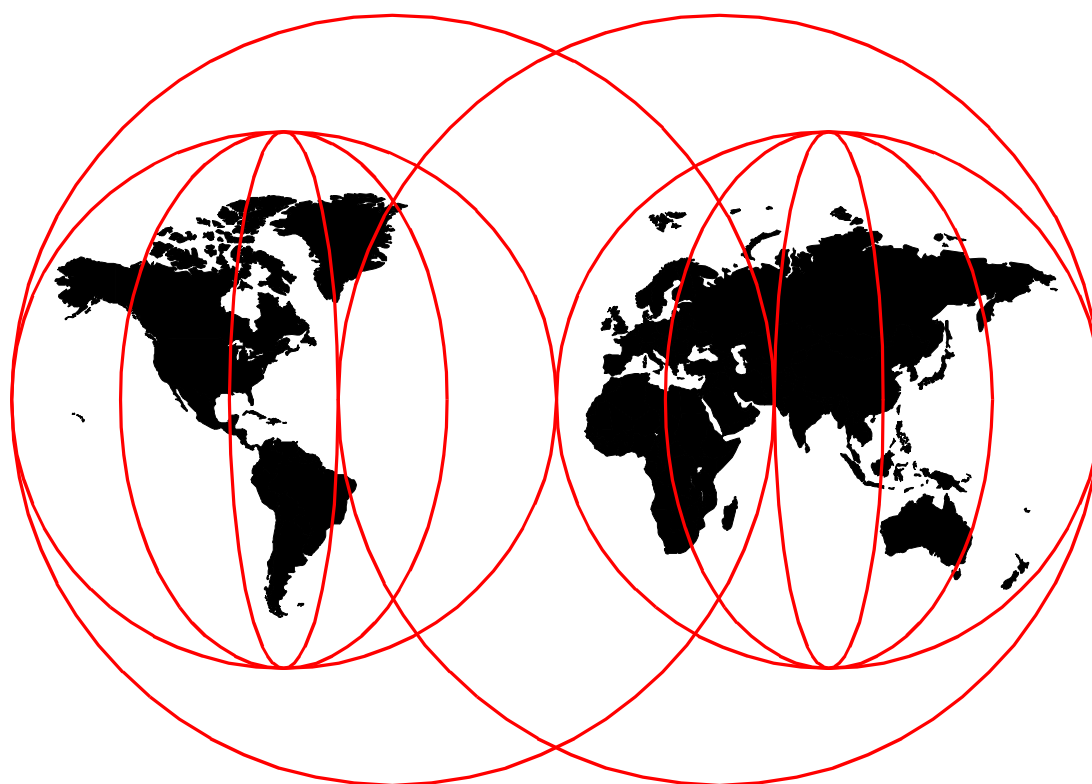




Storage Management with DB2 for OS/390

Paolo Bruni, Hans Duerr, Daniel Leplaideur, Steve Wintle



International Technical Support Organization

www.redbooks.ibm.com



International Technical Support Organization

SG24-5462-00

Storage Management with DB2 for OS/390

September 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix F, "Special Notices" on page 239.

First Edition (September 1999)

This edition applies to Version 5 of DB2 for OS/390, Program Number 5655-DB2, and Version 1 Release 4 of DFSMS/MVS, Program Number 5695-DF1, unless otherwise stated.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures	xi
Tables	xv
Preface	xvii
The Team That Wrote This Redbook	xvii
Comments Welcome	xix
<hr/>	
Part 1. Introduction and Summary	1
Chapter 1. Introduction	3
Chapter 2. Summary of Considerations	5
2.1 DB2 and Storage Management	5
2.1.1 Benefits of DFSMS	5
2.1.2 Managing DB2 Data Sets with DFSMS	6
2.1.3 Examples for Managing DB2 Data Sets with DFSMS	6
2.2 DB2 and Storage Servers	6
2.2.1 Data Placement	6
2.2.2 Large Cache	6
2.2.3 Log Structured File	7
2.2.4 RAMAC Architecture	7
2.2.5 SMS Storage Groups	7
2.2.6 Performance Management	8
<hr/>	
Part 2. DB2 and System Managed Storage	9
Chapter 3. DB2 Storage Objects	11
3.1 DB2 Overview	11
3.2 DB2 Data Objects	11
3.2.1 TABLE	12
3.2.2 TABLESPACE	12
3.2.3 INDEX	13
3.2.4 INDEXSPACE	13
3.2.5 DATABASE	13
3.2.6 STOGROUP	13
3.3 Creating Table Spaces and Index Spaces	13
3.3.1 DB2 Defined and Managed	14
3.3.2 User Defined and Managed	14
3.4 DB2 System Table Spaces	15
3.4.1 The DB2 Catalog and Directory	15
3.4.2 The Work Database	15
3.4.3 SYSIBM.SYSCOPY	16
3.4.4 SYSIBM.SYSLGRNX	16
3.5 DB2 Application Table Spaces	17
3.6 DB2 Recovery Data Sets	17
3.6.1 Bootstrap Data Sets	17
3.6.2 Active Logs	18
3.6.3 Archive Logs	19
3.6.4 Image Copies	20
3.6.5 Other Copies	22

3.7	Other DB2 Data Sets	22
3.7.1	DB2 Library Data Sets	22
3.7.2	DB2 Temporary Data Sets	22
3.8	DB2 Data Sets Naming Conventions	22
3.8.1	Table Space and Index Space Names	23
3.8.2	BSDS Names	23
3.8.3	Active Log Names	23
3.8.4	Archive Log and BSDS Backup Names	24
3.8.5	Image Copy Names	24
	Chapter 4. System Managed Storage Concepts and Components	25
4.1	Background	25
4.2	Evolution	25
4.3	DFSMS/MVS Components	26
4.3.1	DFSMSdfp	26
4.3.1.1	ISMF for the End User	26
4.3.1.2	ISMF for the Storage Administrator	27
4.3.2	DFSMSdss	27
4.3.2.1	Functionality	27
4.3.2.2	Filtering	28
4.3.2.3	Converting Data to SMS	28
4.3.3	DFSMSHsm	28
4.3.3.1	Space Management	29
4.3.3.2	Availability Management	30
4.3.4	DFSMSrmm	31
4.3.5	DFSMSopt	31
4.3.6	SMF Records 42(6)	32
4.4	Benefits	32
	Chapter 5. Storage Management with DFSMS	35
5.1	Introduction	35
5.1.1	Base Configuration	35
5.1.2	Class and Storage Group Definitions	36
5.2	Automatic Class Selection Routines	36
5.3	SMS Classes	38
5.3.1	Data Class	38
5.3.1.1	Description	38
5.3.1.2	Planning for Implementation	39
5.3.2	Storage Class	39
5.3.2.1	Description	39
5.3.2.2	Planning for Implementation	40
5.3.3	Management Class	41
5.3.3.1	Description	41
5.3.3.2	Planning for Implementation	41
5.3.4	Storage Group	43
5.3.4.1	Description	43
5.3.4.2	Planning for Implementation	44
5.3.4.3	Mapping Devices to Storage Groups for Performance	45
5.4	Naming Standards	46
5.5	Examples	46
	Chapter 6. Managing DB2 Databases with SMS	47
6.1	SMS Examples for DB2 Databases	47
6.1.1	Using ISMF to Display SMS Constructs	47

6.1.2	SMS Data Class	47
6.1.3	SMS Storage Class	48
6.1.4	SMS Management Class	49
6.1.5	SMS Storage Groups	50
6.1.6	DB2 STOGROUPs and SMS Storage Groups	52
6.1.7	Assigning SMS Classes to DB2 Table Spaces and Index Spaces	53
6.1.8	Table Space and Index Space Names for SMS	56
6.1.9	Managing Partitioned Table Spaces with SMS	56
6.2	User Databases	57
6.2.1	Online Production Databases	58
6.2.1.1	Storage Classes	58
6.2.1.2	Management Classes	58
6.2.2	Batch Production Databases	58
6.2.2.1	Storage Classes	58
6.2.2.2	Management Classes	58
6.2.3	Data Warehouse Databases	58
6.2.3.1	Storage Classes	59
6.2.3.2	Management Classes	59
6.2.4	Development and Test Databases	59
6.2.4.1	Storage Classes	59
6.2.4.2	Management Classes	59
6.2.5	Summary	59
6.3	DB2 System Databases	60
6.3.1	Catalog and Directory Databases	60
6.3.1.1	Storage Classes	61
6.3.1.2	Management Classes	61
6.3.2	Work Database	61
6.3.2.1	Storage Classes	61
6.3.2.2	Management Classes	61
6.3.3	Summary	61
Chapter 7. Managing DB2 Recovery Data Sets with SMS		63
7.1	SMS Examples for DB2 Recovery Data Sets	63
7.1.1	SMS Data Class	63
7.1.2	SMS Storage Class	63
7.1.3	SMS Management Class	64
7.1.4	SMS Storage Groups	65
7.1.5	Assigning SMS Classes to DB2 Recovery Data Sets	66
7.2	BSDS	67
7.2.1	Storage Class	67
7.2.2	Management Class	67
7.2.3	Storage Group	68
7.2.4	ACS Example	68
7.3	Active Logs	68
7.3.1	Storage Class	69
7.3.2	Management Class	69
7.3.3	Storage Group	69
7.3.4	ACS Example	69
7.4	Archive Logs	69
7.4.1	Storage Class	71
7.4.2	Management Class	71
7.4.3	Storage Group	71
7.4.4	ACS Example	71
7.5	Image Copies	71

7.5.1 Storage Class	72
7.5.2 Management Class	72
7.5.3 Storage Group	73
7.6 Summary	73
Chapter 8. Converting DB2 to Systems Managed Storage	75
8.1 Overview	75
8.2 Advantages of SMS Managing DB2 Data	75
8.3 SMS Management Goals	76
8.4 Positioning for Implementation	77
8.4.1 Prerequisite Planning	77
8.4.2 Service Level Agreement	78
8.5 Conversion Process	78
8.5.1 Sequence	78
8.5.2 Methodology	79
8.5.2.1 Conversion Window	79
8.5.2.2 Data Movement	79
8.5.2.3 Tailor Online Conversion	79
8.5.2.4 Contingency Time Frame	80
8.5.3 SMS Implementation	80
8.5.4 Post Implementation	81
8.6 DFSMS FIT	81
8.7 NaviQuest	81

Part 3. DB2 and Storage Servers 83

Chapter 9. Disk Environment Overview	85
9.1 Evolution of Disk Architecture	85
9.1.1 3380 and 3390 Volumes	85
9.1.2 Arrays	85
9.1.3 Log Structured File and SnapShot	86
9.1.4 Virtual Volumes	88
9.2 Disk Control Units	89
9.2.1 Storage Server	89
9.2.2 Storage Devices	90
9.2.3 Logical Control Unit	90
9.3 Cache Management	90
9.3.1 Track Caching	91
9.3.2 Read Record Caching	91
9.3.3 Write Record Caching (Quickwrite)	92
9.3.4 Sequential Caching	92
9.3.5 No Caching—Bypass Cache	92
9.3.6 No Caching—Inhibit Cache Load	92
9.3.7 DB2 Cache Parameters (DSNTIPE)	92
9.3.8 Dynamic Cache Management Enhancement	92
9.4 Paths and Bandwidth Evolution	93
9.5 Capabilities	93
9.5.1 Dual Copy	93
9.5.2 Concurrent Copy	94
9.5.3 Virtual Concurrent Copy	95
9.5.4 Remote Copy	95
9.5.4.1 PPRC	96
9.5.4.2 Geographically Dispersed Parallel Sysplex	97

9.5.4.3 Extended Remote Copy	99
9.5.5 Compression	100
9.5.6 Sequential Data Striping	101
Chapter 10. DB2 I/O Operations	103
10.1 Avoiding I/O Operations	103
10.2 Data Read Operations	104
10.2.1 Normal Read	104
10.2.2 Sequential Prefetch	104
10.2.3 Dynamic Prefetch	105
10.2.4 List Prefetch	105
10.2.5 Prefetch Quantity	105
10.2.6 Data Management Threshold	106
10.2.7 Sequential Prefetch Threshold	107
10.3 Data Write Operations	107
10.3.1 Asynchronous Writes	107
10.3.2 Synchronous Writes	108
10.3.3 Immediate Write Threshold	108
10.3.4 Write Quantity	108
10.3.5 Tuning Write Frequency	108
10.4 Log Writes	111
10.4.1 Asynchronous Writes	112
10.4.2 Synchronous Writes	112
10.4.3 Writing to Two Logs	112
10.4.4 Two-Phase Commit Log Writes	112
10.4.5 Improving Log Write Performance	114
10.5 Log Reads	115
10.5.1 Improving Log Read Performance	116
10.5.2 Active Log Size	117
Chapter 11. I/O Performance and Monitoring Tools	119
11.1 DB2 PM Overview	119
11.1.1 Accounting I/O Information	120
11.1.1.1 I/O Operations	120
11.1.1.2 I/O Suspensions	121
11.1.2 Statistics I/O Information	121
11.1.2.1 Data I/O Operations	121
11.1.2.2 Log Activity	122
11.1.3 Performance I/O Information and I/O Activity	123
11.2 RMF Monitoring	124
11.2.1 RMF Report Analysis	125
11.2.1.1 Cache Subsystem Activity Reports	125
11.2.1.2 Direct Access Device Activity Report	128
11.2.2 Using RMF Reports	132
11.2.2.1 Resource Level Analysis	132
11.2.2.2 RMF Reporting at Storage Group Level	133
11.2.2.3 Tools Providing More In-Depth Analysis than RMF	133
11.2.2.4 Spreadsheet Tools for RMF Analyzis	133
11.2.2.5 Global View of a DB2 I/O by DB2 PM and RMF	134
11.3 IXFP Monitoring	135
11.3.1 Device Performance Reports	136
11.3.2 Cache Effectiveness Report	137
11.3.3 Space Utilization Report	138

Chapter 12. Case Study	141
12.1 DB2 Case Study Analysis	141
12.1.1 General Analysis	141
12.1.1.1 Elapsed and CPU Time.	141
12.1.1.2 SQL Statements	141
12.1.1.3 Time Not Accounted	143
12.1.2 Data Access	144
12.1.3 Suspend Times	145
12.1.3.1 Synchronous I/O	145
12.1.3.2 Asynchronous Read I/O	146
12.1.3.3 I/O Rate.	146
12.1.4 Conclusions	147
12.2 Storage Server Analysis	147
12.2.1 RMF Views	147
12.2.1.1 Device Activity Report Analysis.	149
12.2.1.2 I/O Queuing Activity Report Analysis	149
12.2.1.3 Channel Path Activity Report Analysis	150
12.2.1.4 Cache Subsystem Activity Reports Analysis.	151
12.2.2 IXFP View	152
12.2.2.1 Device Performance Overall Summary	152
12.2.2.2 Cache effectiveness Overall Summary	153
12.2.2.3 Space Utilization Summary	153
12.3 Case Study Summary	154

Part 4. Appendixes 159

Appendix A. Test Cases for DB2 Table Space Data Sets	161
A.1 Test Environment	161
A.2 Partitioned Table Space, DB2 Defined, Without SMS	162
A.2.1 Create Eight STOGROUPS.	162
A.2.2 Create the Database	162
A.2.3 Create the Table Space	163
A.2.4 Display a Volume	163
A.3 Partitioned Table Space, User Defined, Without SMS.	164
A.3.1 DEFINE CLUSTER for 16 Partitions	164
A.3.2 CREATE STOGROUP	164
A.3.3 CREATE DATABASE	164
A.3.4 CREATE TABLESPACE	164
A.3.5 Display a Volume	165
A.4 DB2 Table Spaces Using SMS, Existing Names	165
A.4.1 Storage Classes.	166
A.4.2 Management Class	167
A.4.3 Storage Group	169
A.4.4 ISMF Test Cases	171
A.4.5 Updating the Active Configuration	172
A.4.6 DB2 Definitions	172
A.4.7 Data Set Allocation Results	173
A.5 DB2 Table Spaces Using SMS, Coded Names	174
A.5.1 Storage Class	174
A.5.2 Management Class	175
A.5.3 Storage Groups	175
A.5.4 DB2 Definitions	177
A.5.5 Data Set Allocation Results	177

A.6	Partitioned Table Space Using SMS Distribution	178
A.6.1	Define Volumes to SMS Storage Group	179
A.6.2	ACS Routines	179
A.6.3	DB2 Definitions	179
A.6.4	Data Set Allocation Results	180
A.7	Partitioned Table Spaces Using SMS, User Distribution	181
A.7.1	Create Storage Groups	181
A.7.2	ACS Routines	182
A.7.3	DB2 Definitions	183
A.7.4	Data Set Allocation Results	183
Appendix B. Test Cases for DB2 Recovery Data Sets		185
B.1	BSDS and Active Logs	185
B.1.1	SMS Storage Class	185
B.1.2	SMS Management Class	186
B.1.3	Storage Group	187
B.1.4	ISMF Test Cases	188
B.1.5	Data Set Allocation Results	189
B.2	Archive Logs	191
B.2.1	Storage Class	191
B.2.2	Management Class	192
B.2.3	Storage Group	193
B.2.4	Data Set Allocation Results	193
B.3	Image Copies	194
B.3.1	Storage Class	195
B.3.2	Management Class	196
B.3.3	Storage Group	197
B.3.4	Data Set Allocation Results	197
Appendix C. DB2 PM Accounting Trace Report		201
Appendix D. DB2 PM Statistics Report		205
Appendix E. Disk Storage Server Reports		223
Appendix F. Special Notices		239
Appendix G. Related Publications		241
G.1	International Technical Support Organization Publications	241
G.2	Redbooks on CD-ROMs	241
G.3	Other Publications	242
G.4	Web Sites	242
How to Get ITSO Redbooks		243
IBM Redbook Fax Order Form		244
List of Abbreviations		245
Index		247
ITSO Redbook Evaluation		251

Figures

1. Creating a STOGROUP Defined Table Space	14
2. User Defined Table Space: Step 1—Define the Cluster	14
3. User Defined Table Space: Step 2— Define the Table Space	15
4. Installation Panel for Sizing DB2 System Objects	16
5. DB2 Log and Its Data Sets.	19
6. Image Copy SHRLEVEL REFERENCE.	20
7. Image Copy SHRLEVEL CHANGE	21
8. ISMF Primary Option Menu for End Users.	26
9. ISMF Primary Option Menu for Storage Administrators.	27
10. DFSMShsm Components	29
11. Implementing an SMS Configuration	36
12. ACS Routine Execution Process	37
13. SMS Construct Relationship	38
14. Display a Data Class	48
15. Data Class DCDB2	48
16. Display of Storage Group SGDB20	52
17. Volumes in Storage Group SGDB20	53
18. ACS Routine Extract Using Table and Index Name Filter List	55
19. Example VSAM Definition of one BSDS	67
20. Example VSAM Definition of One Active Log	69
21. Archive Log Installation Panel DSNTIPA	70
22. RAMAC3 Drawer Logical Volume Mapping	86
23. LSF Concept 1	87
24. LSF Concept 2	87
25. Snapshot Overview	88
26. Schema of a Backup with Concurrent Copy	94
27. Virtual Concurrent Copy Operation Steps	95
28. Profile of a PPRC Write	96
29. Time Sequenced I/Os	97
30. GDPS Architecture.	98
31. XRC Data Flow	100
32. Storage Hierarchy	104
33. DB2PM Accounting Trace Buffer Pool Report Extract.	109
34. DB2 PM Statistic Report Buffer Pool Reads	110
35. DB2 PM Statistic Report Buffer Pool Writes	110
36. Display Buffer Pool Data Set Statistics	111
37. Log Record Path to Disk	113
38. Two-Phase Commit with Dual Active Logs	113
39. Minimum Active Log Data Set Distribution.	115
40. Installation Panel DSNTIPL	115
41. Log Statistics in a Sample DB2 PM Statistics Report	118
42. Scope of Performance Analysis Tools	119
43. Installation Panel DSNTIPN	120
44. DB2 PM Accounting, Buffer Pool Section	121
45. DB2 PM Statistics, Buffer Pool Read Operations Section	122
46. DB2 PM Statistics, Log Activity Section.	123
47. Buffer Pool Section from I/O Activity Summary Report	124
48. Cache Subsystem Activity Status and Overview Reports	127
49. Cache Subsystem Activity Device Overview Report	127
50. Direct Access Device Activity Report.	129

51. I/O Queuing Activity Report.	131
52. Channel Path Activity Report: LPAR Mode.	132
53. DB2 I/O	135
54. IXFP Device Performance Subsystem Summary Report	137
55. IXFP Cache Effectiveness Subsystem Summary Report	138
56. IXFP Space Utilization Subsystem Report	139
57. DB2 PM Accounting, Class 1 and Class 2 Sections	142
58. DB2 PM Accounting, SQL DML Section	142
59. DB2 PM Accounting, SQL DCL Section	143
60. DB2 PM Accounting, Parallel Query Section	143
61. DB2 PM Statistics, Global DDF Activity Section	144
62. DB2 PM Accounting, BP2 Section	144
63. DB2 PM Accounting, BP4 Section	145
64. DB2 PM Accounting Class 3 Times	145
65. DB2 PM Accounting Highlights	146
66. DB2 PM Accounting Buffer Pool Summary.	147
67. Reducing the RMF Data to Analyze	148
68. Case Study RMF Direct Access Device Activity Report Extract	149
69. Case Study RMF I/O Queuing Activity Extract	150
70. Case Study RMF Channel Path Activity Extract	150
71. Case Study RMF Cache Subsystem Activity Extracts	151
72. Case Study IXFP Device Performance Case Summary Extract.	152
73. Case Study IXFP Cache Effectiveness Overall Extract	153
74. Case Study IXFP Space Utilization Summary Extract	154
75. DB2PM I/O Summary	155
76. Device Activity Summary	155
77. Cache Activity Summary.	156
78. IXFP Summary	156
79. Case Study I/O Flows	157
80. Disk Volume Configuration Used in the Test Environment.	161
81. Test Case 1 - CREATE STOGROUP	162
82. Test Case 1 - CREATE DATABASE.	162
83. Test Case 1 - CREATE TABLESPACE	163
84. Test Case 1 - Display of Volume RV1CU0	163
85. Test Case 2 - DEFINE CLUSTER.	164
86. Test Case 2 - CREATE STOGROUP	164
87. Test Case 2 - CREATE TABLESPACE	165
88. Test Case 2 - Display of Volume RV2CU1	165
89. Test Case 3 - ISMF Storage Class Definition	166
90. Test Case 3 - Storage Class Routine Extract	167
91. Test Case 3 - ISMF Management Class Definition	168
92. Test Case 3 - Management Class Routine Extract.	168
93. Test Case 3 - ISMF Pool Storage Group Definition	169
94. Test Case 3 - Storage Group Routine Extract	169
95. Test Case 3 - ISMF Storage Group Volume Definition	170
96. Test Case 3 - DFSMSdss CONVERTV JCL.	170
97. Test Case 3 - DFSMSdss CONVERTV Output.	171
98. Test Case 3 - ISMF Test against the ACDS	171
99. Test Case 3 - ISMF Test against the Updated SCDS.	172
100. Test Case 3 - CREATE STOGROUP.	172
101. Test Case 3 - CREATE DATABASE Extract	173
102. Test Case 3 - CREATE TABLESPACE Extract	173
103. Test Case 3 - ISPF Data Set List Display.	173

104.Test Case 3 - IDCAMS LISTCAT Display Extract	174
105.Test Case 4 - Storage Class Routine Extract	176
106.Test Case 4 - Management Class Extract.	176
107.Test Case 4 - IDCAMS LISTCAT Extract	178
108.Test Case 5 - ISMF Volume List Display.	179
109.Test Case 5 - CREATE DATABASE	179
110.Test Case 5 - CREATE TABLESPACE Extract.	180
111.Test Case 5 - ISPF Data Set List of Table Space Partitions	180
112.Test Case 5 - ISMF Storage Group Volume Display.	181
113.Test Case 5 - IDCAMS LISTCAT Display Extract	181
114.Test Case 6 - ISMF Storage Group List	182
115.Test Case 6 - Storage Group ACS Routine Extract.	183
116.Test Case 6 - CREATE TABLESPACE Extract.	184
117.Test Case 6 - ISMF Data Set List Extract	184
118.ISMF Storage Class Definition for BSDS and Active Logs	186
119.Storage Class Routine Extract for BSDS and Active Logs	186
120.Management Class Routine Extract for BSDS and Active Logs	187
121.Storage Group Routine Extract for BSDS and Active Logs.	188
122.ISMF Test Result for BSDS (1)	188
123.ISMF Test Result for BSDS (2)	189
124.IDCAMS Definition Extract for BSDS.	189
125.ISPF Data Set List of BSDS's	190
126.SYSPRINT Messages Extract for Active Log IDCAMS Definition	190
127.ISPF Data Set List of Active Logs.	191
128.Storage Class Routine Incorporating Archive Logs	192
129.Management Class Routine Incorporating Archive Logs.	192
130.Storage Group Routine Incorporating Archive Logs.	193
131.SYSLOG Message Ouput Extract for Archive Logs	193
132.ISPF Data Set List of Archive Logs	194
133.IDCAMS LISTCAT of Management Class Comparison for Archive Logs . . .	194
134.Storage Class Routine Extract Incorporating Image Copies	195
135.Management Class Routine Extract Incorporating Image Copies	196
136.Storage Group Routine Extract Incorporating Image Copies	197
137.JCL for Image Copy Allocation	198
138.Image Copy Allocation JES Output Messages	198
139.ISPF Data Set List of Image Copies	199
140.IDCAMS LISTCAT Extract of Image Copy Data Sets	199

Tables

1. Summary of Partition and Partitioned Table Space Sizes	12
2. DB2 Image Copy with and without Concurrent Copy	21
3. Table Space and Index Space Names	23
4. BSDS Names	23
5. Active Log Data Set Names	23
6. Archive Log and BSDS Backup Names	24
7. Sample Image Copy Names	24
8. Data Class Attributes	39
9. Storage Class Attributes	40
10. Management Class Attributes	42
11. Storage Group Attributes	45
12. SMS Storage Classes for DB2 Databases	49
13. SMS Management Classes for DB2 Databases	50
14. Relating SMS Storage and Management Classes to Storage Groups	51
15. SMS Storage Groups for DB2 Databases	51
16. Table Space and Index Space Names with SMS Codes	56
17. Examples of SMS Class Usage for DB2 User Databases	60
18. DB2 System Database Requirements	61
19. SMS Storage Classes for Recovery Data Sets	64
20. Management Classes for Recovery Data Sets	65
21. Relating SMS Storage and Management Classes to Storage Groups	66
22. SMS Storage Groups for DB2 Recovery Data Sets	66
23. Storage Groups for DB2 Recovery Data Sets	73
24. Number of Pages Read Asynchronously in One Prefetch Request	106
25. Maximum Pages in One Write Operation	108
26. Trace Requirement for the I/O Activity Reports	124
27. RVA Space Utilization Comparison	153
28. Test Case 3 - Storage Group Volumes	170
29. Test Case 4 - Storage Group Volumes	175
30. Test Case 5 - Storage Group Volumes	179
31. Test Case 6 - Storage Group Volumes	182
32. BSDS and Active Logs - Storage Group Volumes	187
33. Archive Logs—Storage Group Volumes	193
34. Image Copies—Storage Group Volumes	197

Preface

This redbook will help you tailor and configure DFSMS constructs to be used in a DB2 for OS/390 environment. In addition, this redbook provides a broad understanding of new disk architectures and their impact in DB2 data set management for large installations.

This book addresses both the storage administrator and the DB2 administrator. The DB2 administrator will find information on how to use DFSMS for managing DB2's data sets. The storage administrator will find information on the characteristics of DB2 data sets and how DB2 uses the disks.

After introducing the overall topics of this book, we provide a summary of our conclusions. This will be especially useful for readers responsible for organizing and managing DB2 data in an installation.

The Team That Wrote This Redbook

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

Paolo Bruni is a Data Management Specialist for DB2 for OS/390 at the International Technical Support Organization, San Jose Center, where he conducts projects on all areas of DB2 for OS/390. Paolo has been with IBM for 30 years and has been mostly working with data base management systems.

Hans Duerr is an independent database consultant, specializing in mainframe databases, mainly DB2 for OS/390. He has 17 years of experience with DB2 and has worked for 33 years with IBM in many different countries. He holds a degree in Engineering from the Universidad Tecnica Federico Santa Maria, Valparaiso, Chile. He has been an assignee to the ITSO and has published several red books and conducted workshops in the data management area. Hans is currently based in Madrid, Spain, from where he supports customers all over the world.

Daniel Leplaideur is a technical marketing support specialist at the EMEA ATSC SSD Large System Disks in Mainz. He is based in Paris. Daniel joined IBM in 1967 as a mathematician to develop packages for customers. Since then he worked in the field as Systems Engineer for large accounts. His current areas of expertise are Large System Disks such as RVAs, Disaster/Recovery with XRC, PPRC-GDPS, and DFSMS. He teamworks with EMEA local, ISC and Lab people on ESPs and Disaster/Recovery projects.

Steve Wintle is a systems programmer working for General Electric (Information Delivery Services division), and is based in the United Kingdom. He has 20 years of experience in the MVS field. His areas of expertise include operations support and storage management.

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

Mary Lovelace
Markus Muetschard
Hans-Peter Nagel
Alison Pate
Toru Yamazaki

International Technical Support Organization, San Jose Center

Ted Blank
John Campbell
Paramesh Desai
Ching Lee
Rick Levin
Roger Miller
Akira Shibamiya
Jim Teng
Horacio Terrizzano
Jeff Todd
Steve Turnbaugh
Jay Yothers

IBM Development, Santa Teresa

Bob Kern
Lee La Frese
IBM Development, Tucson

Jeffrey Berger
Bruce Mc Nutt
Paulus Usong
IBM Development, San Jose

Andrea Harris
Nin Lei
S/390 Teraplex Integration Center, Poughkeepsie

Eneo Baborsky
IBM Italy

Philippe Riou
IBM France

Martin Packer
IBM UK

John Burg
Nghì Eakin
IBM Gaithersburg

David Petersen
IBM Washington

Thanks to Elsa Martinez for administration support, Maggie Cutler and Yvonne Lyon for technical editing, and Emma Jacobs for the graphics.

Comments Welcome

Your comments are important to us!

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

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 251 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users <http://www.redbooks.ibm.com/>

For IBM Intranet users <http://w3.itso.ibm.com/>

- Send us a note at the following address:

redbook@us.ibm.com

Part 1. Introduction and Summary

Chapter 1. Introduction

Auxiliary storage management in the DB2 environment for the MVS platform has, so far, been mainly the responsibility of the database administrators.

In the first few years of its usage, DB2's implicit definition of page sets through its Storage Groups (STOGROUP) often replaced the more traditional method of explicitly allocating VSAM data sets because of DB2's simplicity and ease of use.

Database administrators worried about separation of critical data sets, like data from indexes, data from log, copies of log and BSDS, spreading workfiles, through the usage of multiple Storage Groups and the careful association of volumes to Storage Groups.

Until only few years ago, operators, storage managers, system programmers and performance analysts had to interact frequently with the database administrators in order to resolve issues related to DB2 data set management. Furthermore, database administrators did not look favorably at SMS space management because they felt that it interfered with the hand-placement of critical DB2 data sets; SMS usage was limited to some hierarchical management of backup data sets (image copies and archived logs).

Today, on one side we have a growing number of data warehousing types of applications which require very large table spaces and query parallelism, causing an explosion of the number of DB2 objects; on the other side we have more flexible functions in SMS related products and innovative changes in the disk architecture that can provide very useful functions for space and back-up management. Most medium to large DB2 installations have to devote quite a considerable amount of resources to the management of several thousand DB2 objects.

Furthermore, as processors and disk control units provide more capacity and more memory, DB2 exploits its larger buffer pools as a second level of cache for I/O execution, reducing the I/O frequency and making it mostly asynchronous. This implies that the criticality of data set placement is greatly reduced.

In this redbook, as a level set, first we examine DB2 data set and I/O characteristics, then we look at the main concepts and functions of SMS, and then at the recent evolution of storage servers (disks).

We then provide a mapping of the possible applicability of SMS for all but the most critical applications. This allows the database administrators to concentrate on DB2 data sets relative to the applications with the highest service level requirements, while the storage administrators can use SMS to simplify disk use and control.

We finally look at the impact that large cache and the virtual architecture of the current disk technology have on dealing with DB2 data.

Because of the necessity to monitor performance to avoid surprises, we also show how to look at DB2 and I/O performance tools output from the overall storage management perspective. Several examples are reported in the appendixes.

Chapter 2. Summary of Considerations

This book describes the exploitation of storage by DB2 for OS/390 (DB2). Two major areas are analyzed:

1. DB2 and storage management
2. DB2 and storage servers

This chapter summarizes the major conclusions of this project.

2.1 DB2 and Storage Management

A detailed analysis of the different types of DB2 data sets shows that DFSMS can automatically manage all of the data sets DB2 uses and requires. However, there are considerations and choices that need to be made to tailor DFSMS to suit the individual customer's systems environment and organization.

In general, a large percentage of your data sets can be managed with DFSMS storage pools, thus reducing the workload and the interaction of your DB2 database administrators (DBAs) and storage administrators. Only the most critical data, as defined with service level agreements or as revealed by monitoring, may require special attention.

2.1.1 Benefits of DFSMS

Using DFSMS, the DB2 administrator gains the following benefits:

- Simplified data allocation
- Improved allocation control
- Improved performance management
- Automated disk space management
- Improved data availability management
- Simplified data movement

See 4.4, "Benefits" on page 32 for more details.

Another very important benefit is that, with DFSMS, the DB2 environment is positioned to take immediate advantage of available and future enhancements. For example, the following enhancements are available today to DB2 with the appropriate level of DFSMS:

- DFSMS 1.4
 - Space allocation failures are reduced with the support of a maximum number of 255 extents per component of VSAM data set for multivolume data sets (the limit is 123 data sets for a single volume allocation).
 - Image copy with concurrent copy support for RAMAC Virtual Array SnapShot.
- DFSMS 1.5
 - Support for 254 table space or index space partitions or pieces up to 64 GB with the use of VSAM Extended Addressability for Linear Data Sets; also 4,000 TB support for LOBs.

- DB2 data sharing performance improvement for open/close of data sets (especially beneficial during DB2 start-up) with Enhanced Catalog Sharing (ECS); ECS reduces the path length and supports the ICF shared catalog on the coupling facility.

You can check the Appendix section G.4, “Web Sites” on page 242 for sites on DB2 and DFSMS reporting the most current information on the supported functions.

2.1.2 Managing DB2 Data Sets with DFSMS

The DB2 administrator can use DFSMS to achieve all the objectives for data set placement and design. DFSMS has the necessary flexibility to support everything the DB2 administrator may want. There is no reason whatsoever for not taking advantage of DFSMS for DB2 data sets.

To achieve a successful implementation, an agreement between the storage administrator and the DB2 administrator is required so that they can together establish an environment that satisfies both their objectives.

2.1.3 Examples for Managing DB2 Data Sets with DFSMS

Examples are shown to describe one possible way to manage DB2 data sets with DFSMS. These examples are not supposed to be a recommendation. The examples are shown to give an idea on the possibilities that DFSMS offers for DB2. Each example is just one out of many choices of how a medium to complex installation may approach the implementation of DB2 data sets with DFSMS.

Many installations may find a simpler implementation more adequate, while others may want to have a more specific management than the one shown.

2.2 DB2 and Storage Servers

DB2 has some special requirements in the way its storage objects are defined and utilized. Disk technology has evolved introducing RAID architecture, large cache, virtual architecture. DBAs and storage administrators need to agree on common actions in order to take advantage of the available enhancements.

2.2.1 Data Placement

With smaller disk devices, without cache, data locality was important for performance, to reduce seek and rotation times. The new disk architectures, with concepts like log structured files and with cache in the gigabyte sizes, have a noticeable impact on database physical design considerations. Conventional database design rules based on data set placement are becoming less important and can be ignored in most cases.

2.2.2 Large Cache

Most storage servers with large cache (greater than 1 GB) ignore the *bypass cache* or *inhibit cache load* requests from the application. They always use the cache; however, they continue to take into account the specifications of usage from the applications by just scaling down or up the track retention into the cache for reuse.

Installations having these devices could use *sequential caching* as an installation option. Installations with a mixture of devices with large, small, or no cache can benefit from the *bypass cache* option.

2.2.3 Log Structured File

Devices using the log structured file technique (like the RVA) do not maintain data location during data updates. For these devices there exists a concept of logical location of data, independent from the the physical location. The logical location is used by the device to present the data to the application: the user sees a contiguous extent on a 3390 volume, while the data is in reality scattered across the LSF.

A REORG of a DB2 table space provides a logical sequence of records which could not be corresponding to a physical sequence. This is a function of the space management of the storage server.

Worrying about reorganizing data to reclaim space extents is now much less critical with the new disk architecture. REORG does not need to be run in order to reclaim fragmented space in this case, only to reestablish the clustering (logical) sequence and the DB2 internal free space. When the DB2 optimizer chooses sequential prefetch as a valid access path, the storage server detects the logical sequential access and initiates pre-staging of the logically sequenced tracks into cache, providing improvement to the I/O response time for the subsequent prefetch accesses.

2.2.4 RAMAC Architecture

Disk architecture defines each volume in a logical way through tables. These tables do an effective mapping between the logical view of the volume onto the disk array with data and rotating parity physical disks. This means that each I/O operation takes place to or from several physical disks. However, the host still views only the logical volume topology, and it bases its optimizing and scheduling strategies on this view, as it used to do with native 3380 and 3390.

2.2.5 SMS Storage Groups

Volume separation is easy when you have hundreds of volumes available. But this separation is good only if your volumes have separate access paths. Path separation is important to achieve high parallel data transfer rates.

Without DFSMS, the user is responsible for distributing DB2 data sets among disks. This process needs to be reviewed periodically, either when the workload changes, or when the storage server configuration changes.

With DFSMS, the user can distribute the DFSMS Storage Groups among storage servers with the purpose of optimizing access parallelism. Another purpose could be managing availability for disaster recovery planning. This can be combined with the previous purpose by letting DFSMS automatically fill in these Storage Groups with data sets, by applying policies defined in the automatic class selection routines.

Changes to the topology of the Storage Group can be managed to minimize the application outages. This can be done simply by adding new volumes to the Storage Group, then managing the allocation enablement (opening it on new

volumes, closing it on volumes to be removed), and finally removing the volumes you want to exclude from the Storage Group. All those functions can be accomplished while the data is on line. Data sets that were unmovable, never-closed, or never reallocated could be moved using remote copy techniques, then, after a short outage, the critical application can be switched onto the new volumes.

2.2.6 Performance Management

Monitoring I/O performance of DB2 requires also teamwork between DB2 and storage administrators to adopt a common approach with tools of both disciplines in analyzing performance situations. Performance monitoring should be done at the Storage Group level to have a consistent action.

Part 2. DB2 and System Managed Storage

Chapter 3. DB2 Storage Objects

This chapter represents an introduction to DB2 for OS/390 (DB2 throughout this redbook) for the storage administrators interested in understanding the different types of data related objects used in a DB2 environment. Special emphasis is placed on the data sets managed directly by DB2.

3.1 DB2 Overview

DB2 is a database management system based on the relational data model. Many customers use DB2 for applications which require good performance and/or high availability for large amounts of data. This data is stored in data sets directly associated to DB2 table spaces and distributed across DB2 databases. Data in table spaces is often accessed through indexes; indexes are stored in index spaces.

Data table spaces can be divided into two groups: system table spaces and user table spaces. Both of these have identical data attributes. The difference is that system table spaces are required to control and manage the DB2 subsystem and the user data. The consequence of this is that system table spaces require the highest availability and some special consideration. User data cannot be accessed without system data or with obsolete system data.

In addition to the data table spaces, DB2 requires a group of traditional data sets, not associated to table spaces, that are used by DB2 in order to provide the appropriate high level of data availability, the back-up and recovery data sets. Proper management of these data sets is required to achieve this objective.

In summary, the three main data set types in a DB2 subsystem are:

1. DB2 back-up and recovery data sets
2. DB2 system table spaces
3. DB2 user table spaces

3.2 DB2 Data Objects

DB2 manages data by associating it to a set of DB2 objects. These objects are logical objects. Some of these objects have a physical representation on storage devices. The DB2 data objects are:

- TABLE
- TABLESPACE
- INDEX
- INDEXSPACE
- DATABASE
- STOGROUP

A complete description of all DB2 objects and their implementation can be found in the *DB2 for OS/390 Administration Guide*, SC26-8957, in Section 2. Designing a database.

3.2.1 TABLE

All data managed by DB2 is associated to a table. The data within the table is organized in columns and rows, and this represents the minimum unit of data that can be identified by the user.

The table is the main object used by DB2 applications. The SQL DML used by application programs and end users directly references data in tables.

3.2.2 TABLESPACE

A table space is used to store one or more tables. A table space is physically implemented with one or more data sets. Table spaces are VSAM linear data sets (LDS). Because table spaces can be larger than the largest possible VSAM data set, a DB2 table space may require more than one VSAM data set.

One of three different types of table spaces may be chosen for a specific table:

1. Simple (normally containing one table)
2. Segmented (containing one or more tables)
3. Partitioned (containing one, often large, table)
4. LOB (large object, new type of table space introduced with DB2 V6).

The maximum size of simple and segmented table spaces is 64 GB, corresponding to the concatenation of up to 32 data sets, each of up to 2 GB in size.

A single LOB column can be 2 GB, and the collection of all LOB values for a given LOB column can be up to 4000 TB.

Table 1 on page 12 summarizes the partition and partitioned table space sizes for different versions of DB2.

Table 1. Summary of Partition and Partitioned Table Space Sizes

DB2 Version	Number of Partitions	Maximum Size	Total Maximum Size
V4*	1 to 16	4 GB	64 GB
	17 to 32	2 GB	64 GB
	33 to 64	1 GB	64 GB
V5**	254	4 GB	1,016 GB
V6***	254	64 GB	16,256 GB
Note: * For a maximum total of 64 GB			
Note: ** Requires LARGE parameter in CREATE TABLESPACE			
Note: *** Requires DFSMS/MVS 1.5 and SMS-managed table space			

Up to DB2 V4 the total maximum size of a partitioned table space is 64 GB. Starting with DB2 V5, with the introduction of the LARGE parameter at creation time, partitioned table spaces may have a total size of 1,016 GB, corresponding to up to 254 partitions each with a data set size of 4 GB.

DB2 V6 has increased the maximum size of a partitioned table space to almost 16 TB, increasing the maximum data set size to 64 GB. This is supported only if they are defined and managed with DFSMS 1.5.

3.2.3 INDEX

A table can have zero or more indexes. An index contains keys. Each key may point to one or more data rows. The purpose of indexes is to establish a way to get a direct and faster access to the data in a table. An index with the UNIQUE attribute enforces distinct keys and uniqueness of all rows in the referenced table. An index with the CLUSTER attribute can be used to establish and maintain a physical sequence in the data.

3.2.4 INDEXSPACE

An index space is used to store an index. An index space is physically represented by one or more VSAM LDS data sets.

When a non-partitioning index needs to be split across multiple data sets in order to improve I/O performance, these particular type of data sets are called PIECES. DB2 V5 has introduced the capability of defining up to 128 pieces with a maximum size of 2 GB. DB2 V6 and DFSMS 1.5 increase the limit to 254 pieces of up to 64 GB.

3.2.5 DATABASE

A database is a DB2 representation of a group of related objects. Each of the previously named objects has to belong to a database. DB2 databases are used to organize and manage these objects. Normally a database is the association of table spaces and index spaces used by an application or a coherent part of an application.

3.2.6 STOGROUP

A DB2 Storage Group (STOGROUP) is a list of storage volumes. STOGROUPs are assigned to databases, table spaces or index spaces when using DB2 managed objects. DB2 uses STOGROUPs for disk allocation of the table and index spaces.

Installations that are SMS managed can define STOGROUP with VOLUMES(*). This specification implies that SMS assigns a volume to the table and index spaces in that STOGROUP. In order to do this, SMS uses ACS routines to assign a Storage Class, a Management Class and a Storage Group to the table or index space.

3.3 Creating Table Spaces and Index Spaces

Table and index spaces can be created in one of two ways:

- DB2 defined and managed
- User defined and managed

See the *DB2 for OS/390 Administration Guide*, SC26-8957, Chapter 2, Section 2-7, "Designing Storage Groups and Managing DB2 Data Sets", for a detailed explanation on this subject.

DB2 defined and managed spaces should be the choice by default. It is the easier of these solutions and is adequate for most table and index spaces in the majority of situations.

User defined table spaces provide more control of the data set placement and all VSAM definition options are available. Examples of where user defined table spaces may be required are:

- Table spaces that require specific DFSMS classes
- Table spaces with critical performance and availability requirements
- (In general, table spaces with special requirements)

3.3.1 DB2 Defined and Managed

Figure 1 on page 14 shows an example of a DB2 defined table space. The CREATE TABLESPACE resolves the physical allocation and defines this table space to DB2. The STOGROUP DSN8G610 defines a set of volumes for the data set and PRIQTY specifies the size in KB. Indexes are created with a CREATE INDEX statement. The CREATE INDEX statement defines both the index and the associated index space. The CREATE INDEX also loads the index with entries that point to the data (index rows) unless the DEFER YES parameter is specified.

```
CREATE TABLESPACE PAOLOR11
  IN DSN8D61A
  USING STOGROUP DSN8G610
      PRIQTY 20
      SECQTY 20
      ERASE NO
  LOCKSIZE ANY LOCKMAX SYSTEM
  BUFFERPOOL BP0
  CLOSE NO
  CCSID EBCDIC;
```

Figure 1. Creating a STOGROUP Defined Table Space

3.3.2 User Defined and Managed

Two steps are required to create user defined table spaces or index spaces.

1. The physical allocation is done with an IDCAMS DEFINE CLUSTER; this is shown in Figure 2 on page 14.

```
DEFINE CLUSTER -
  ( NAME (DB2V610Z.DSNDBC.DSN8D61A.PAOLOR1.I0001.A001) -
    LINEAR -
    REUSE -
    VOLUMES (SBOX10) -
    RECORDS (4096 50) -
    SHAREOPTIONS (3 3) ) -
  DATA -
  ( NAME (DB2V610Z.DSNDBD.DSN8D61A.PAOLOR1.I0001.A001) ) -
  CATALOG (DS2V6)
```

Figure 2. User Defined Table Space: Step 1—Define the Cluster

2. The table space or index space must be defined to DB2. An example is shown in Figure 3 on page 15. It must be noted that the high level qualifier, the

database name, and the table space name in Figure 2 on page 14 must match the definitions on Figure 3 on page 15.

```
CREATE TABLESPACE PAOLOR1 IN DSN8D61A
      BUFFERPOOL BP0
      CLOSE NO
      USING VCAT DB2V610Z;
```

Figure 3. User Defined Table Space: Step 2— Define the Table Space

3.4 DB2 System Table Spaces

DB2 uses four internally defined databases to control and manage itself and the application data. The databases are:

- The Catalog database
- The Directory database
- The Work database
- The Default database

This section provides a general description of these databases.

Two tables, SYSIBM.SYSCOPY, belonging to the DB2 catalog, and SYSIBM.SYSLGRNX, belonging to the DB2 directory, are directly used by DB2 to manage backup and recovery; they are also described in this section.

3.4.1 The DB2 Catalog and Directory

The Catalog database is named DSNDB06. The Directory database is named DSNDB01. Both databases contain DB2 system tables. DB2 system tables store data definitions, security information, data statistics and recovery information for the DB2 system. The DB2 system tables reside in DB2 system table spaces.

The DB2 system table spaces are allocated when a DB2 system is first created, that is, during the installation process. DB2 provides the IDCAMS statements required to allocate these data sets as VSAM LDSs. The size of these LDSs is calculated from user parameters specified on DB2 installation panels. Figure 4 on page 16 shows panel DSNTIPD with default values for sizing DB2 system table spaces. In this figure, parameters numbered 1 to 12 are used to size DB2 catalog and directory table spaces.

3.4.2 The Work Database

In a non-data sharing environment, the Work database is called DSNDB07. In a data sharing environment, the name is chosen by the user. The Work database is used by DB2 to resolve SQL queries that require temporary work space. For example, SQL statements containing JOIN, ORDER BY, GROUP BY, may require space in the Work database, but this depends on available storage for the DB2 internal sort and the access path chosen by the DB2 optimizer.

Multiple table spaces can be created for the Work database. These table spaces follow the normal rules for creating a table space. At least two table spaces should be created, one with a 4K page size, and the other one with a 32K page size. DB2 V6 supports page sizes of 8K and 16K for table spaces, but not for the

Work database table spaces. When required, 8K and 16K pages are placed in a 32K table space of the Work database.

Figure 4 on page 16 shows how the size and number of table spaces in the Work database is defined. Parameters 13 through 16 on this figure show the default number of table spaces (one for each page size), and default sizes for the 4K table space (16 MB) and the 32K table space (4 MB).

```

DSNTIPD          INSTALL DB2 - SIZES
====>

Check numbers and reenter to change:

 1 DATABASES          ===> 200      In this subsystem
 2 TABLES           ===> 10       Per database (average)
 3 COLUMNS           ===> 10       Per table (average)
 4 VIEWS              ===> 3        Per table (average)
 5 TABLE SPACES     ===> 10       Per database (average)
 6 PLANS              ===> 200      In this subsystem
 7 PLAN STATEMENTS   ===> 30       SQL statements per plan (average)
 8 PACKAGES           ===> 300      In this subsystem
 9 PACKAGE STATEMENTS ===> 10      SQL statements per package (average)
10 PACKAGE LISTS     ===> 2        Package lists per plan (average)
11 EXECUTED SIMTS    ===> 15       SQL statements executed (average)
12 TABLES IN SIMT   ===> 2        Tables per SQL statement (average)
13 TEMP 4K SPACE     ===> 16       Amount of 4K-page work space (megabytes)
14 TEMP 4K DATA SETS ===> 1      Number of data sets for 4K data
15 TEMP 32K SPACE    ===> 4        Amount of 32K-page work space (megabytes)
16 TEMP 32K DATA SETS ===> 1     Number of data sets for 32K data

F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

Figure 4. Installation Panel for Sizing DB2 System Objects

3.4.3 SYSIBM.SYSCOPY

This table is a DB2 Catalog table. It is referred to by its short name: SYSCOPY. The table space in which SYSCOPY is stored is called: DSNDB06.SYSCOPY.

SYSCOPY contains recovery related information for each table space. Its main purpose is to manage image copies, but other related recovery information is also recorded here. For example, SYSCOPY contains information of:

- Image copies
- Quiesce points
- LOAD executions
- REORG executions
- RECOVER executions

3.4.4 SYSIBM.SYSLGRNX

This table is a DB2 Directory table. It is referred to by its short name: SYSLGRNX. The table space in which SYSLGRNX is stored is called: DSNDB01.SYSLGRNX.

SYSLGRNX stores records that serve to improve recovery performance by limiting the scan of the log to changes that must be applied. The SYSLGRNX records contain the first log RBA (or LRSN in a data sharing group) and the last

log RBA (or LRSN) of updates to a table space. The record is opened when a first update is detected, and closed after an interval of read only activity. The interval is defined with two read-only switch parameters on the DB2 installation panel DSNTIPN.

3.5 DB2 Application Table Spaces

All application data in DB2 is organized in the objects described in 3.2, "DB2 Data Objects" on page 11. Application table spaces and index spaces are created as shown in 3.3, "Creating Table Spaces and Index Spaces" on page 13.

Application table spaces and index spaces are VSAM LDS data sets, with exactly the same data attributes as DB2 system table spaces and index spaces. The distinction between system and application data is made only because they have different performance and availability requirements.

3.6 DB2 Recovery Data Sets

In order to ensure data integrity, DB2 uses several traditional data sets for recovery purposes. Not all of these are always needed by DB2, but all of them are required for contingency reasons. DB2 supports two or more copies of these data sets to ensure a high level of data integrity.

A short description of DB2 recovery data sets is provided here. A good description is available in "Managing the Log and the Bootstrap Data Set" in the *Administration Guide of DB2 for OS/390*, SC26-8957. An attempt has been made to avoid redundancy in these descriptions.

3.6.1 Bootstrap Data Sets

DB2 uses the bootstrap data set (BSDS) to manage recovery and other DB2 subsystemwide information. The BSDS contains information needed to restart and to recover DB2 from any abnormal circumstance. For example, all log data sets (active and archive) are automatically recorded within the BSDS. While DB2 is active, the BSDS is open and is updated.

Because the BSDS is critical for DB2 data integrity, DB2 always requires the presence of two copies of the BSDS at start up time. If a copy fails while DB2 is running, DB2 sends a warning message and continues operating with a single BSDS. It is the responsibility of operations to monitor this circumstance and restore the BSDS duality as soon as possible.

To recover a lost BSDS, when DB2 is executing:

1. The failing BSDS must be deleted.
2. The failing BSDS must be redefined, or alternatively, an existing spare BSDS copy must be renamed.
3. The BSDS is rebuilt with a `-RECOVER BSDS` command.

If a BSDS copy fails while DB2 is starting, the startup does not complete.

To recover a lost BSDS, when DB2 is stopped:

1. The failing BSDS must be deleted.

2. The failing BSDS must be redefined, or alternatively, an existing spare BSDS copy must be renamed.
3. The BSDS is rebuilt from the good copy with an IDCAMS REPRO.

3.6.2 Active Logs

The active log data sets are used for data recovery and to ensure data integrity in case of software or hardware errors. DB2 uses active log data sets to record all updates to user and system data.

The active log data sets are open as long as DB2 is active. Active log data sets are reused when the total active log space is used up, but only after the active log (to be overlaid) has been copied to an archive log.

DB2 supports dual active logs. It is strongly recommended to make use of dual active logs for all DB2 production environments.

Sizing Active Logs

The amount of space dedicated to each individual active log data set is not critical for the DB2 administrator. Traditionally, the active logs have been sized for practical reasons, for example, to make best use of the archive log device (tape cartridge or disk volume).

It is the overall size of all active log data sets that is important for the DB2 DBA: this size plays a critical role in the backup and recovery strategy.

The number of active log data sets, multiplied by the space of each active log, defines an amount of log information most readily available: the capacity of the active log. This capacity defines the time period that has the best recovery performance and the highest data availability service level. The reason is that the DB2 RECOVER utility generally performs better with an active log than with an archive log. See 10.5.2, "Active Log Size" on page 117 for more details.

Impact of Log Size on Backup and Recovery Strategy

The relationship between the different types of log data sets is shown in Figure 5 on page 19. This figure shows a timeline that begins when a DB2 subsystem is first started (Start Time) and proceeds until the current time (Current Time). During this whole time, log data has been generated; this is shown by the DB2 LOG bar.

The log data sets have limited capacity and cannot cover the total time period.

The amount of DB2 log in the active log data sets (the active log capacity) is shown as the time period from Time 2 to the Current Time. The oldest still available archive log corresponds to Time 1. Because the whole log is not available, recoveries are only possible throughout the period from Time 1 to Current Time. The time period from Time 2 to Current Time corresponds to the period with most efficient recoveries because, generally, the active log is allocated on faster devices. The archive log usually overlaps with the active log for a minimum of the last pair of active log data sets not yet archived up to some time after Time 2 and before Current Time. If the data needed for RECOVER or RESTART has been archived, but is still available on an active log data set not yet reused, DB2 accesses the active log.

A good backup and recovery strategy considers:

- The amount of time to cover with all logs (Time 1 up to Current Time)
- The amount of time to cover with active logs (Time 2 up to Current Time)

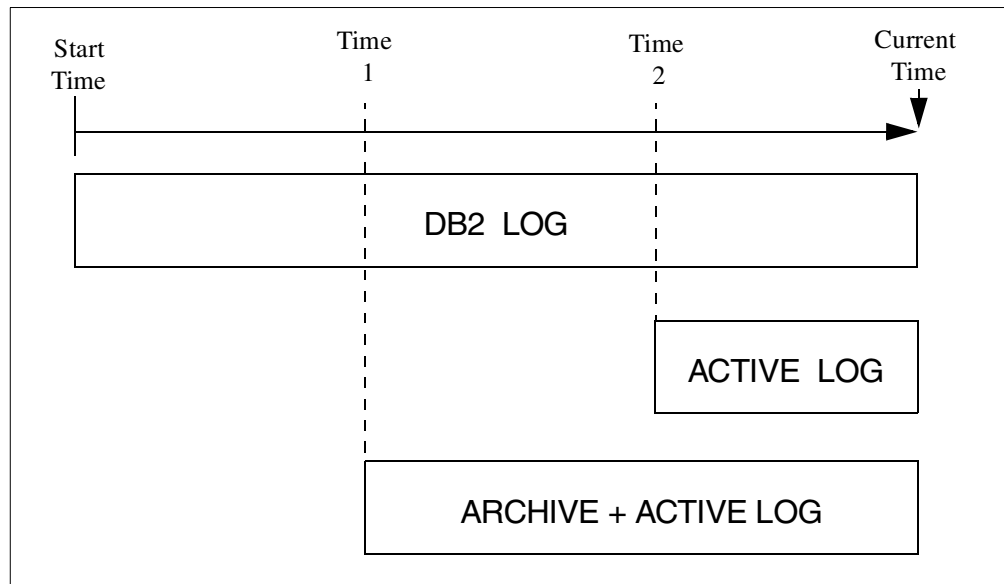


Figure 5. DB2 Log and Its Data Sets

3.6.3 Archive Logs

Archive log data sets are DB2 managed backups of the active log data sets. Archive log data sets are created automatically by DB2 whenever an active log is filled. They may also be created with the `-ARCHIVE` command for operational requirements. Additional circumstances may trigger the archiving process. The *DB2 for OS/390 Administration Guide, SC26-8957*, describes these in detail.

DB2 supports dual archive logs and it is recommended to use dual archive log data sets for all production environments. When dual archive log is specified, during the archiving, the primary active log is read and two archive log data sets are written in parallel. For better archive log availability, customers should define both copies on different devices (or SMS classes) to physically separate the dual data sets.

Archive log data sets are required for any recovery that spans a period of time in excess of the time covered by the active logs. Archive log data sets are sequential data sets that can be:

- Defined on disk or on tape
- Migrated
- Deleted with standard procedures

Archive log data sets are required for data integrity. Procedures are required to ensure that archive log data sets are only deleted when they are not going to be required anymore.

This book contains references related to archive log deletes, in the following sections:

- “Deleting Image Copies and Archive Logs” on page 21.
- “Impact of Log Size on Backup and Recovery Strategy” on page 18.

3.6.4 Image Copies

Image copies are the backup of user and system data. DB2 V6 introduces the possibility of taking image copy for indexes. For a well-managed backup and recovery policy, it is recommended that the amount of data in image copy data sets should cover at least three generations of image copies in order to guarantee recoverability. This means that a large number of image copy data sets is required and needs to be managed in DB2 installations.

Image Copy Availability

Image copies ensure user and system data integrity. Their availability is critical for DB2 system and application availability. DB2 can optionally generate up to four image copies of a table space, index space, or data set (for a multiple data set table space or index space). Two of these copies are intended for a disaster recovery at a remote site. For better image copy availability, customers should define the copies on different devices (or SMS classes) to physically separate the data sets.

Image Copy Options

Image copies can be run in two important varieties, either FULL or INCREMENTAL. Full image copies are complete backups of a table space or data set. Incremental copies only contain the changes since the last full image copy. Incremental and image copies can be combined (merged) to create other incremental or full image copies.

The SHRLEVEL option is used to specify application access during the copy. SHRLEVEL REFERENCE creates a consistent copy. During the SHRLEVEL REFERENCE copy, only read access is allowed. SHRLEVEL CHANGE creates a copy while the data is updated. Figure 6 on page 20 and Figure 7 on page 21 illustrate the impact these copies have on application read and write processing.

The DB2 RECOVER utility can handle the updates not reflected in a SHRLEVEL CHANGE copy by applying the log records corresponding to those updates.

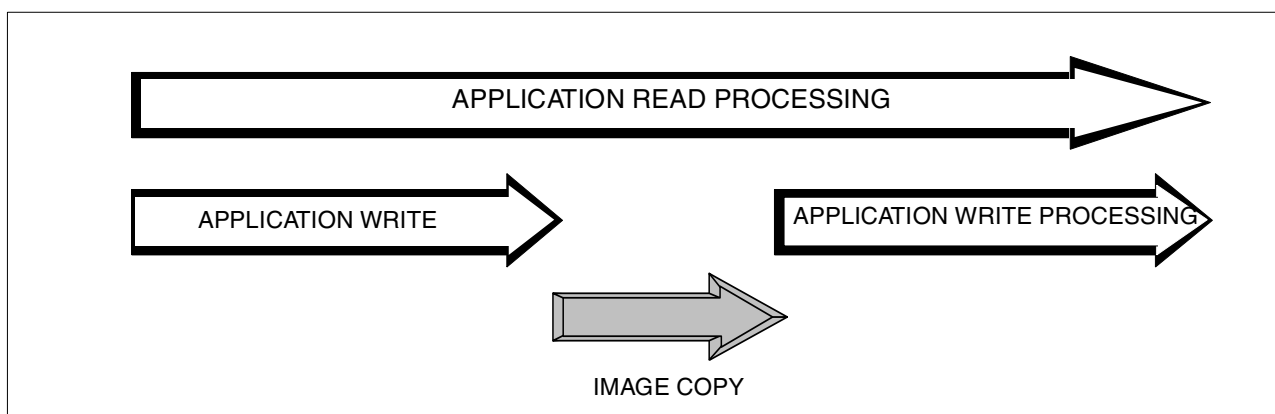


Figure 6. Image Copy SHRLEVEL REFERENCE

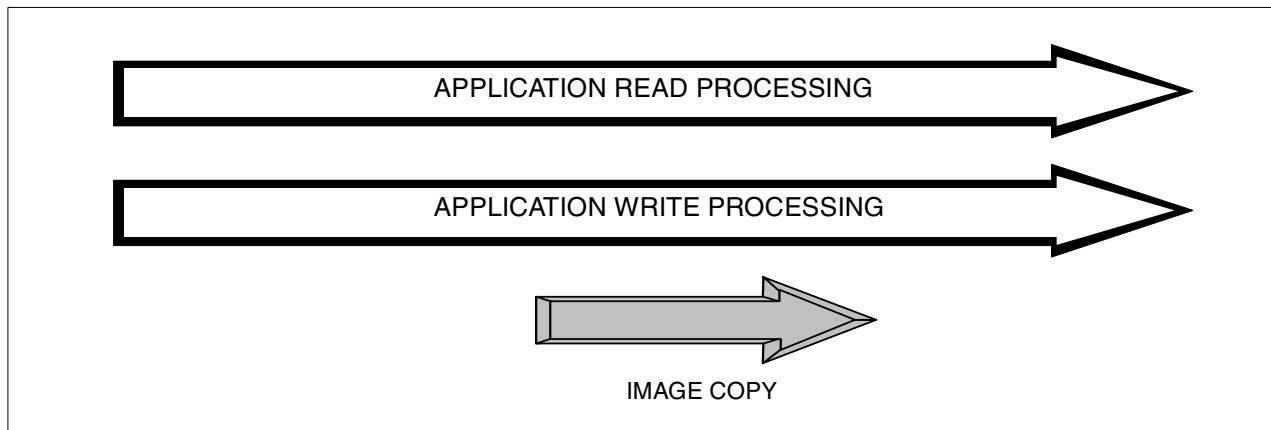


Figure 7. Image Copy SHRLEVEL CHANGE

Another option for image copies is the use of the concurrent copy feature, with or without SnapShot. Concurrent copy and SnapShot are described in 9.5.2, “Concurrent Copy” on page 94. These features allow DB2 to create full image copies with only a short time interval of data unavailability. This is illustrated in Figure 26 on page 94. The DB2 RECOVER utility is able to handle these copies. Table 2 on page 21 shows the different options available for an image copy with and without the concurrent option.

Table 2. DB2 Image Copy with and without Concurrent Copy

CONCURRENT	TYPE		SHRLEVEL	
	FULL	INCR	REFERENCE	CHANGE
YES	YES	NO	YES	YES ^{a,b}
NO	YES	YES	YES	YES

a. Short unavailability at data set level

b. Not valid for page size larger than 4K

Image Copy Failures During Recovery

During a recovery, an image copy may fail (for example, due to an I/O error). In this case, RECOVER attempts to use the dual image copy, assuming that such a copy exists. If the copy does not exist or also fails, RECOVER ignores the copy if it is an incremental image copy, and uses the log for recovery. If the failing image copy is a full copy, RECOVER falls back to an earlier full image copy to complete the recovery. The fallback has a performance penalty, but it helps to insure availability.

Because the fallback insures recoverability, some installations do not generate dual image copies. These installations prefer to run frequent incremental image copies instead.

Deleting Image Copies and Archive Logs

Image copies are required for data integrity. The customer must have procedures to ensure that image copies are deleted only when they are not required anymore. Moreover, because image copies and archive logs are used together, the deletion of these data sets has to be synchronized. For example, there no use

for an archive log that is older than the oldest image copy unless other types of backups, not just image copies, are also used for recovery.

Image copies and archive logs are recorded in DB2 and optionally cataloged in an ICF Catalog. Physical deletion of the data sets removes them from the ICF catalog. This physical deletion must be coordinated with a DB2 cleanup procedure to remove obsolete information in SYSIBM.SYSCOPY. This cleanup is performed with the MODIFY utility.

The deletion from the MVS catalog and the DB2 catalog of image copy data sets must also be synchronized with the deletion of the log data sets from the MVS catalog and from the BSDS.

3.6.5 Other Copies

DB2 table and index spaces can be copied by other utilities, not under DB2 control. This can include both IBM (DFSMSdfp, DSN1COPY) and non-IBM products. DB2 has a limited support for these copies. The copies must be restored outside of DB2, and the user must execute a RECOVER with option LOGONLY to apply the changes not reflected in the external copy in order to maintain data integrity and consistency.

3.7 Other DB2 Data Sets

Apart from table spaces and recovery data sets, DB2 requires also data sets to store the product (libraries), to manage its execution (Clists, JCL procedures, and work data sets). These data sets are standard MVS data sets, either partitioned or sequential.

3.7.1 DB2 Library Data Sets

DB2 uses a set of library data sets to store distribution code, executable code, ISPF data sets, TSO data sets, SMP/E data sets, and so on. These library data sets are all MVS partitioned data sets (PDS). These data sets are defined during the SMP/E installation of the DB2 product.

3.7.2 DB2 Temporary Data Sets

DB2 also uses temporary data sets. Examples are utility work and external sort data sets. Most temporary data sets are standard sequential files. These data sets are defined explicitly in utility JCL or are created dynamically at utility execution time.

To allocate these data sets, DB2 has internal default attributes that can be overridden by the user in the JCL stream.

3.8 DB2 Data Sets Naming Conventions

This section describes the naming standards used by DB2 for its data sets.

3.8.1 Table Space and Index Space Names

The names for DB2 table spaces and index spaces have the following structure:

Table 3. Table Space and Index Space Names

hlq.DSNDBx.dbname.spname.ynnnn.Ammm

The elements of this name are:

<i>hlq</i>	VSAM catalog high level qualifier
<i>DSNDB</i>	Standard part of the name
<i>x</i>	Identifies a VSAM cluster or data component <i>C</i> Cluster <i>D</i> Data
<i>dbname</i>	Database name
<i>spname</i>	Space name. Either a table space name or an index name. Because index names can be more than 8 characters long, DB2 sometimes needs to generate an 8 character name. To avoid randomly generated names, and to be able to correlate the index name to the index space, it is recommended to limit index names to 8 characters. This is also true for table names for implicitly defined table spaces (that is, the creation of the table is done without having created the table space), since DB2 will assign a unique table space name.
<i>y</i>	Data set type: <i>I</i> Standard data set <i>S</i> Shadow data set <i>T</i> Temporary data set
<i>nnnn</i>	number = 0001
<i>A</i>	Standard character, A
<i>mmm</i>	Used for table spaces or index spaces with multiple data sets; <i>mmm</i> is either <i>001</i> , the data set number, or the partition number.

3.8.2 BSDS Names

The default names for BSDSs have the following structure:

Table 4. BSDS Names

hlq.BSDS0n

<i>hlq</i>	VSAM catalog high level qualifier
<i>BSDS0</i>	Standard part of the name
<i>n</i>	BSDS copy, 1 or 2

3.8.3 Active Log Names

The default names for active log data sets have the following structure:

Table 5. Active Log Data Set Names

hlq.LOGCOPYn.DSmm

<i>hlq</i>	VSAM catalog high level qualifier
------------	-----------------------------------

LOGCOPY Standard part of the name
n Active log copy, 1 or 2
mm Active log number, 01 to 31

3.8.4 Archive Log and BSDS Backup Names

The default names for archive log and BSDS backup data sets have the following optional structure:

Table 6. Archive Log and BSDS Backup Names

hlq.ARCHLOGn.Dyyddd.Thhmsst.axxxxxx

hlq VSAM catalog high level qualifier
ARCHLOG Standard part of the name
n Archive log copy, 1 or 2
Dyyddd Date, yy=year (2 or 4 digits), ddd=day of year
Thhmsst Time, hh=hour, mm=minute, ss=seconds, t=tenths
a A=Archive log, B=BSDS backup
xxxxxx File sequence

Dyyddd and Thhmsst are optional qualifiers defined in DSNZPARM in the **TIMESTAMP ARCHIVES** option (YES or NO) of the DSNTIPH panel, and Dyyddd can assume the format Dyyyddd if the **TIMESTAMP ARCHIVES** option is set to EXT (extended).

3.8.5 Image Copy Names

The names for image copy data sets are not defined by DB2. Each installation needs to define a standard naming convention to make these data sets distinct and significant. Table 7 on page 24 shows a sample naming structure for image copies.

Table 7. Sample Image Copy Names

hlq.wxiiydd.Thhmsst.sssss.Ammm

hlq VSAM catalog high level qualifier
w Copy type, P=Primary, S=Secondary copy
x Copy requirement, S=Standard, H=Critical
i Copy frequency, D=Daily, W=Weekly, M=Monthly
yyddd Date, yy=year, ddd=day of year
Thhmsst Time, hh=hour, mm=minute, ss=seconds
sssss table space or index space name
Ammm Data set identifier

Chapter 4. System Managed Storage Concepts and Components

This chapter is designed to familiarize the DB2 database administrator (DBA) with the concepts and components of system managed storage (SMS). The following topics are discussed:

- Background
- Evolution
- DFSMS/MVS components

4.1 Background

The continued growth of data requires the need for a more effective and efficient way of managing both data and the storage on which it resides. SMS was introduced in 1988 both as a concept and then as a group of products of MVS¹ to provide a solution for managing disk storage. Based upon user specifications, SMS can determine data placement, backup, migration, performance and expiration. The goals of SMS are:

- A reduction in the number of personnel required to manage that data, by allowing the system to manage as much as possible
- A reduction in labor-intensive related tasks of disk management, by centralizing control, automating tasks, and providing interactive tools
- A reduction in the necessity for user knowledge of placement, performance, and space management of data

4.2 Evolution

Although initially a concept, with a small number of products offering limited functionality, the introduction of DFSMS² has provided the functions needed for a comprehensive storage management subsystem, which provides:

- Management of storage growth
- Improvement of storage utilization
- Centralized control of external storage
- Exploitation of the capabilities of available hardware
- Management of data availability

With each subsequent release of the product, more features have been introduced that further exploit the concepts of SMS managed data, and this is likely to continue; for example, advanced functions for all types of VSAM files, which require the use of the extended addressability (EXT) attribute in the Data Class. It is therefore important to understand that those customers who have taken the time and effort to implement an SMS policy ultimately gain more from DFSMS enhancements than those who have not.

The characteristics of a DB2 system allows for the management of its data by SMS. However, there are considerations and choices that need to be made to tailor it to suit the individual customer's environment. These considerations are discussed in the following sections.

¹ The term MVS (OS/390) refers to the family of products which, when combined, provide a fully integrated operating system.

² The term DFSMS refers to the family of products which, when combined, provide a system managed storage environment.

4.3 DFSMS/MVS Components

DFSMS/MVS provides and enhances functions formerly provided by MVS/DFP, Data Facility Data Set Services (DFDSS), and the Data Facility Hierarchical Storage Manager (DFHSM). The product is now easier to install and order than the combination of the earlier offerings. This chapter describes the main components of the DFSMS/MVS family:

- DFSMSdfp
- DFSMSdss
- DFSMSHsm
- DFSMSrmm
- DFSMSopt

4.3.1 DFSMSdfp

The Data Facility Product (DFP) component provides storage, data, program, tape and device management functions.

DFP is responsible for the creation and accessing of data sets. It provides a variety of different access methods to organize, store and retrieve data, through program and utility interfaces to VSAM, partitioned, sequential, and direct access types.

DFP also provides the Interactive Storage Management Facility (ISMF) which allows the definition and maintenance of storage management policies interactively. It is designed to be used by both storage administrators and end users.

4.3.1.1 ISMF for the End User

Figure 8 on page 26 shows the ISMF primary option menu displayed for an end user. Various options allow the user to list the available SMS classes, display their attributes, and build lists of data sets based upon a selection criteria. These lists are built from VTOC or catalog information, and tailored using filtering, sorting, masking criteria. This panel is selected from the ISPF/PDF primary menu (dependent upon site customization).

```
ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.5
Enter Selection or Command ====>

Select one of the following options and press Enter:

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

Figure 8. ISMF Primary Option Menu for End Users

4.3.1.2 ISMF for the Storage Administrator

Figure 9 on page 27 shows the ISMF primary option menu displayed for a storage administrator. Options allow lists to be built from volume selection criteria (Storage Group), as well as the Management Class, Data Class, and Storage Class facilities allowing the individual to define, alter, copy, and delete SMS classes, volumes, and data sets. Again, these lists are built from VTOC or catalog information, and tailored in the same way. This panel is selected from the ISPF/PDF primary menu (dependent upon site customization).

```
ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.5
Enter Selection or Command ====>

Select one of the following options and press Enter:

0 ISMF Profile           - Specify ISMF User Profile
1 Data Set              - Perform Functions Against Data Sets
2 Volume                - Perform Functions Against Volumes
3 Management Class     - Specify Data Set Backup and Migration Criteria
4 Data Class           - Specify Data Set Allocation Parameters
5 Storage Class        - Specify Data Set Performance and Availability
6 Storage Group        - Specify Volume Names and Free Space Thresholds
7 Automatic Class Selection - Specify ACS Routines and Test Criteria
8 Control Data Set     - Specify System Names and Default Criteria
9 Aggregate Group      - Specify Data Set Recovery Parameters
10 Library Management  - Specify Library and Drive Configurations
11 Enhanced ACS Management - Perform Enhanced Test/Configuration Management
C Data Collection      - Process Data Collection Function
L List                 - Perform Functions Against Saved ISMF Lists
R Removable Media Manager - Perform Functions Against Removable Media
```

Figure 9. ISMF Primary Option Menu for Storage Administrators

4.3.2 DFSMSdss

The Data Set Services (DSS) component is a disk storage management tool. It can be invoked using ISMF. The following sections describe DSS capabilities.

4.3.2.1 Functionality

The DSS component is able to perform a variety of space management functions. Of these, the most common are listed below:

COMPRESS	Compresses partitioned data sets.
CONVERTV	Converts existing volumes to and from SMS management without data movement.
COPY	Performs data set, volume, and track movement, allowing the movement of data from one disk volume to another, including unlike device types.
COPYDUMP	Allows the generation of 1 to 255 copies of DFSMSdss produced dump data. The data to be copied can be tape or disk based, and copies can be written to tape or disk volumes.
DEFRAG	Reduces or eliminates free-space fragmentation on a disk volume.
DUMP	Performs the dumping of disk data to a sequential volume (either tape or disk). The process allows either:

Logical processing, which is data set oriented. This means it performs against data sets independently of the physical device format.

Physical processing, which can perform against data sets, volumes, and tracks, but moves data at the track-image level.

PRINT	Used to print both VSAM and non-VSAM data sets, track ranges, or a VTOC.
RELEASE	Releases allocated but unused space from all eligible sequential, partitioned, and extended format VSAM data sets.
RESTORE	Data can be restored to disk volumes from DFSMSdss produced dump volumes, as individual data sets, an entire volume, or a range of tracks. Again, <i>logical</i> or <i>physical</i> processing can be used.

4.3.2.2 Filtering

The DSS component uses a filtering process to select data sets based upon specified criteria. These include:

- Fully or partially qualified data set name
- Last reference data
- Size of data sets
- Number of extents
- Allocation type (CYLS, TRKS, BLKS)
- SMS Class

4.3.2.3 Converting Data to SMS

Data sets can be converted by data movement using the COPY or DUMP and RESTORE functions. However, it is possible and quite common to convert data in place, without the need for movement, by using the CONVERTV command. This allows:

- The preparation of a volume for conversion by preventing new data set allocations.
- Conversion simulation, to test for any data sets that do not match the expected criteria.
- Actual conversion of a volume to SMS management.
- Converting a volume from SMS management, for example, as part of a Disaster Recovery scenario.

4.3.3 DFSMSHsm

The Hierarchical Storage Manager (HSM)³ component provides availability and space management functions. For more information, please refer to the *DFSSMSHsm Primer*, SG24-5272.

HSM improves productivity by making the most efficient use of disk storage, primarily by making better use of the primary volumes. It performs both availability management and space management automatically, periodically, and

³ Other Vendor products exist which have similar capabilities, and can be used in place of HSM, although this may restrict full exploitation of ISMF.

by issuing specific commands when manual operations are appropriate. Volumes can be:

- Managed by SMS. In this case the Storage Group definitions controls HSM initiated automatic functions, depending upon the appropriate Management Class of the data set.
- Managed by HSM. These volumes are commonly known as primary or level 0. Here each volume is defined to HSM individually by the ADDVOL parameter and governed accordingly.
- Owned by HSM. These incorporate migration level 1 (ML1), migration level 2 (ML2), backup, dump and aggregate volumes, and are a combination of disk and tape. Alternate tape copies can be made of ML2 and backup tapes for off-site storage. Also, spill volumes can be generated from backups and ML2 volumes to improve tape usage efficiency.

Figure 10 shows the relationship between the main components of the HSM environment.

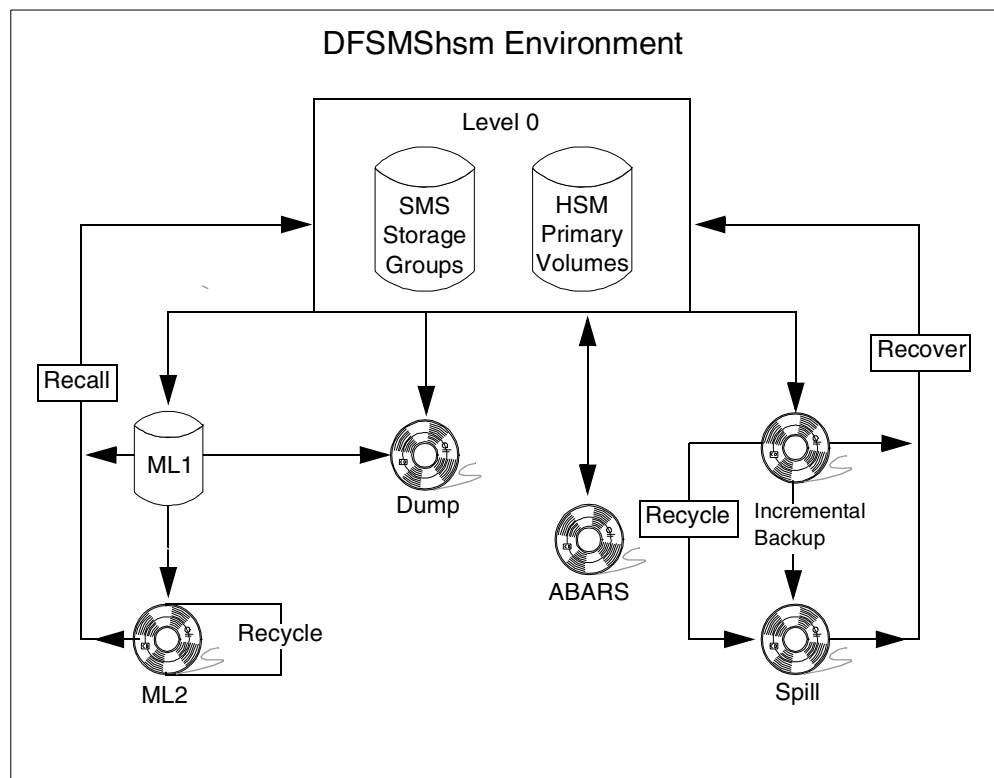


Figure 10. DFSMShsm Components

4.3.3.1 Space Management

On a daily basis, HSM performs automatic space management, depending upon whether the volume is part of an SMS Storage Group or managed individually by HSM. Space management consists of the following functions:

- Migration

This is the process of moving eligible, inactive, cataloged data sets to either ML1 or ML2 volumes, from primary volumes (Level 0). This is

determined by either the Management Class for SMS managed data sets, or set by the ADDVOL parameter for HSM managed. It can also be controlled in combination with volume thresholds set by the storage administrator. Data sets may be migrated to ML1 (normally disk)⁴, after a period of inactivity, and then onto ML2 (tape) following a further period of non-usage. It is feasible, and maybe more appropriate in certain cases, to migrate directly to ML2. Additionally, there is an *interval migration* process which can be used to free space on volumes during times of high activity.

- Expiration processing

This is based upon the inactivity of data sets. For HSM managed volumes, all data sets are treated in the same way on each volume. For SMS managed volumes, the expiration of a data set is determined by the Management Class attributes for that data set.

- Release of unused space

HSM can release over-allocated space of a data set for both SMS managed and non-SMS managed data sets.

- Recall

There are two types of recall:

- Automatically retrieving a data set when a user or task attempts to access it
- When the HRECALL command is issued

All recalls are filtered; if the data set is SMS managed, then SMS controls the volume selection. If the data set is non-SMS, HSM directs the volume allocation. However, it is possible for the storage administrator to control a recall and place it on an appropriate volume or Storage Group if required.

4.3.3.2 Availability Management

The purpose of availability management is to provide backup copies of data sets for recovery scenarios. HSM can then restore the volumes and recover the data sets when they are needed.

- Incremental backups

This is the process of taking a copy of a data set, depending upon whether it has changed (open for output; browse is not sufficient), since its last backup. For SMS managed volumes, HSM performs the backup according to the attributes of the Management Class of the individual data set. For non-SMS managed volumes, HSM performs the backup according to the ADDVOL definition.

- Full volume dumps

A full volume dump backs up all data sets on a given volume, by invoking DSS. HSM Dump Classes exist which describe how often the process is activated, for example, daily, weekly, monthly, quarterly or annually.

- Aggregate backup

ABARS is the process of backing up user defined groups of data sets that are business critical, to enable recovery should a scenario arise.

⁴ Very large data sets, in excess of 64,000 tracks, cannot be migrated to disk: they must be migrated to migration level 2 tape.

- Recovery

Recovery can be either at the individual data set level or to physically restore a full volume. This is applicable for both SMS and non-SMS managed data sets.

Note that full exploitation of this component requires the use of the DSS and Optimizer components of DFSMS.

4.3.4 DFSMSrmm

The Removable Media Manager (RMM)⁵ component provides facilities for tape and cartridge formats, including library, shelf, volume and data set level management.

Tapes are an effective media for storing many types of data, especially backup and archive copies of disk data sets. However, tapes must be mounted to be used, and the capacity of tapes is often not fully used. DFSMS/MVS can be used to assist in more effective use of tape resources.

With SMS, a group of disks can be defined to act as a buffer for tape drives, and allow HSM to manage writing the tape volumes and data sets on those volumes. DFSMS/MVS permits the use of system managed tape volumes, and RMM can be used to manage the inventory of system managed and non system managed tapes.

For further information on this topic, see the *DFSMS/MVS V1R4 DFSMSrmm Guide and Reference*, SC26-4931-05.

4.3.5 DFSMSopt

The Optimizer (OPT) component is one of the supporting products of DFSMS/MVS, and is a separately orderable feature. It provides data analysis and simulation information, which assists in improving storage usage and reducing storage costs. It can be used to:

Monitor and tune HSM functions.

Create and maintain a historical database of system and data activity.

Perform analysis of Management Class, and Storage Class policies, including simulation, costing analysis, and recommendations for data placement.

Identify high I/O activity data sets, and offer recommendations for data placement.

Monitor storage hardware performance of subsystems and volumes, including I/O rate, response time, and caching statistics.

Fine tune the SMS configuration, by presenting information to help understand how current SMS practices and procedures are affecting the way data is managed.

⁵ Other Vendor products exist which have similar capabilities, and can be used in place of RMM, although this may restrict full exploitation of ISMF.

- Simulate potential policy changes and understand the costs of those changes.
- Produce presentation quality charts.

For more information on the DFSMS/MVS Optimizer Feature, see the following publications:

- *DFSMS Optimizer V1R2 User's Guide and Reference*, SC26-7047-04
- *DFSMS Optimizer: The New HSM Monitor/Tuner*, SG24-5248

4.3.6 SMF Records 42(6)

DFSMS statistics and configuration records are recorded in SMF record type 42.

The type 42, subtype 6, SMF record provides information about data set level performance. DFSMS must be active, but the data set does not need to be SMS managed. Two events cause this record to be generated: data set close time and each type 30 interval record being written.

You can use DFSMSopt or any SMF specialized package to format and display this useful record. For instance, you can start by looking at the list of the first 20 data sets in terms of activity rate at the specified interval and verify that the most accessed DB2 data sets are performing as expected both in terms of I/O and usage of DB2 buffer pools.

Also, accesses to critical data sets can be tracked periodically by data set name. Their performance can be mapped against the DB2 PM accounting to determine detailed characteristics of the I/O executed, and to verify cache utilization.

4.4 Benefits

A summary of the benefits of SMS follows:

- Simplified data allocation

SMS enables users to simplify their data allocations. Without using SMS, a user would have to specify the unit and volume on which the system should allocate the data set. In addition, space requirements would need to be calculated and coded for the data set. With SMS, users can let the system select the unit, volume and space allocation. The user therefore, does not need to know anything about the physical characteristics of the devices in the installation.

- Improved allocation control

Free space requirements can be set using SMS across a set of disk volumes. Sufficient levels of free space can be guaranteed to avoid space abends. The system automatically places data on a volume containing adequate freespace.

- Improved performance

SMS can assist in improving disk I/O performance, and at the same time reduce the need for manual tuning by defining performance goals for each class of data. Cache statistics, recorded in system management facilities (SMF) in conjunction with the Optimizer feature, can be used to assist in evaluating performance. Sequential data set performance can be improved by using extended sequential data sets. The DFSMS environment makes the most effective use of the caching abilities of disk technology.

- Automated disk space management

SMS has the facility to automatically reclaim space which is allocated to old and unused data sets. Policies can be defined that determine how long an unused data set are allowed to reside on level 0 volumes (active data). Redundant or expired data can be removed by the process of migration to other volumes (disk or tape), or the data can be deleted. Allocated but unused space can be automatically released, which is then available for new allocations and active data sets.

- Improved data availability management

With SMS, different backup requirements can be provided for data residing on the same primary volume. Therefore, all data on a single volume can be treated independently. The HSM component can be used to automatically back up data. The ABARS facility can be used to group data sets into logical components, so that the group is backed up at the same time, allowing for recovery of an application.

- Simplified data movement

SMS permits the movement of data to new volumes without the necessity for users to amend their JCL. Because users in a DFSMS environment do not need to specify the unit and volume which contains their data, it does not matter to them if their data resides on a specific volume or device type. This allows the replacement of old devices with minimum intervention from the user.

System determined block sizes can be utilized to automatically reblock sequential and partitioned data sets that can be reblocked.

Chapter 5. Storage Management with DFSMS

This chapter is designed to familiarize the DB2 administrator with the functionality of SMS. This chapter covers the following topics:

- The SMS base configuration
- The automatic class selection routine
- The SMS constructs
- Objectives of using SMS with a DB2 system
- How SMS can handle DB2 special data

Further information can be found in the following publications:

- *DFSMS/MVS V1R4 Implementing System-Managed Storage*, SC26-3123
- *MVS/ESA SML: Managing Data*, SC26-3124
- *MVS/ESA SML: Managing Storage Groups*, SC26-3125

5.1 Introduction

SMS manages an installation's data and storage requirements according to the storage management policy in use. Using ISMF, the storage administrator defines an installation's storage management policy in an SMS configuration, which consists of:

- The base configuration
- Class and Storage Group definitions

5.1.1 Base Configuration

The base configuration contains defaults and identifies the systems which are SMS managed. The information is stored in SMS control data sets, which are VSAM linear data sets. These consist of:

- Source control data set (SCDS)
- Active control data set (ACDS)
- Communications data set (COMMDS)

Source Control Data Set

The source control data set (SCDS) contains the information that defines a single storage management policy, called an SMS configuration. More than one SCDS can be defined, but only one can be used to activate a configuration at any given time.

Active Control Data Set

The active control data set (ACDS) contains the SCDS that has been activated to control the storage management policy for the installation. When a configuration is activated, SMS copies the existing configuration from the specified SCDS into the ACDS. While SMS uses the ACDS, the storage administrator can continue to create copies of the ACDS, modify, and define a new SCDS if desired.

Communications Data Set

The communications data set (COMMDS) holds the name of the ACDS and provides communication between SMS systems in a multisystem environment. The COMMDS also contains statistics on the SMS, and MVS status for each SMS volume, including space.

5.1.2 Class and Storage Group Definitions

The storage management policies are defined to the system by use of classes. Data sets have classes assigned to them. These are *Data Class*, *Storage Class*, and *Management Class*, and they determine the volume allocation which forms the *Storage Group*. The classes manage the data sets, and the Storage Groups manage the volumes on which the data sets reside.

Figure 11 on page 36 shows the steps taken by an installation to implement an active SMS configuration.

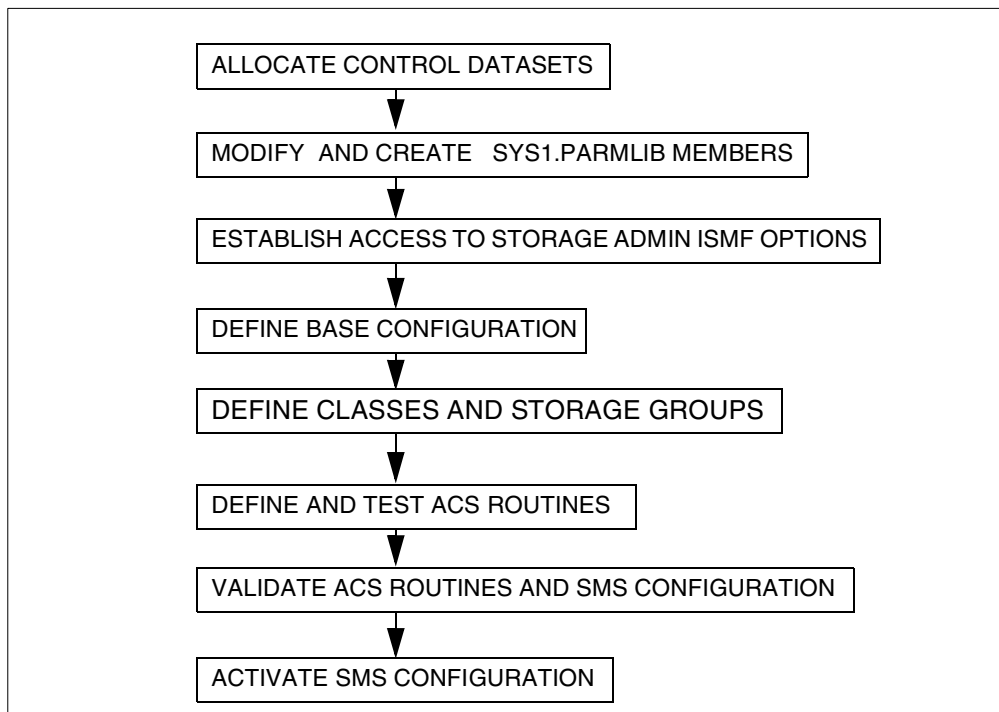


Figure 11. Implementing an SMS Configuration

5.2 Automatic Class Selection Routines

Automatic Class Selection (ACS) routines are the mechanism for assigning SMS classes and Storage Groups. They determine the placement of all new data set allocations, plus allocations involving the copying, moving, recalling, recovering, and converting of data sets. ACS routines are written in a high level REXX style programming language. If SMS is activated, all new data set allocations are subject to automatic class selection.

There are four ACS routines. Aggregate groups also exist, but for the purposes of this book are only mentioned where relevant. The ACS routines are executed in the following order:

1. Data Class—data definition parameters.
2. Storage Class—performance and accessibility requirements.
3. Management Class—migration, backup and retention attributes.
4. Storage Group—candidate allocation volumes.

Because data set allocations, whether dynamic or through JCL, are processed through ACS routines, installation standards can be enforced on those allocations on both SMS and non-SMS managed volumes. ACS routines permit user defined specifications for Data, Storage, and Management Classes, and requests for specific volumes, to be overridden, thereby offering more control of where data sets are positioned within the system.

For a data set to qualify for SMS management, it must satisfy the Storage Class component. If a *valid* Storage class is assigned, then the data set is managed by SMS, and proceeds via the Management Class and Storage Group routines before being allocated on a SMS volume. If a *null* Storage class is assigned, the data set exits from the process, is not managed by SMS, and is allocated on a non-SMS volume.

Figure 12 on page 37 shows the execution process for defining data sets in an MVS system (with SMS active).

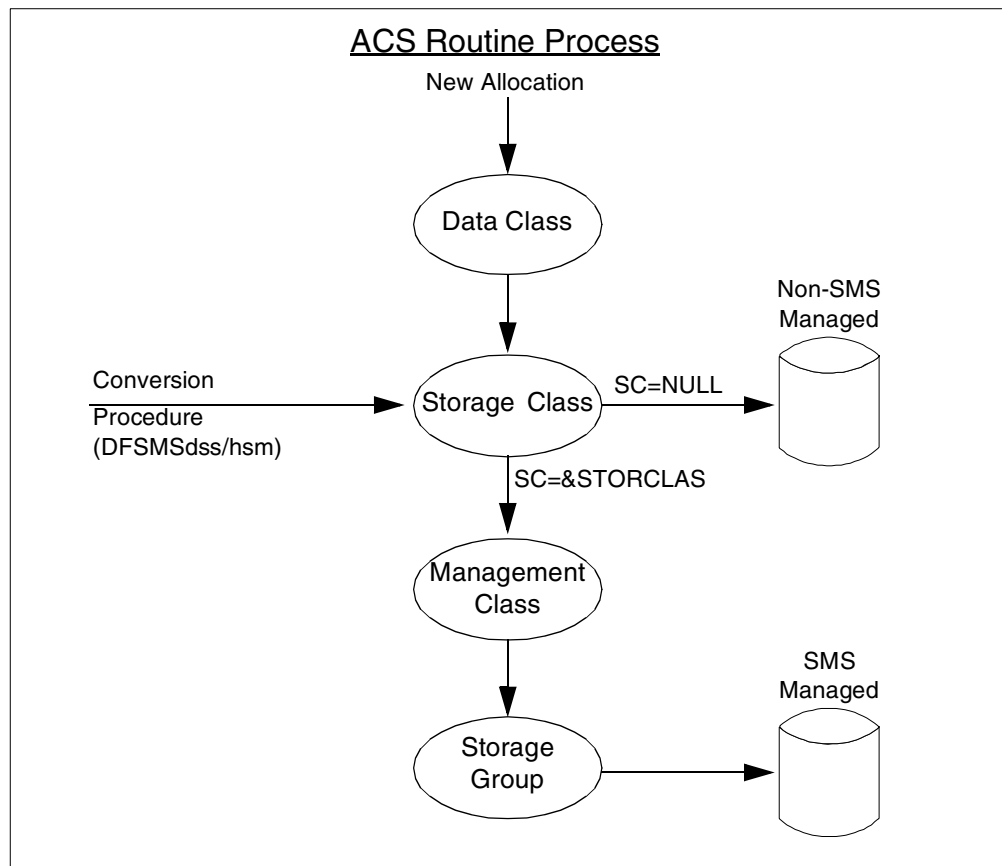


Figure 12. ACS Routine Execution Process

5.3 SMS Classes

The storage administrator uses ISMF to create an ACS routine for each of the three types of classes and one to assign the Storage Groups. These routines, used together with the Data Class, Storage Class, Management Class, Storage Group definitions, and the base configuration, define an installation's SMS configuration.

Figure 13 on page 38 shows the relationship of each of the four constructs which make up the SMS ACS routine environment. The following sections describe each of the classes in more detail.

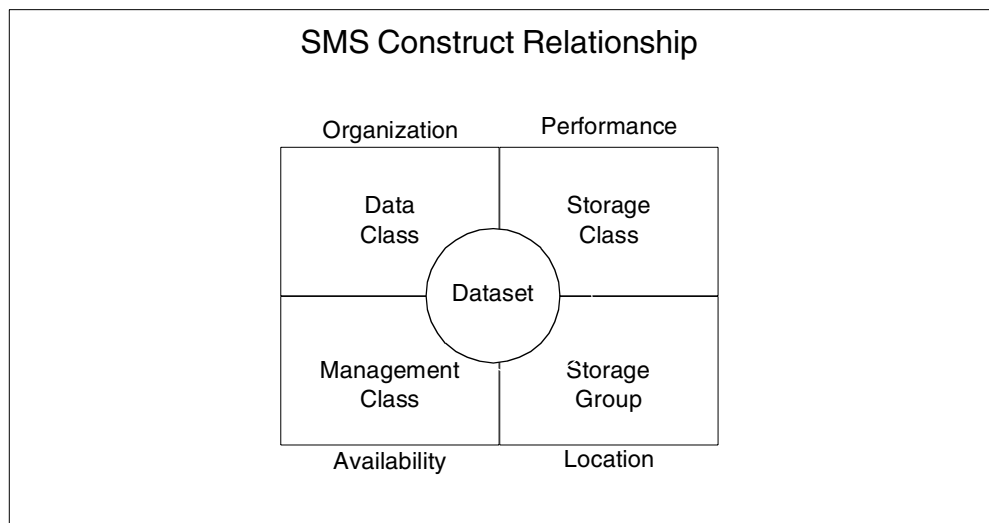


Figure 13. SMS Construct Relationship

5.3.1 Data Class

5.3.1.1 Description

Formerly, when allocating new data sets, users were required to specify a full set of attributes. Even for multiple allocations, repetitive coding was required. With the introduction of SMS, this process is now simplified, and also helps to enforce standards by use of the Data Class.

A Data Class is a named list of data set allocation and space attributes that SMS assigns to a data set when it is created. It contains associated default values set by the storage administrator. Data Classes can be assigned implicitly through the ACS routines or explicitly using the following:

- JCL statement. The user only need specify the relevant Data Class keyword such as DATACLAS=DCLIB
- TSO/E ALLOCATE command DATACLAS(DCLIB)
- IDCAMS ALLOCATE and DEFINE commands
- ISPF/PDF Data set allocation panel

When specified, the Data Class allocates a data set in a single operation. Any disk data set can be allocated with a Data Class whether managed by SMS or not. Tape data sets cannot be allocated with Data Classes.

User defined allocations take precedence over default Data Classes. For example, if a Data Class specifies an LRECL of 80 bytes, and the JCL allocation specifies an LRECL of 100 bytes, then 100 bytes are allocated. If the Data Class is altered by the storage administrator, attributes previously allocated by the Class remains unchanged. Alterations are only be honored for *new* allocations.

5.3.1.2 Planning for Implementation

To identify and reference a particular Data Class, a unique one to eight character name is used, for example, DCDBKSDS.

For each group of data sets that have similar attributes, a Data Class can exist, but is not mandatory. An example where it could be used is with DB2 tablespaces, as they have identical allocation characteristics.

Prior to the definition of Data Classes, an analysis of common data types needs to be undertaken. This should include deciding whether to use ACS routines only for their allocation, or allow users (in this case, the DBA) to assign them as well. There may be a requirement to standardize naming conventions, and agree upon default space allocations.

Attributes include many of the data set characteristics specified on JCL statements, and IDCAMS DEFINE commands. Only those applicable to a particular data set type should be coded, all others should be left blank. Table 8 on page 39 shows a list of attributes for consideration.

Table 8. Data Class Attributes

ATTRIBUTE	COMMENT
Data set Organization	<ul style="list-style-type: none"> - VSAM type (KSDS, ESDS, RRDS or LINR) - Non VSAM type (Sequential, partitioned) - Record format (RECFM) - Logical record length (LRECL) - Key Length (VSAM)
Space requirements	<ul style="list-style-type: none"> - Average record length value - Size of primary allocation - Size of secondary allocation - Number of directory blocks, if a library
VSAM, data and volume specifics	<ul style="list-style-type: none"> - Size of Control Interval and Control Area - Percentage freespace - Replicate - Imbed - Share options—volume count - Backup while open - Extended addressability - Reuse - Space constraint relief - Spanned/non spanned - Initial load (speed and recovery)

5.3.2 Storage Class

5.3.2.1 Description

Prior to SMS, critical and important data sets that required improved performance or availability were allocated to specific volumes manually. Data sets that required low response times were placed on low activity volumes, where caching

was available. SMS allows the separation of performance and service level of data sets by use of the Storage Class.

A Storage Class construct details the intended performance characteristics required for a data set assigned to a given class. The response times set for each Storage Class are target response times for the disk controller to achieve when processing an I/O request. It decides if the volume should be chosen by the user or by SMS. Also, it determines whether SMS, when allocating space on the first volume of a multi-volume data set, should allocate space on the remaining volumes as well. The assignment of a Storage Class does not guarantee its performance objective, but SMS selects a volume that offers performance as close as possible. Only SMS managed data sets use Storage Classes. Changes in a Storage Class applies to the data sets that are already using that class.

Storage Classes can be assigned implicitly through the ACS routines, or explicitly by using the following:

- JCL statement. The user only needs to specify the relevant Data Class keyword, such as `STORCLAS=SCDBGS`.
- DSS COPY and RESTORE commands.
- TSO/E ALLOCATE command such as `STORCLAS(SCDBGS)`.
- IDCAMS ALLOCATE and DEFINE commands.
- ISPF/PDF data set allocation panel.

Unlike the Data Class, users *cannot* override individual attribute values when allocating data sets.

5.3.2.2 Planning for Implementation

For each group of data sets that have similar performance objectives, a Storage Class can exist. To identify and reference a particular Storage Class, a unique one to eight character name is used, for example, SCDBFAST.

Consideration needs to be given as to whether a user is authorized to select a specific volume within a Storage Group, which is governed by the *Guaranteed Space* parameter. This is an arrangement which needs to be agreed upon between the storage administrator and the DBA.

An example of the use of this parameter is in the allocation of the *Active logs and BSDS*, where these data sets have critical performance and availability requirements. The DBA should be allowed to allocate them on specific volumes, which is especially important for dual logging capability, to ensure that the logs are allocated on separate volumes. After being defined, these data sets are rarely redefined.

Table 9 on page 40 provides a list of attributes for consideration.

Table 9. Storage Class Attributes

ATTRIBUTE	COMMENT
Performance objectives	<ul style="list-style-type: none"> - Direct bias - Direct millisecond response - Initial access response - Sequential millisecond response - Sustained data rate

ATTRIBUTE	COMMENT
Availability objectives	- Accessibility - Availability - Guaranteed space - Guaranteed synchronous write
Caching	- Weighting

5.3.3 Management Class

5.3.3.1 Description

Prior to SMS, DFHSM managed the data sets at volume level, applying a standard management criteria for all data sets on a given volume. Although this is still applicable for non-SMS managed data sets, with the introduction of SMS, the control is carried out at *data set level* by use of the Management Class.

The Management Class is *only* assigned if the Storage Class construct selects a valid Storage Class, as can be seen in Figure 12 on page 37. For each data set, it consists of attributes that determine the necessary control of:

- Retention
- Backup
- Migration
- Expiration
- Management of generation data set groups (GDGs) and their data sets (GDSs)
- Space release

When assigned to a data set, the Management Class expands on attributes previously specified by JCL, IDCAMS DEFINE and DFHSM commands.

If an attribute is altered on a particular Management Class, the change is applied to previously created data sets which have the same attribute, when the next management cycle commences. A default Management Class can be specified to cater to those groups of data sets that do not belong to a Management Class, thereby ensuring that all SMS managed data sets have a set level of availability.

5.3.3.2 Planning for Implementation

A Management Class should be generated for each group of data sets that have similar availability requirements. To identify and reference a particular Management Class, a unique one to eight character name should be used, for example, MCDB21.

Before defining Management Classes, a number criteria should be established by the storage administrator. Based upon this information, the storage administrator defines Management Classes that provide a centralized storage environment. This includes the decision on whether to allow users the ability to assign Management Classes explicitly by using JCL, as well as implicitly by the ACS routines.

Because most production database data has specialized backup and recovery requirements, it is recommended that standard DB2 system utilities be used to perform backup and recovery. However, consider using DFSMSdss or

DFSMSHsm's automatic backup services, supported by concurrent copy, to help with *point of consistency* backups.

It is not advisable to use HSM to manage most production databases. Therefore, use a NOMIGRATE Management Class for this type of data. This prevents HSM space and availability management from operating. Specifically, AUTO BACKUP is set to NO so that HSM does not back up the data set, ADMIN OR USER COMMAND BACKUP is set to NONE to prevent manual backup, and expiration attributes are set to NOLIMIT to prevent data set deletion.

Although production database data does receive automatic backup service, DFSMSdss can be set to run concurrent copy for production databases. ACCESSIBILITY may be set to CONTINUOUS for Storage Classes assigned to production databases to ensure that the data set is allocated behind a storage control with concurrent copy support.

Testing or end user databases that have less critical availability requirements can benefit from system management using HSM. Additionally, selected data types for production systems could also be effectively managed using HSM.

For DB2 systems, it is possible to manage archive logs and image copies with HSM. These data sets can be retained on primary devices for a short period of time, and then migrated directly to tape (ML2).

HSM uses the same ML2 tape until it is full. Therefore, unless another mechanism is used, to avoid placing consecutive archive logs on the same ML2 tape, ensure that the storage administrator defines the HSM parameter SETSYS(PARTIALTAPE(MARKFULL)), so it uses a new ML2 tape each time space management is executed.

Requirements should be agreed upon by the storage administrator and the DBA. Table 10 on page 42 displays a list of attributes for consideration.

Table 10. Management Class Attributes

ATTRIBUTE	COMMENT
Space management	- Release unused space in the data set (applies to non-VSAM only).
Expiration	- Delete data set after a number of days or a date. - Delete after a number of days of non usage. - Use of Retention or Expiration periods.
Migration	- What method a data set can migrate (Command or automatically, or both). - Number of days non-usage on level 0 volumes before migration commences. - Number of days non-usage on level 1 volumes before migration commences.

ATTRIBUTE	COMMENT
Backup	-Who can back up the data (the storage administrator or the user, or both). - If automatic backup should be taken for a data set. - Backup frequency (number of days between backups). - Number of backup versions (data set exists). - Number of backup versions (data set deleted). - Retention of backups once a data set has been deleted. - Backup copy type (incremental, full volume dump).

5.3.4 Storage Group

5.3.4.1 Description

Prior to SMS, disk storage was maintained as individual volumes, requiring manual intervention to prevent volumes from filling up, and to prevent I/O bottlenecks. SMS significantly improves the management of disk by building on DFHSM capabilities to pool volumes together in Storage Groups.

A Storage Group is a pool of disk volumes upon which SMS managed data sets are placed. These groups are normally transparent to users. A data set is placed on an appropriate volume within a Storage Group depending upon the Storage Class, volume, Storage Group status, and available free space. New data set allocations can be directed to as many as 15 Storage Groups, although only one Storage Group is finally selected.

Each Storage Group has attributes which determine a range of characteristics for the volumes in that group. This includes backup, migration, and space thresholds.

A volume can belong to one of the following main Storage Group types; POOL, DUMMY or VIO. Three other types also exist; OBJECT, OBJECT BACKUP and TAPE, although these are not as commonly used.

Storage Groups:

- Cannot share a volume.
- Cannot share data sets.
- Must contain whole volumes.
- Must contain volumes of the same device geometry.
- Can contain multi-volume data sets.
- Must contain a VTOC and a VVDS.

SMS selects the volumes used for data set allocation by building a list of all volumes from the Storage Groups assigned by the ACS routine. Volumes are then either removed from further consideration or flagged as primary, secondary, or tertiary volumes.

Primary volumes meet the following criteria:

- SMS Storage Group and volume status of ENABLED
- MVS volume status of ONLINE
- Requested initial access response time (IART)

- The number of volumes in the Storage Group satisfies the volume count
- Accessibility requested
- Availability (dual copy or RAMAC) requested
- The volume was explicitly requested and guaranteed space is YES
- Sufficient free space to perform the allocation without exceeding the high threshold
- Volumes fall within a pre-determined range of millisecond response times based on the request
- The volume supports extended format if EXT=PREFERRED or REQUIRED is requested in the data class

Candidates for secondary volumes are primary volumes that, in addition:

- Are at or above high threshold, or exceed high threshold following the allocation.
- Are quiesced, or the Storage Group to which the volume belongs is quiesced.
- Did not meet the requested millisecond response time.
- Did not meet the accessibility request of standard or preferred.
- Did not meet the IART of greater than zero.
- Did not meet the availability request of standard or preferred.

When a Storage Group does not contain enough volumes to satisfy the volume count, all volumes in the Storage Group are flagged tertiary. Tertiary volumes are only selected when there are no primary or secondary volumes and the request is for a non-VSAM non-GUARANTEED SPACE request.

After the system selects the primary allocation volume, that volume's associated Storage Group is used to select any remaining volumes requested.

5.3.4.2 Planning for Implementation

It is important that unique Storage Groups are used for production databases and recovery data sets, because of their critical status. Appropriate groups should be defined by the storage administrator to prevent automatic migration (AUTO MIGRATE), and automatic backup (AUTO BACKUP). However non-production databases should be considered for placement on standard primary volumes (possibility shared with other data types), as their availability is normally not as critical.

DB2 allows the DBA to define a collection of volumes that DB2 uses to find space for new data set allocation, known as STOGROUPs. When converting DB2 databases to SMS, and DB2 STOGROUPs are used to manage DB2 database data, one way is to design the SMS Storage Groups so they are compatible with existing STOGROUP definitions.

Once the conversion is complete, it is recommended that SMS be used, rather than DB2 to allocate databases. To allow SMS control over volume selection, define DB2 STOGROUPs with VOLUMES(*).

Electing DB2 to select the volume requires assigning a Storage Class with guaranteed space. However, guaranteed space reduces the benefits of SMS allocation, so this approach is not recommended.

If you do choose to use specific volume assignments, additional manual space management must be performed. Unlike non-SMS, SMS does not retry to skip a volume that cannot satisfy the requested space. Free space must be managed for each individual volume to prevent failures during the initial allocation and extension. This will generally require more time for space management, and will result in more space shortages. Guaranteed space should only be used where the space needs are relatively small and do not change.

To identify and reference a particular Storage Group, a unique one to eight character name is used, for example, SGDBFAST.

Table 11 provides a list of attributes for consideration.

Table 11. Storage Group Attributes

ATTRIBUTE	COMMENT
Auto Migrate	Specifies whether migration should be permitted on this Storage Group.
Auto Backup	Specifies whether backup should be performed on this Storage Group.
Auto Dump	Specifies whether full volume dumping should be performed on this Storage Group.
Migration Threshold (High and low)	A percentage figure which when exceeded forces migration to occur. Likewise, a percentage figure, at which migration stops.
Dump Classes	Specifies the frequency of auto dumping, if required.
Guaranteed Backup Frequency	Specifies the maximum number of days that can elapse between backups. NOLIMIT indicates data sets in the Storage Group are backed up according to their Management Class.

For further information on all SMS Class attributes and definitions, see *DFSMS/MVS DFSMSdfp Storage Administration Reference*, SC26-4920.

5.3.4.3 Mapping Devices to Storage Groups for Performance

From the performance point of view, migrating to SMS offers the opportunity to *automatically set DB2 allocations to predefined disk areas*. Each storage server offers a predetermined level of parallel access. For example, the RVA Turbo allows eight concurrent data transfers to and from the host. DB2 administrators and storage administrators can distribute the Storage Groups *to maximize the use of parallel capability offered by each storage server type*. For example, with RVA servers, a Storage Group should have a multiple of eight volumes per RVA and be spread over several RVAs. A performance oriented small Storage Group could have just eight volumes defined per RVA (in groups of two by LCU, and by RVA) and spread over the number of RVAs required for best parallel access.

SMS offers a performance oriented automated allocation mechanism provided that a defined Storage Group logical topology matches the current installed

hardware capabilities. For a few specific and exceptional cases, the storage class GUARANTEED SPACE option can be used. As the Storage Group definition exists only in SMS tables, its logical mapping onto volumes can be *redistributed* when a hardware change occurs, without any DB2 application outage, provided that DB2 and storage administrators act in concert (in particular for allocating new DB2 objects). Notice that *redefining a Storage Group does not require application outage*.

5.4 Naming Standards

To assist in the successful implementation of SMS, a vital requirement is that of generating and adhering to a constructive and meaningful naming standard policy. The more formal the policy, the easier it is to maintain the ACS routines. This can be of particular use in the formation of policies for Data and Management Classes.

These policies can:

- Simplify service-level assignments to data.
- Facilitate writing and maintaining ACS routines
- Allow data to be mixed in a system-managed environment while retaining separate management criteria
- Provide a filtering technique useful with many storage management products
- Simplify the data definition step of aggregate backup and recovery support

Most naming conventions are based on the high level qualifier (HLQ) and low level qualifier (LLQ) of the data set name. Additional levels of qualifiers can be used to identify generation data sets and databases. They can also be used to help users to identify their own data. It must be stressed that each installation has different naming conventions, and therefore requires careful planning.

DB2 systems generate their own data set names, so it is necessary to ensure that the storage administrator understands the implications, and is able to define a policy and build the ACS routines so they incorporate this feature.

5.5 Examples

Examples of SMS constructs for DB2 data sets are described in this book:

- Chapter 6, “Managing DB2 Databases with SMS” on page 47.
- Chapter 7, “Managing DB2 Recovery Data Sets with SMS” on page 63.

A test implementation of these examples is shown in:

- Appendix A, “Test Cases for DB2 Table Space Data Sets” on page 161.
- Appendix B, “Test Cases for DB2 Recovery Data Sets” on page 185.

Chapter 6. Managing DB2 Databases with SMS

This chapter describes DB2 databases from the point of view of their attributes for SMS management, and provides examples for these databases. Due to their stricter availability requirements, the DB2 system databases are analyzed separately.

This chapter includes examples of SMS Data, Storage, and Management Classes for DB2 table spaces. These examples are applied to DB2 system tablespaces and to DB2 application table spaces grouping them in four different user environments.

6.1 SMS Examples for DB2 Databases

The following examples are provided to show how SMS can be used to manage DB2 table spaces. These examples do not show all possibilities SMS offers to an installation. Each installation can review these examples and create those classes that best suit its environment. The examples shown here are extracted and adapted from *DFSMS/MVS Implementing System-Managed Storage*, SC26-3123.

Naming Convention:

The following naming structure is used for the example SMS constructs. Each name has a two-character SMS construct identifier, two characters 'DB' to identify them as SMS constructs used for DB2, followed by a variable length (aaaa) text. This naming convention is:

DCDBaaaa	SMS Data Classes for DB2
SCDBaaaa	SMS Storage Classes for DB2
MCDBaaaa	SMS Management Classes for DB2
SGDBaaaa	SMS Storage Groups for DB2

6.1.1 Using ISMF to Display SMS Constructs

A DB2 administrator can use ISMF to access and examine the different SMS constructs in the installation. A storage administrator uses ISMF to create and to manage the SMS constructs. Figure 14 on page 48 shows how to display the active Data Class DCDB2.

The options available on the DATA CLASS APPLICATION SELECTION panel are dependent on the authorization of the user. Only a user authorized to manage SMS constructs is allowed to define or alter them. Other users may only have options 1 (List) and 2 (Display) available.

6.1.2 SMS Data Class

All DB2 table spaces, either for system or for user data, have exactly the same attributes. One Data Class can be defined for all these data sets. The Data Class allows an override of space parameters (primary and secondary allocation quantity) because those will be different for each table space. Figure 15 on page 48 shows some information from the Data Class DCDB2, a Data Class example for DB2 table spaces and index spaces.

```

DATA CLASS APPLICATION SELECTION
Command ==>

To perform Data Class Operations, Specify:
CDS Name . . . . . 'ACTIVE'
                                     (1 to 44 character data set name or 'Active' )
Data Class Name . . DCDB2           (For Data Class List, fully or partially
                                     specified or * for all)

Select one of the following options :
2 1. List   - Generate a list of Data Classes
   2. Display - Display a Data Class
   3. Define - Define a Data Class
   4. Alter  - Alter a Data Class

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

```

Figure 14. Display a Data Class

```

Data Class Name : DCDB2

Description : DATA CLASS FOR DB2 TABLESPACES

Recorg . . . . . : LS
Recfm . . . . . :
Lrecl . . . . . :
Keylen . . . . . :
Keyoff . . . . . :
Space Avgrec . . . . . : M
  Avg Value . . . . . : 1
  Primary . . . . . : 1
  Secondary . . . . . : 1
  Directory . . . . . :
Retpd Or Expdt . . . . . :
Volume Count . . . . . : 1
  Add'l Volume Amount . . :
Imbed . . . . . :
Replicate . . . . . :
Ctsize Data . . . . . : 4096
% Freespace CI . . . . . :
  CA . . . . . :
Shareoptions Xregion . . : 3
  Xsystem . . . . . : 3
Compaction . . . . . :

```

Figure 15. Data Class DCDB2

6.1.3 SMS Storage Class

Some DB2 installations may use only one Storage Class for DB2, but in others, the DB2 administrators may require several Storage Classes for DB2 table spaces. Table 12 on page 49 shows four examples. These are:

SCDBMED This Storage Class is intended for the majority of table spaces. It provides good performance and good availability.

- SCDBFAST** This Storage Class is intended for table spaces belonging to applications requiring performance. It provides high performance and good availability.
- SCDBCRT** This Storage Class is intended for table spaces belonging to critical applications. It provides high performance and continuous availability. SMS attempts to place these table spaces on disks with dual copy or on RAID.
- SCDBTEST** This Storage Class is intended for environments with lower requirements. Examples are test systems, development systems, and data warehouse environments. These table spaces will have average performance and average availability.

Table 12. SMS Storage Classes for DB2 Databases

	SCDBMED	SCDBFAST	SCDBCRT	SCDBTEST
DIRECT RESPONSE (MSEC)	10	5	5	20
DIRECT BIAS			Yes	
SEQUENTIAL RESPONSE (MSEC)	10	5	5	20
SEQUENTIAL BIAS				
SUSTAINED DATA RATE (MB/sec)	10	20	20	10
AVAILABILITY ^a	Preferred	Preferred	Continuous	Standard
ACCESSIBILITY ^b	Standard	Standard	Standard	Standard
GUARANTEED SPACE	No	No	No	No
GUARANTEED SYNCHRONOUS WRITE	No	No	No	No
CACHE SET NAME				
CF DIRECT WEIGHT				
CF SEQUENTIAL WEIGHT				

a. Continuous=Duplexed or RAID Disk, Preferred=Array Disk, Standard=Array or Simplex Disk

b. If a device with Concurrent Copy capability is desired, specify Continuous or Continuous Preferred

6.1.4 SMS Management Class

Table 13 on page 50 shows three examples of Management Classes for DB2 table spaces. The purpose of these three Management Classes is to allow different DFSMSHsm specifications for DB2 table spaces and index spaces. These Management Classes are:

- MCDB20** This Management Class is intended for production table spaces and table spaces that require average or higher availability. This Management Class inhibits DFSMSHsm migration of table spaces.
- MCDB21** This Management Class is intended for table spaces that are allowed to migrate. This Management Class causes migration after one week of inactivity; and after two weeks in level 1, the table space will migrate to level 2. For example, this Management Class can be used for Data Warehouse table spaces.

MCDB22 This Management Class is intended for table spaces that are allowed to migrate and require less availability than that defined in the MCDB2M1 Management Class. This Management Class causes migration after one week of inactivity and the table space will migrate directly to level 2. For example, this Management Class can be used for Test or Development table spaces.

Table 13. SMS Management Classes for DB2 Databases

	MCDB20	MCDB21	MCDB22
Expire after Days Non-usage	NOLIMIT	NOLIMIT	NOLIMIT
Expire after Date/Days	NOLIMIT	NOLIMIT	NOLIMIT
Retention Limit	NOLIMIT	NOLIMIT	NOLIMIT
Primary Days Non-usage		7	7
Level 1 Days Date/Days		14	0
Command or Auto Migrate	None	Both	Both
Backup Options	Not Applicable	Not Applicable	Not Applicable

6.1.5 SMS Storage Groups

SMS Storage Classes and Management Classes are combined to generate Storage Groups. This function is performed by the ACS routines. Table 14 on page 51 shows the relationship between SMS Storage Classes and the SMS Management Classes with the SMS Storage Groups. Only those Storage Groups required to satisfy DB2 database requirements need to be defined.

SMS Storage Groups should be created based on available disk types and data set requirements as defined in Storage Classes and Management Classes. Only exceptionally, for large (partitioned) table spaces and for critical table spaces, may strict data set placement be an issue. Only in these cases may special SMS Storage Groups be defined.

Table 15 on page 51 shows the attributes of the example SMS Storage Groups for DB2 table and index spaces. These are:

SGDB20	Standard Storage Group intended to satisfy most of the DB2 table spaces and index spaces. No DFSMSHsm migration nor DFSMS backup is required.
SGDB21	Storage Group for DB2 table spaces and index spaces that may migrate; for example, data warehouse table and index spaces.
SGDB22	Storage Group for DB2 table spaces and index spaces that may migrate to level 2. For example, development system table and index spaces.
SGDBFAST	Storage Group for DB2 table spaces and index spaces requiring performance. No DFSMSHsm migration nor DFSMS backup.
SGDBCRIT	Storage Group for DB2 table spaces and index spaces requiring performance and availability. No DFSMSHsm migration nor DFSMS backup.

- SGDBTEST Storage Group for DB2 table spaces and index spaces with low performance and availability requirements. This Storage Group allows migration.
- SGDBXXXX Other Storage Groups intended for specific partitioned DB2 table spaces and index spaces, or for other critical table spaces, where strict placement is considered essential. XXXX is any four characters.

The attributes of these Storage Groups are similar to one of the other Storage Groups. These Storage Groups are intended to give DB2 administrators the possibility of placing individual partitions on specific volumes.

Table 14. Relating SMS Storage and Management Classes to Storage Groups

Storage Classes	Management Classes		
	MCDB20	MCDB21	MCDB22
SCDBMED	SGDB20 SGDBXXXX	SGDB21	SGDB22
SCDBFAST	SGDBFAST SGDBXXXX		
SCDBCRT	SGDBCRT SGDBXXXX		
SCDBTEST	SGDBTEST	SGDBTEST	SGDBTEST

The Storage Groups are defined with specific attributes. Table 15 on page 51 shows attributes for the example Storage Groups.

Table 15. SMS Storage Groups for DB2 Databases

Storage Group	Auto-Migrate	Auto-Backup	Auto-Dump	High-Low Thr
SGDB20	No	No	No	70-70
SGDB21	Yes	No	No	70-50
SGDB22	Yes	No	No	60-25
SGDBFAST	No	No	No	70-70
SGDBCRT	No	No	No	60-60
SGDBTEST	Yes	No	No	60-25
SGDBXXXX	No	No	No	99-00

Figure 16 on page 52 is an ISMF display of the Storage Group SGDB20. A Storage Group requires volumes for data set allocation. Figure 17 on page 53 shows the volumes in Storage Group SGDB20. Twelve volumes are assigned to SGDB20. The names of these volumes range from VOL001 to VOL012.

```

Panel  Utilities  Help
-----
                                POOL STORAGE GROUP ALTER
Command ==>>

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : SCDB20
To ALTER Storage Group, Specify:
  Description ==> STANDARD STORAGE GROUP FOR DB2 TABLE AND INDEX SPACES
                ==>
Auto Migrate . . N (Y, N, I or P)  Migrate Sys/Sys Group Name . .
Auto Backup . . N (Y or N)         Backup Sys/Sys Group Name . .
Auto Dump . . . N (Y or N)         Dump Sys/Sys Group Name . . .

Dump Class . . .                    (1 to 8 characters)
Dump Class . . .                    Dump Class . . .
Dump Class . . .                    Dump Class . . .

Allocation/migration Threshold: High . . 70 (1-99)      Low . . 70 (0-99)
Guaranteed Backup Frequency . . . . . NOLIMIT (1 to 9999 or NOLIMIT)

ALTER SMS Storage Group Status . . . . . N (Y or N)

F1=Help   F2=Split   F3=End   F4=Return   F7=Up   F8=Down   F9=Swap
F10=Left  F11=Right  F12=Cursor

```

Figure 16. Display of Storage Group SGDB20

6.1.6 DB2 STOGROUPs and SMS Storage Groups

The concepts of DB2 STOGROUP and SMS Storage Group are different, but very similar. While a Storage Group refers to a set of volumes in an installation, the STOGROUP refers to a set of volumes containing a set of data. Different STOGROUPs can share the same disk volume or volumes.

DB2 administrators normally define many STOGROUPs for their applications. Sometimes they have STOGROUPs for each individual volume and use it to direct the table spaces to that specific volume. Other installations have STOGROUPs at database or application level.

To make the best use of DFSMS, DB2 administrators should define their STOGROUPs as before, but using a generic volume reference (VOLUMES '**'). See the example in Figure 100 on page 172. The generic volume reference allows DFSMS to choose a volume based on the SMS classes assigned to a table or index space.

Sometimes these generic volume references cannot be used for DB2 STOGROUPs. This can happen, for example, during the conversion process from non-SMS to SMS management. If generic volume references cannot be used, SMS Storage Groups can be made to match DB2 STOGROUPs. One important restriction may have to be resolved; this is:

One disk volume can only belong to one SMS Storage Group.

```

Panel  Utilities  Help
-----
STORAGE GROUP VOLUME SELECTION
Command ==>>

CDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name : SCDB20
Storage Group Type : POOL

Select One of the following Options:
  1. Display      - Display SMS Volume Statuses (Pool only)
  2. Define       - Add Volumes to Volume Serial Number List
  3. Alter        - Alter Volume Statuses (Pool only)
  4. Delete       - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
  Prefix  From  To  Suffix  Hex
-----
====> VOL    001  012          - ('X' in HEX field allows
====>                                     FROM - TO range to include
====>                                     hex values A through F.)
====>

F1=Help   F2=Split  F3=End    F4=Return F7=Up     F8=Down   F9=Swap
F10=Left  F11=Right  F12=Cursor

```

Figure 17. Volumes in Storage Group SGDB20

6.1.7 Assigning SMS Classes to DB2 Table Spaces and Index Spaces

SMS classes and Storage Groups are assigned to DB2 table spaces and index spaces through ACS routines. Normally, ACS routines have only the data set name available for their decision making process. Many methods can be devised with specific naming standards to assign SMS classes based on the names of the DB2 data sets. Two types of methods are described, the *filter* method and the *code* method.

The *filter* method must be used when the names are established and cannot be changed. This is the case when an existing DB2 system converts to SMS management. The filter method uses lists of names inside the ACS routine to determine SMS classes.

The *code* method requires naming conventions for DB2 objects that must be strictly enforced. SMS related codes are inserted into the DB2 object names. These codes are used to determine SMS classes. At least two codes are required, one to define the Storage Class, and one to define the Management Class.

The DB2 data set names have a specific structure, shown in Table 3 on page 23. These names have only three components that are dependent on the user and can contain meaningful information for the ACS routine to use. These are:

- High level qualifier
- Database name
- Table space name

The ACS routines can use the filter method, the code method, or a combination of the filter and code methods, and apply these to the three types of names, or to

combinations of these names. This provides the installation with great flexibility in implementation alternatives, such as:

High Level Qualifier Filter

The ACS routines contain a list of high level qualifiers. These qualifiers are used to assign the specific SMS classes. The high level qualifiers can provide a meaningful distinction between data of different DB2 subsystems. This method is recommended as a starting point, because of its simplicity. Some installations may have multiple, complex requirements and may prefer to use another method.

A variant of this method is used in the example shown in Appendix A, section A.4, "DB2 Table Spaces Using SMS, Existing Names" on page 165. In this appendix, Figure 92 on page 168 shows an ACS routine that assigns Management Classes based on a high level qualifier (which is also the DB2 subsystem name). The variant introduced is that certain databases (name starting with *B*) in the DB2D subsystem are assigned a separate Management Class.

Database Name Filter

The ACS routines contain a list of DB2 databases. The database name is used to assign the specific SMS classes. All table spaces and index spaces within a database would have the same SMS classes. When a new database is created, the ACS routine has to be modified.

Table Space Name Filter

The ACS routines contain a list of DB2 databases and table and index spaces. These names are used to assign the specific SMS classes. Each table space and each index space can have distinct SMS classes. When a new table or index space is created, the ACS routine has to be modified. This technique is only manageable in static installations. A simple example of an ACS routine using this method is shown in Figure 18 on page 55.

High Level Qualifier Codes

The high level qualifiers contain a Storage Class code and a Management Class code. These codes are used to assign the specific SMS classes. Multiple high level qualifiers are required to obtain a meaningful distinction between data with different requirements.

Database Name Codes

The DB2 database names contain a Storage Class code and a Management Class code. These codes are used to assign the specific SMS classes. All table spaces and index spaces within a database would have the same SMS classes. The ACS routine does not need maintenance for new databases. This method provides a resolution at database or application level.

```

/*****
/* PARTITION FILTER
/*****

    FILTLIST &PTSP INCLUDE ('LINEITEM','ORDER','PART','PARTSUPP',
                          'SUPPLIER','NATION','REGION')
                          /* Supply a list of the partitioned tablespaces
*/

    FILTLIST &PNDX INCLUDE ('PXL@OK','PXO@OK','PXP@PK','PXPS@SK',
                          'PXS@SK','PXN@NK','PXR@RK')
                          /* Supply a list of the partitioned indexes
*/

    WHEN ( (&DSN(4) = &PTSP OR &DSN(4) = &PNDX)
          AND (&LLQ EQ 'A001' OR &LLQ EQ 'A002') )
        SET &STOGROUP EQ 'SGDB2GRA'

    WHEN ( (&DSN(4) = &PTSP OR &DSN(4) = &PNDX)
          AND (&LLQ EQ 'A003' OR &LLQ EQ 'A004') )
        SET &STOGROUP EQ 'SGDB2GRB'

    WHEN ( (&DSN(4) = &PTSP OR &DSN(4) = &PNDX)
          AND (&LLQ EQ 'A005' OR &LLQ EQ 'A006') )
        SET &STOGROUP EQ 'SGDB2GRC'

/* Repeat the previous WHEN statement for as many STOGROUPs as reqd

```

Figure 18. ACS Routine Extract Using Table and Index Name Filter List

Table Space Name Codes

The DB2 table space and index space names contain a Storage Class code and a Management Class code. These codes are used to assign the specific SMS classes. Each table space and each index space can have distinct SMS classes. The ACS routines do not need maintenance for new table spaces and index spaces. This method is recommended when multiple requirements have to be satisfied. This method provides the most detailed granularity for SMS management and has limited maintenance concerns.

The names of DB2 indexes, including the SMS codes, must not exceed 8 characters. DB2 may change the index space name for indexes having names in excess of 8 characters. The changed names may invalidate this method.

An example of how to structure DB2 data set names to use this method is shown in 6.1.8, “Table Space and Index Space Names for SMS” on page 56.

An implementation example of this method is shown in Appendix A, section A.5, “DB2 Table Spaces Using SMS, Coded Names” on page 174. In this appendix, Figure 105 on page 176 and Figure 106 on page 176 show ACS routines that assign Storage Classes and Management Classes based on codes within the table space name.

6.1.8 Table Space and Index Space Names for SMS

The recommendation in this book for finely tuned SMS installations is to imbed SMS codes into the names of DB2 table and index spaces. This is shown in Table 16 on page 56. The data set names have the structure shown in Table 3 on page 23, with a change in the space name itself. As explained in 6.1.7, “Assigning SMS Classes to DB2 Table Spaces and Index Spaces” on page 53, this name contains codes for the ACS routines. The ACS routines use these codes to establish Storage Classes, Management Classes and Storage Groups.

Table 16. Table Space and Index Space Names with SMS Codes

hlq.DSNDBx.dbname.uvssssss.ynnnn.Ammm

The elements of the space name are:

<i>uvssssss</i>	Space name:
u	Storage Class code
v	Management Class code
ssssss	User assigned name

6.1.9 Managing Partitioned Table Spaces with SMS

With DB2, the user has choices on how to allocate and distribute the individual data sets of a partitioned table space. Two examples are shown in:

- Appendix A, Section A.2, “Partitioned Table Space, DB2 Defined, Without SMS” on page 162
- Appendix A, Section A.3, “Partitioned Table Space, User Defined, Without SMS” on page 164

SMS can also be used to distribute the individual partitions. Several different methods are possible; for example:

- Let SMS manage everything.
- Use one SMS Storage Group for each partition.
- Use one SMS Storage Group for one partitioned table space.

Let SMS Manage Everything

For many partitioned table spaces and index spaces, SMS can handle the partition distribution. If the number of volumes in the Storage Group is much larger than the number of partitions, then SMS will most likely place each partition on a separate volume.

Storage group SGDB20 is an example of this case. Table spaces and index spaces are allocated by SMS on these volumes, trying to balance the I/O activity on the volumes. This method should be adequate for many installations. This method is the preferred technique for storage administrators, because it has the advantage of simplicity.

This method can be used for table spaces and partitions, where each data set is a large fraction of a volume. Because a volume can only handle one partition, the separation is automatic. On the other hand, space fragmentation on the volumes

may not leave enough volumes with adequate free space; this could cause a REORG to fail due to lack of space. The following methods address this issue.

Use One SMS Storage Group for Each Partition

A one-volume SMS Storage Group can be defined for each partition. The ACS routine assigns to each partition its corresponding Storage Group. This method is similar to creating a DB2 defined partitioned table space, using one STOGROUP for each partition. One SMS Storage Group is defined for each DB2 STOGROUP.

The advantage of this method is strict data set placement. The DB2 administrator will have the same disk distribution as he has without SMS. The disadvantage of this method is that many SMS Storage Groups are required, and the ACS routines become more complex and dependent on DB2 table space names. For an example, see Appendix A, section A.7, “Partitioned Table Spaces Using SMS, User Distribution” on page 181.

Use One SMS Storage Group for One Partitioned Table Space

Another alternative is to have one specific SMS Storage Group for each partitioned table space. Enough volumes are assigned to the Storage Group for all the partitions. SMS distributes the partitions on those volumes. Because this Storage Group is dedicated to the table space, no other data sets are ever allocated on these volumes, practically reserving the space for this table space.

If specific volumes of the SMS Storage Group are desired, guaranteed space must be used to assign the partitions to the specific volumes.

For this situation, we do not recommend guaranteed space unless the space requirements are relatively small and static.

To use guaranteed space with DB2 defined data sets, multiple DB2 STOGROUPs are required. Each of these STOGROUP must refer to a volume of the SMS Storage Group. To avoid possible allocation or extension failures, if guaranteed space storage class is used, the storage administrator should run the DFSMSHsm space management function more frequently on the set of volumes assigned to the DB2 STOGROUPs.

If SMS manages the allocation, or if user defined tablespaces are used, only one DB2 STOGROUP is required, defined with (VOLUMES "***"). The example in Figure 100 on page 172 shows a definition of such a STOGROUP. The example described in Appendix A, section A.6, “Partitioned Table Space Using SMS Distribution” on page 178 shows how to allocate a partitioned table space using this method.

6.2 User Databases

This section shows how to assign SMS classes to different types of data. For the purpose of these examples, the data has been divided into the following environments:

- Online Production Databases
- Batch Production Databases
- Data Warehouse Databases
- Development and Test Databases

6.2.1 Online Production Databases

The databases used in production normally contain important data and have special availability and security requirements. Performance may become a critical issue, if the databases are used in an online environment.

Because online production databases contain important data, the DB2 Database Administrators typically monitor them very frequently. These databases are sometimes placed on specific disks to manage the data individually. These databases should not be mixed on a disk with other high activity databases or data sets.

6.2.1.1 Storage Classes

The following example Storage Classes can be used for online production table spaces:

- SCDBMED
- SCDBFAST
- SCDBCRIT

6.2.1.2 Management Classes

The following example Management Class can be used for online production table spaces:

- MCDB20

6.2.2 Batch Production Databases

Most production databases are accessed both in batch and online. In this case, the more restrictive requirement for online databases should be applied to the batch environment. This description applies only to databases exploited in a batch environment.

6.2.2.1 Storage Classes

The following example Storage Classes can be used for batch production table spaces:

- SCDBMED
- SCDBFAST

6.2.2.2 Management Classes

The following example Management Classes can be used for batch production table spaces:

- MCDB20
- MCDB21
- MCDB22

6.2.3 Data Warehouse Databases

Data Warehouse databases contain a special type of production data. Their requirement is normally more oriented to usability and end user access. The end users expects performance to be reasonable (and this is a subjective matter), and the same applies to availability. Some customers run applications in their Data Warehouse environment, and their requirements could become similar to batch or even online production.

6.2.3.1 Storage Classes

The following example Storage Classes can be used for Data Warehouse table spaces:

- SCDBMED
- SCDBTEST
- SCDBFAST

6.2.3.2 Management Classes

The following example Management Classes can be used for Data Warehouse table spaces:

- MCDB20
- MCDB21
- MCDB22

6.2.4 Development and Test Databases

Development and test databases are not essential for immediate business needs. Their performance and availability requirements should not have priority over production databases. Some customers with permanent development and test database environments may have stricter requirements than those shown here. Test environments used for performance and capacity testing may also have stricter requirements.

6.2.4.1 Storage Classes

The following example Storage Classes can be used for development and test table spaces:

- SCDBTEST

6.2.4.2 Management Classes

The following example Management Classes can be used for development and test table spaces:

- MCDB21
- MCDB22

6.2.5 Summary

Table 17 on page 60 shows some examples of how the SMS Storage Classes and Management Classes can be combined to provide for different database requirements. In this table, the concepts of low, average, good, and high, represent service levels agreed upon between the storage administrator and the DB2 administrator. These examples are not meant to be exhaustive, but are intended to provide an idea on how the SMS classes can be used to manage table spaces in DB2 user databases.

Table 17. Examples of SMS Class Usage for DB2 User Databases

Databases	Performance	Availability	Migration	Storage Group
Online Production	Avg	Avg	NO	SGDB20
Online Production	High	Avg	NO	SGDBFAST
Online Production	High	High	NO	SGDBCRT
Batch Production	Low	Low	YES	SGDB21
Batch Production	Good	Avg	NO	SGDB20
Batch Production	High	High	NO	SGDBCRT
Data Warehouse	Low	Avg	YES	SGDB21
Data Warehouse	High	Good	NO	SGDBFAST
Development	Low	Low	YES	SGDB22
Test	Low	Low	YES	SGDBTEST

6.3 DB2 System Databases

A DB2 subsystem stores data about itself within a set of tables stored in table spaces in system databases. The system databases are:

- Catalog database (DSNDB06)
- Directory database (DSNDB01)
- Work database (user defined name or DSNDB07)

DB2 supports a Default database (DSNDB04), which is used when the database is omitted in a table space or index space creation. The Default database can be considered a user database and may be handled in the same way as other user databases. It is not considered in this section.

The system databases have the same data organization, data type and naming convention as user databases. The system databases have stricter availability requirements. The examples of SMS classes in Chapter 6.1, "SMS Examples for DB2 Databases" on page 47 are applicable to DB2 system databases.

6.3.1 Catalog and Directory Databases

The DB2 Catalog and Directory databases contain data definitions, recovery information, security information for the data managed by DB2. If these databases become unavailable, business data is also unavailable. Recovering these table spaces is a lengthy and complex process.

To ensure that the availability requirement of the production databases is met, even in case of an outage of a DB2 system database, the DB2 Catalog and Directory databases must have an availability requirement at least as stringent as those of the production database with the highest availability requirement. Corollary: continuous availability of the DB2 Catalog and Directory is required in order to have continuous availability for a DB2 application.

6.3.1.1 Storage Classes

The following example Storage Classes can be used for online production table spaces:

- SCDBCRIT

6.3.1.2 Management Classes

The following example Management Class can be used for online production table spaces:

- MCDB20

6.3.2 Work Database

All DB2 subsystems use table spaces in a Work database. For example, the Work database stores an intermediate result of a query, or is the workarea for an internal sort of a result table. To avoid contention with DB2 environments requiring high performance, the table spaces in the Work database should not go to the Storage Group where the high performance table spaces are placed (SGFAST and SGCRIT).

The Work database only stores temporary data. If the Work database is lost, DB2 rebuilds its contents automatically on a restart, or manually with a START DATABASE command. This should be adequate for most production environments, no special availability requirements are necessary.

6.3.2.1 Storage Classes

The following example Storage Classes can be used for development and test table spaces:

- SCDBMED

6.3.2.2 Management Classes

The following example Management Class can be used for development and test table spaces:

- MCDB20

6.3.3 Summary

Table 18. DB2 System Database Requirements

Databases	Performance	Availability	Migration	Storage Group
Catalog	Good	Very High	NO	SGDBCRIT
Directory	Good	Very High	NO	SGDBCRIT
Work	Good	Low	NO	SGDB20

Chapter 7. Managing DB2 Recovery Data Sets with SMS

Some DB2 data sets are standard sequential files or partitioned data sets. Many installations already manage these data sets with SMS and have already SMS classes defined for these data sets. Therefore, this chapter only analyzes DB2 recovery related data sets.

This chapter describes attributes for SMS management of the DB2 recovery data sets and provides example SMS constructs for these data sets. DB2 recovery data sets are described in 3.6, “DB2 Recovery Data Sets” on page 17. This chapter includes examples of SMS Data, Storage and Management Classes for the following data sets:

- Bootstrap data sets (BSDS)
- Active log data sets
- Archive log data sets
- Image copy data sets

7.1 SMS Examples for DB2 Recovery Data Sets

The examples shown in this section do not demonstrate all the possibilities that SMS offers; neither can they consider all the different requirements specific to each DB2 installation. Each installation is advised to review these examples and create those classes that best suit its requirements. The examples shown here are extracted and adapted from *DFSMS/MVS Implementing System-Managed Storage*, SC26-3123.

7.1.1 SMS Data Class

DB2 recovery data sets have different attributes. Data Classes can optionally be defined for these data sets.

7.1.2 SMS Storage Class

DB2 administrators may require several Storage Classes for DB2 recovery data sets. These Storage Classes have high availability requirements. Performance requirements may be less severe, with the exception of the active log data sets which have very high performance requirements. Table 19 on page 64 shows four examples of Storage Classes for the DB2 recovery data sets. These are:

SCDBIC	This Storage Class is intended for image copy data sets. It provides good performance and good availability.
SCDBICH	This Storage Class is intended for image copy data sets with a high availability and high performance requirement.
SCDBARCH	This Storage Class is intended for archive log data sets. It provides good performance and high availability.
SCDBACTL	This Storage Class is intended for the BSDS and active log data sets. These data sets are allocated once and rarely redefined. Strict placement is important in order to obtain high availability and high performance. SCDBACTL uses guaranteed space to allocate the data sets on specific volumes within the assigned Storage Group.

Table 19. SMS Storage Classes for Recovery Data Sets

Attribute	SCDBIC	SCDBICH	SCDBARCH	SCDBACTL
Direct response (MSEC)	10	5	10	5
Direct bias				
Sequential response (MSEC)	10	5	10	1
Sequential bias				
Sustained data rate (MB/sec)	10	20	20	40
Availability ^a	Standard	Continuous	Continuous	Continuous
Accessibility ^b	Standard	Standard	Standard	Standard
Guaranteed space	No	No	No	Yes
Guaranteed synchronous write	No	No	No	No
Cache set name				
CF direct weight				
CF sequential weight				

a. Continuous=Duplexed or RAID Disk, Preferred=Array Disk, Standard=Array or Simplex Disk

b. If a device with Concurrent Copy capability is desired, specify Continuous or Continuous Preferred

7.1.3 SMS Management Class

DB2 administrators may require several Management Classes for DB2 recovery data sets. These Management Classes have different expiration and backup requirements. Table 20 on page 65 shows five examples of Management Classes for the DB2 recovery data sets. These are:

- MCDBICD This Management Class is intended for image copy data sets created daily. These data sets will expire after four days.
- MCDBICW This Management Class is intended for image copy data sets created weekly. These data sets will expire after 25 days.
- MCDBICM This Management Class is intended for image copy data sets created monthly and primary copies of the archive logs. These data sets will expire after 365 days.
- MCDBLV2 This Management Class is intended for secondary archive logs and secondary image copy data sets. Using this Management Class, these data sets will be migrated directly to level two.
- MCDBACTL This Management Class is intended for active logs and BSDS data sets. These data sets do not require SMS management.

Table 20. Management Classes for Recovery Data Sets

Attribute	MCDBICD	MCDBICW	MCDBICM	MCDBLV2	MCDBACTL
Expire after days non-usage	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
Expire after date/days	4	25	365	365	NOLIMIT
Retention Limit	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
Primary days non-usage		7	7	0	
Level 1 days date/days		7	7	0	
Command or auto migrate	Both	Both	Both	Both	None
# GDG elements on primary					
Rolled-off GDS action					
Backup frequency	1	7	90	1	
Number of backup versions (data set exists)	1	1	2	1	
Number of backup versions (data set deleted)	2	2	2	1	
Retain days only backup version (data set deleted)	28	28	370	1	
Retain days extra backup versions	28	28	370		
Admin or user command backup	Both	Both	Both	Both	
Auto backup	Yes	Yes	Yes	No	No

7.1.4 SMS Storage Groups

SMS Storage Classes and Management Classes are combined to generate Storage Groups. This function is performed by the ACS routines. Table 21 on page 66 shows the relationship between SMS Storage Classes and the SMS Management Classes with the SMS Storage Groups for DB2 recovery data sets. Only those Storage Groups needed to satisfy DB2 recovery data set requirements are defined.

Table 22 on page 66 shows the attributes of the example Storage Groups for DB2 table and index spaces. The five example SMS Storage Groups are:

- SGDBIC Storage Group intended for standard image copies.
- SGDBICH Storage Group intended for high availability image copies.
- SGDBARCH Storage Group intended for primary and secondary archive logs and for secondary image copies. These data sets will be migrated by DFSMSshm.

- SGDBACTL** Storage Group intended for BSDSs and active logs for all non-production DB2 subsystems. Because the corresponding Storage Class has guaranteed space defined as *yes*, the DB2 administrator can direct the allocation of the data sets on volumes which are dedicated to a specific DB2 subsystem.
- SGDB2PLG** Storage Group intended for BSDSs and active logs for the production DB2P subsystem. The Storage Class contains the volumes for the DB2P subsystem. The DB2 administrator can direct the allocation of the data sets on specific volumes of this Storage Group. Because guaranteed space is used for the SGDBACTL and SGDB2PLG Storage Groups, it is not strictly necessary to create a separate SMS Storage Group for each DB2 subsystem, it simply is one of the many choices available to the DB2 administrator.

Table 21. Relating SMS Storage and Management Classes to Storage Groups

Storage Classes	Management Classes				
	MCDBICD	MCDBICW	MCDBICM	MCDBLV2	MCDBACTL
SCDBIC	SGDBIC	SGDBIC	SGDBIC	SGDBARCH	
SCDBICH	SGDBICH	SGDBICH	SGDBICH	SGDBARCH	
SCDBARCH			SGDBARCH	SGDBARCH	
SCDBACTL					SGDBACTL SGDB2PLG

Table 22. SMS Storage Groups for DB2 Recovery Data Sets

Storage Group	Auto-Migrate	Auto-Backup	Auto-Dump	High-Low Thr
SGDBIC	Yes	No	No	70-50
SGDBICH	Yes	No	No	70-50
SGDBARCH	Yes	No	No	60-40
SGDBACTL	No	No	No	99-0
SGDB2PLG	No	No	No	99-0

7.1.5 Assigning SMS Classes to DB2 Recovery Data Sets

SMS classes and Storage Groups are assigned through ACS routines. The naming standard from 3.8, "DB2 Data Sets Naming Conventions" on page 22 is used for these examples. This naming standard provides ACS routines with the necessary information for deciding the SMS classes.

7.2 BSDS

The bootstrap data set (BSDS) contains the information required by DB2 to start the subsystem in normal circumstances. It also handles the restart and recovery in any abnormal circumstance. For example, all log data sets (active and archive) are automatically registered within the BSDS.

Data Organization

The BSDS is a VSAM KSDS. The data control interval is 4 KB; the index control interval is 1 KB. Figure 19 on page 67 shows an example VSAM definition of a BSDS.

Performance

While DB2 is executing, the BSDS is updated periodically. The frequency of these updates is not high, but is dependent on general DB2 subsystem activity. For example, the BSDS is updated at every DB2 checkpoint and at every archive process.

Availability

The BSDS is a critical resource for DB2. Because of this, DB2 has implemented dual copies for the BSDS. DB2 requires the presence of two copies of the BSDS during restart, to ensure high availability. While DB2 is running, a BSDS may fail and DB2 continues operating with one BSDS. The second BSDS should be restored as soon as possible, to avoid DB2 shutdown, which would occur if the last available BSDS also fails.

```
DEFINE CLUSTER                -
  ( NAME(DB2V610Z.BSDS01)    -
    VOLUMES(SBOX10)          -
    REUSE                     -
    SHAREOPTIONS(2 3) )      -
  DATA                       -
  ( NAME(DB2V610Z.BSDS01.DATA) -
    RECORDS(180 20)          -
    RECORDSIZE(4089 4089)    -
    CONTROLINTERVALSIZE(4096) -
    FREESPACE(0 20)          -
    KEYS(4 0) )              -
  INDEX                       -
  ( NAME(DB2V610Z.BSDS01.INDEX) -
    RECORDS(5 5)             -
    CONTROLINTERVALSIZE(1024) )
```

Figure 19. Example VSAM Definition of one BSDS

7.2.1 Storage Class

BSDSs use a Storage Class with guaranteed space. This allows the DB2 administrator to decide the location of each BSDS.

- SCDBACTL

7.2.2 Management Class

No SMS management is required for the BSDS data sets. The following Management Class has been defined for this purpose.

- MCDBACTL

7.2.3 Storage Group

Because the Storage Class has guaranteed space, the BSDS data sets are allocated on the disk volumes requested by the DB2 administrator. The volumes must belong to the assigned Storage Group (such as: SGDBACTL), and the disk volume must be eligible for SMS. For example, this can be done with the DFSMSdss CONVERT command.

- SGDBACTL for several DB2 subsystems
- SGDB2PLG for the DB2P subsystem

7.2.4 ACS Example

An example of ACS routines to allocate these SMS classes and Storage Groups for BSDSs is shown in Appendix B, section B.1, “BSDS and Active Logs” on page 185.

7.3 Active Logs

The active log data sets are used for data recovery and ensure data integrity in case of software or hardware errors. Active log data sets record all updates to user and system data. If the active log is not available, DB2 cannot guarantee data integrity.

The active log data sets are open as long as DB2 is active. Active log data sets are reused when the total active log space is used up, but only after the active log to be reused has been copied to an archive log.

Data Organization

The active log data sets are VSAM LDSs. Figure 20 on page 69 shows an example definition of an active log data set.

Performance

For DB2 subsystems with high update transaction rates, the active logs have a very high I/O activity (mainly write I/O). The performance of the active logs has an important impact on the overall DB2 subsystem performance. See 10.4.5, “Improving Log Write Performance” on page 114 and 10.5.1, “Improving Log Read Performance” on page 116 for more information.

Availability

The active log data sets have a very high availability requirement for DB2 data integrity. To ensure this, DB2 optionally supports two copies for each active log data set (dual active logs). Dual active logs are highly recommended for DB2 production environments.

To improve active log availability, RAID disks or disks with dual copy can be considered for the active logs.

Migration

Active log data sets should never be migrated by DFSMSHsm.

Backup

Every time an active log data set is filled, DB2 attempts to create an automatic backup. The backup copies of the active log data sets are the archive log data sets.

```

DEFINE CLUSTER -
  ( NAME (DB2V610Z.LOGCOPY1.DS01) -
    VOLUMES (SBOX09) -
    REUSE -
    RECORDS (8640) -
    LINEAR ) -
  DATA -
  ( NAME (DB2V610Z.LOGCOPY1.DS01.DATA) -
    )

```

Figure 20. Example VSAM Definition of One Active Log

7.3.1 Storage Class

A Storage Class with guaranteed space set to *yes*, enables the DB2 administrator to decide the location of the active logs.

- SCDBACTL

7.3.2 Management Class

The following Management Class has been defined for active logs, no SMS management is required.

- MCDBACTL

7.3.3 Storage Group

The same Storage Groups used for the BSDSs with guaranteed space set to *yes*, are also used for the active logs.

- SGDBACTL for several DB2 subsystems
- SGDB2PLG for the DB2P subsystem

7.3.4 ACS Example

An example of ACS routines to allocate these SMS classes and Storage Groups for active logs is shown in Appendix B, section B.1, “BSDS and Active Logs” on page 185.

7.4 Archive Logs

Archive log data sets are DB2 managed backups of the active log data sets. Archive log data sets are required for any recovery that spans a period of time in excess of the time covered by the active logs. This is illustrated in Figure 5 on page 19. Archive log data sets are created automatically by DB2 when an active log fills up, but they may also be created with the `-ARCHIVE` command.

Data Organization

Archive Log data sets are physical sequential data sets. Record size is 4096 and the block size is typically 28672 bytes. The allocation of archive logs is done dynamically by DB2. The DB2 system administrator can influence this process, specifying options in the DB2 parameter module (default name = `DSNZPARM`). Those parameters are defined on installation panel `DSNTIPA`. An example definition is shown in Figure 21 on page 70. On this panel, the DB2 administrator

can define two separate device types for the primary and secondary archive log. This can be seen on line 5 and 6 of Figure 21.

```

DSNTIPA          INSTALL DB2 - ARCHIVE LOG DATA SET PARAMETERS
====>

Enter data below:

  1 ALLOCATION UNITS  ===> CYL           Blk, Trk, or Cyl
  2 PRIMARY QUANTITY ===> 3320         Primary space allocation
  3 SECONDARY QTY.   ===> 0           Secondary space allocation
  4 CATALOG DATA   ===> YES          YES or NO to catalog archive data sets
  5 DEVICE TYPE 1    ===> DASD         Unit name for COPY1 archive logs
  6 DEVICE TYPE 2    ===> DASD         Unit name for COPY2 archive logs
  7 BLOCK SIZE       ===> 28672        Rounded up to 4096 multiple
  8 READ TAPE UNITS ===> 2            Number of allocated read tape units
  9 DEALLOC PERIOD   ===> 0            Time interval to deallocate tape units
 10 RECORDING MAX    ===> 1000         Number of data sets recorded in BSDS
 11 WRITE TO OPER    ===> YES          Issue WTOR before mount for archive
 12 WTOR ROUTE CODE ===> 1,3,4        Routing codes for archive WTORs

 13 RETENTION PERIOD ===> 365         Days to retain archive log data sets
 14 QUIESCE PERIOD   ===> 5           Maximum quiesce interval (1-999)
 15 COMPACT DATA     ===> NO          YES or NO for data compaction
F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 21. Archive Log Installation Panel DSNTIPA

Performance

The archive log performance requirement is dependent on recovery performance, service level, and available active log. Performance requirements for archive logs are normally not very high.

Availability

In general, archive log availability is important to ensure data and system availability. Archive log availability is a function of the amount of available active log. Some installations have enough active log to cover most of their recovery needs. If this is the case, archive log availability becomes less critical.

To enhance availability, DB2 supports software duplication of archive log data sets.

Migration

Archive logs can be created directly to tape, but may also reside on disk. Disk archive logs are eligible to be migrated by DFSMSHsm. The residence time on disk should ensure that the likelihood of a recall is in agreement with recovery service levels. When dual archive logs are defined, DFSMSHsm should migrate them to different tape volumes or devices to ensure availability. One way of achieving this, would be to have the secondary copy to migrate directly to level 2, while the primary copy remains a certain time on level 1. The examples in this chapter show how this can be achieved.

Recovery from disk archive logs is faster than recovery from archive logs on tape. Recovery from active logs is slightly more efficient than recovery from archive logs. Because of these two reasons, generally the disk space dedicated to archive logs may be better utilized for active logs and sending the archive logs

directly to tape.

Backup

The archive logs are a backup of the active logs. DB2 can create dual archive logs. There is no need for an additional backup of the archive logs.

7.4.1 Storage Class

Storage Class SCDBARCH is an example of a Storage Class for archive logs. This Storage Class has high availability and good performance.

- SCDBARCH

7.4.2 Management Class

Two different Management Classes are used for the archive logs. One is used for the primary copy and the other for the secondary copy. Both allow migration of the data sets. The reason for defining two separate Management Classes is to enable a physical separation of the two copies.

The Management Class MCDBICM is used for the image copies retained longest and for the archive logs. This ensures equivalent expiration dates for image copies and archive logs.

The Management Class MCDBLV2 is used for the secondary archives. This will directly migrate the secondary copy to level 2 of DFSMSHsm and so ensure a physical separation of the two archive copies.

- MCDBICM, used for primary archive log data sets
- MCDBLV2, used for secondary archive log data sets

7.4.3 Storage Group

Primary and secondary archive logs are allocated on volumes of the SGDBARCH Storage Group. These data sets are migrated independently on different dates. This is determined by their Management Class.

- SGDBARCH

An alternative to the above Storage Group could be a TMM Storage Group, but only for the secondary copy of the archive logs. A TMM Storage Group simulates a tape device on disk. Multiple data sets are placed together on the same tape. This could have a performance impact if this archive log is required for a recovery or a restart.

7.4.4 ACS Example

An example of ACS routines to allocate these SMS classes and Storage Group for archive logs is shown in Appendix B, section B.2, “Archive Logs” on page 191.

7.5 Image Copies

Image copies are the backup of user and system data in a DB2 subsystem. For a well managed backup and recovery policy, the amount of data in image copy data sets exceeds the amount of production data by at least a factor of three. This

means that a large number of image copy data sets are required and need to be managed.

Data Organization

Image Copy data sets are physical sequential data sets. Record size is 4096 (for any size of page) and the block size is typically 28672 bytes. Sample statements to execute an image copy are shown in Figure 137 on page 198 in Appendix B, section B.3, “Image Copies” on page 194.

Performance

Most image copies have no special performance requirements, but there are cases when the time to take an image copy becomes critical.

Availability

Image copies ensure user and system data integrity. Their availability is critical for DB2 system and application availability. DB2 can optionally generate up to four image copies of a table space or of a data set (for a multiple data set table space). Two of these copies are intended for a disaster recovery at a remote site.

Migration

Image copies can be created on tape, or on disk. Image copies are eligible for migration. Some installations create image copies on a pool of disks and migrate asynchronously later in order to avoid delays due to contention for tape units. If multiple image copies are created, then a technique such as that described for archive logs may be used to ensure device separation for the different copies.

Backup

Image copies are backups of system and user data. Multiple copies can be generated. A previous image copy can act as backup for the most recent one, but then more log needs to be applied during the recovery process. Additional backups improve image copy availability and more frequent image copies reduce recovery time.

7.5.1 Storage Class

This example assumes different availability and performance requirements for image copies. Because of this, two Storage Classes have been defined for image copies.

- SCDBIC
- SCDBICH

7.5.2 Management Class

This examples assumes different retention cycles for image copies. This is reflected in four Management Classes:

- MCDBICD - Daily image copies
- MCDBICW - Weekly image copies
- MCDBICM - Monthly image copies
- MCDBLV2 - Secondary image copies

7.5.3 Storage Group

For this example, three Storage Groups are defined. These provide different levels of performance and availability. SGDBARCH serves to separate secondary copies from the primary copies.

- SGDBIC
- SGDBICH
- SGDBARCH

7.6 Summary

Table 23. Storage Groups for DB2 Recovery Data Sets

Data Set	Performance	Availability	Migration	St Groups
BSDS	Standard	High	NO	SGDBACTL SGDB2PLG
Active Log	Very High	High	NO	SGDBACTL SGDB2PLG
Primary Archive Log	Standard	High	YES	SGDBARCH
Secondary Archive Log	Low	Standard	YES	SGDBARCH
Primary Image Copy	Medium	High	YES	SGDBIC
	High	High	YES	SGDBICH
Secondary Image Copy	Standard	High	YES	SGDBARCH

Chapter 8. Converting DB2 to Systems Managed Storage

This chapter describes the techniques for converting DB2 data to SMS. However, each customer has unique data sets and facilities to support their online environment. These differences have an impact on recommended storage management procedures. Database data has different space, performance, and availability requirements; therefore, dividing database data into categories will help identify the required SMS services and implement a staged conversion to SMS.

8.1 Overview

In order for the DB2/SMS relationship to be successful, the data base administrator (DBA) must clearly specify the characteristics and requirements of the DB2 data sets. The storage administrator must also ensure they are satisfied in the physical implementation.

All types of DB2 data are important for successful operation of a DB2 environment. Great care must be taken in preparing DB2 for its conversion to SMS management.

Under most circumstances, an installation will have already implemented SMS to some degree prior to considering the management of DB2; likely candidates are batch and TSO data sets. Therefore, it is assumed that sufficient skills exist within the storage administrator's area to provide the levels of support needed.

If possible, it is recommended to first convert a DB2 test system to DFSMS, in order to gain experience with the various aspects of the DB2/SMS relationship. The DB2 administrator and storage administrator should work closely together to test the environment. Once satisfied with this scenario, a migration plan should be developed to convert DB2 data.

The technique and implementation sequence for converting a DB2 system to SMS varies according to each installation. However, the following topics provide a guideline:

- Advantages of SMS managing DB2 data
- SMS management goals
- Positioning for implementation
- Conversion processes
- DFSMS FIT
- NaviQuest

8.2 Advantages of SMS Managing DB2 Data

ACS routines can be designed so that SMS restricts the allocation of data sets in DB2 Storage Groups to production databases and selected system data sets. Only authorized users, such as the DB2 administrator or storage administrator can allocate data in these Storage Groups. They also will have the authority to allocate data sets with critical performance and availability requirements to specific volumes. Dual copy provides high availability for selected data sets that

are not duplexed by the database management system. The use of fast write and cache facilities will provide increased performance for databases and recovery data sets.

DFSMS/MVS enhances the backup and recovery utilities provided by the DB2 system as follows:

- DFSMSdss uses concurrent copy capability to create point-of-consistency backups.
- DFSMSShsm backs up system data sets and end-user database data that is less critical than production database data.
- DFSMSShsm carries out direct migration to migration level 2 for archived recovery data sets on disk storage.
- Testing/end user databases can be migrated by DFSMSShsm through the storage hierarchy, based on database usage.

8.3 SMS Management Goals

The aims and goals for managing SMS will differ for each installation, although there are areas where working practices will have a common ground. These areas can be categorized as follows:

- Positioning for future enhancements to both DFSMS/MVS and DB2.
- Improving the storage management of data:
 - Use of SMS to simplify JCL allocation.
 - Maintain support for disk and data storage growth without increasing staff levels.
 - Use of SMS to simplify data movement.
 - Improve disk storage efficiency by increasing space utilization through better use of allocation control.
 - Bring private disk volumes under centralized control.
 - Segregate production from other data.
 - Reduce disk storage requirements by migration of inactive data.
- Improving the DB2 aspects of data management:
 - Spread partitions for a given table/PI.
 - Spread partitions of tables and indexes likely to be joined.
 - Spread pieces of NPIs.
 - Spread DB2 work files, and temporary data sets likely to be accessed in parallel.
- Exploitation of hardware such as RVA.
 - Physical striping of data.
 - Avoiding UCB contention.
 - Use only what disk space is actually needed.

8.4 Positioning for Implementation

For the DBA, there are a number of items to be considered as prerequisites for the process.

8.4.1 Prerequisite Planning

Categorize each data type into separate groups

The usage characteristics and service requirements will have to be considered for each data type, and will include:

- Response time performance
- Accessibility and availability operations
- Initial sizing of data sets and future growth
- Difference between production and user/testing data sets

DB2 STOGROUPs Mapping to SMS Storage Groups

To ensure consistency, it is recommended that DB2 STOGROUPs are converted to equivalent SMS Storage Groups.

Identify DB2 Data Sets Eligible for HSM Management

Decide for which groups of DB2 data DFSMSHsm should have the authority to migrate or backup. For example, production databases, Active logs, System libraries, and BSDS are candidates for *NO MIGRATION* due to their critical status.

Set DSNZPARM to have DFSMSHsm automatically recall DB2 data sets during DB2 access. Set RECALL to Y. Set RECALLD, the maximum wait for DFSMSHsm to complete recreation of data sets on the primary disk, based on testing with typical end user databases.

Use of Guaranteed Space

As part of the initial phase, the GUARANTEED SPACE option can be used to position data, particularly production tablespaces, and active logs. Once satisfied with the allocation of the data sets, it is recommended that this option be removed, so future allocations can be under the sole control of SMS.

Guaranteed space is recommended for use only during the migration period (from DB2 managed to SMS managed) which should be kept short to prevent failures on initial allocation and data set extension. Unlike non-SMS, SMS does not retry allocation on another volume if the requested space cannot be satisfied on the specified candidate volume.

Guaranteed space is not useful unless the space requirements are relatively small and static.

Ensure That All Data Sets Are Cataloged

SMS requires that all data sets are cataloged in ICF catalogs, enabling the use of standard catalog search routines (VSAM and CVOL catalogs are no longer supported after 1999). For further information see, *DFSMS/MVS Managing Catalogs*, SC26-4914.

DB2 Naming Conventions

Certain parts of tablespace names are generated by DB2. This does not leave the DBA with much scope for a flexible naming convention. For further information on this subject see 6.1.7, “Assigning SMS Classes to DB2 Table Spaces and Index Spaces” on page 53 and 6.1.8, “Table Space and Index Space Names for SMS” on page 56. Ensure that the storage administrator is fully aware of any restrictions so ACS routines can be coded accordingly.

DB2 Recovery Requirements

For purposes of DB2 recovery, the degree of integrity required for active logs, imagecopies and archive logs must be decided upon.

Expiration of Data Sets

Management Class expiration attributes should be synchronized with DB2's expiration information:

- Expiration of archive logs must be consistent with the value of ARCRETN. The BSDS should be updated with the DB2 change log inventory utility to remove deleted archive logs.
- Expiration of archive logs must also be consistent with the expiration of image copies. This is described under “Deleting Image Copies and Archive Logs” on page 21.
- Expiration of any DB2 image copies requires running the MODIFY utility to update SYSCOPY.

8.4.2 Service Level Agreement

The service level agreement has to be drawn up between the DBA and the storage administrator, and will include items mentioned in the previous section:

- The levels of service required by different data types.
- Performance, accessibility, and availability characteristics.
- The use of dedicated volumes.
- The use of the GUARANTEED SPACE parameter.
- The use of HSM management (automatic migration, recall, backup, space release, and data set expiration).
- Data set naming conventions.

8.5 Conversion Process

This topic covers those aspects of planning and converting DB2 data.

8.5.1 Sequence

To ensure minimum disruption to services, the following sequence is suggested for implementation:

- Libraries and other DB2 system data sets.
- Archive logs and imagecopies.
- User and testing tablespaces.
- Production tablespaces.

- Active logs and BSDS.

8.5.2 Methodology

8.5.2.1 Conversion Window

Decide when each type of data is available for conversion. During a normal processing cycle, some data sets will be deleted and reallocated, providing the opportunity for SMS management. Online data must be converted when those services are unavailable (down time). This is the most difficult to schedule, and requires precise planning.

8.5.2.2 Data Movement

Each disk device is either SMS managed or not. A data set is considered SMS managed when:

- It has a valid Storage Class.
- It resides on a volume in an SMS Storage Group, or has been migrated by DFSMSHsm.

Data sets can be:

- Converted with movement
- Converted in place

Converted with Movement

This is achieved by using a space management function such as DFSMSdss COPY, DFSMSdss DUMP/RESTORE or DFSMSHsm. This method is applicable if the data is application owned. However, consideration must be given to the number of spare disk devices required, while this method is in progress. Also, consider using this approach if the disk devices being used are positioned with storage controls of varying performance (caching). An advantage with this method is allowing data to be allocated using volume thresholds set for each Storage Group, thus allowing space management to operate.

For tablespaces, the DB2 utility **REORG** can be used to automatically convert with data movement if the tablespace is DB2 defined. If it user defined, then a IDCAMS DELETE/DEFINE CLUSTER must be executed between the REORG phases.

Converted in Place

This is achieved by using DFSMSdss CONVERTV function. This approach requires exclusive use of the data sets residing on the disk device. If data sets are already positioned in pools of volumes, this may be an appropriate method to use (tablespaces are likely to be grouped this way). Be warned, if the volume and data sets do not meet all SMS requirements, DFSMSdss will set the volume's physical status to initial. This status will allow data sets be accessed, but not extended. New allocations on the volume are prevented. If all requirements are met, DFSMSdss will set the volume status to CONVERTED.

8.5.2.3 Tailor Online Conversion

Many installations have data sets that are open and active most of the time. Staging a conversion into smaller manageable portions of data provides safer implementation results.

8.5.2.4 Contingency Time Frame

Limit the amount of data converted at a particular time, so if problems are experienced, the situation can be recovered or backed out.

8.5.3 SMS Implementation

The storage administrator performs the implementation of SMS, using ISMF to update the ACS routines. However, it is normally the DB2 administrator who is closely involved with the planning and positioning of data. An outline of activities required are listed below:

- **Definition of Data Classes**

This is optional, although it is usually recommended that Data Classes be assigned. Even though it is not saved for non SMS managed data sets, the allocation attributes in the Data Class are used to allocate the data set.

- **Definition of Storage Classes**

Data sets must have a Storage Class to qualify for SMS management. Here GUARANTEED SPACE is specified, along with availability, performance, and accessibility characteristics.

- **Definition of Management Classes**

This is used for migration to level 1 and level 2 with or without backup, and indicates if there should be no HSM management (backup or migration). It also includes expiration of data sets and space release/compaction.

- **Definition of Storage Groups**

The Storage Group contains volumes that satisfy the service requirements of the data sets allocated to them. They can handle more than one type of data. Separate Storage Groups should be defined for production tablespaces, active logs, other production data, and non-production data.

- **Policy Documentation**

The storage administrator defines policies that include:

- Data set naming conventions
- Volume naming conventions
- Restrictions on use of volumes
- The mapping of DB2 STOGROUPS with SMS Storage Groups
- Use of Data Classes
- Use of GUARANTEED SPACE parameter
- Use of Storage Classes
- Use of Management Classes

- **ACS Routines**

The storage administrator uses the agreed policies to implement DB2 data under the control of SMS. This should be a documented procedure that includes:

- Taking copies of the SMS control data set (ACDS, SCDS), and the source of the ACS routines prior to updating, for back out purposes
- Adding the relevant code for the DB2 data to the ACS routines

- Translating and validating the ACS routines
- Generating test cases, to ensure updates to the ACS routines have the desired effect
- Activating the new SMS configuration

8.5.4 Post Implementation

Once DB2 data sets are SMS managed, there must be an ongoing procedure for maintaining the environment:

- Monitoring performance, availability, and accessibility
- Ensuring that DB2 data receives the correct level of service

The use of monitoring tools such as ISMF, Clists, and DFSMSopt can be used to help achieve these goals.

8.6 DFSMS FIT

DFSMS fast implementation techniques (FIT) is a process that supports the customer in implementing SMS. The process was developed after working with a number of DFSMS implementations, and provides a simple proven design that leads to a successful SMS implementation within two or three weeks.

Most installations implement SMS on a phased basis. First, candidates such as batch and TSO data may be targeted. Once some operational experience has been gained, then other categories such as databases can be included.

With DFSMS FIT, a complete design can be developed, and then the implementation can be phased in manageable portions. It uses a question and answer approach for steps of the design process. The documentation includes many samples of implementation, including jobs, coding, and procedures.

The process assumes IBM NaviQuest for MVS will be used for testing.

For more information on the implementation techniques, see the following publications:

- *Get DFSMS FIT: Fast Implementation Techniques*, SG24-2568
- *DFSMS FIT: Fast Implementation Techniques Process Guide*, SG24-4478
- *DFSMS FIT: Fast Implementation Techniques Installation Examples*, SG24-2569

8.7 NaviQuest

IBM NaviQuest for MVS can be used in conjunction with DFSMS FIT. It is a testing and reporting tool for the DFSMS environment, and is designed specifically to work with DFSMS FIT. It provides the following facilities:

- Automatically test the DFSMS configuration.
- Automatically create test cases.
- Automatically test the ACS routines.

- Perform storage reporting, through ISMF and with DCOLLECT and Volume Mount Analyzer (VMA) data.
- Print ISMF lists.
- Run ISMF functions in batch mode, using the REXX EXECs provided.

For more information on this feature, see *DFSMS/MVS V1R3 NaviQuest User's Guide*, SC26-7194.

Part 3. DB2 and Storage Servers

Chapter 9. Disk Environment Overview

This chapter considers the disk architecture from a DB2 point of view. It focuses on concepts and recommendations for their practical implementation, rather than on technical details. In order to facilitate the mutual understanding of some storage terms between DB2 administrators and storage administrators, we highlight them in italics. Several considerations in this chapter could also apply to the new tape server environments, such as the IBM Seascope Virtual Tape Server.

9.1 Evolution of Disk Architecture

We can identify four steps in the evolution of the disk architecture that have progressively separated the concept of volume from the concept of physical device.

9.1.1 3380 and 3390 Volumes

The 3380 and 3390 have been available on the market as physical devices characterized by a one-to-one relationship between a disk drive and a volume. The physical characteristics of these devices also represent a logical view that consists of:

- Track size (*track image*), the number of bytes per track: 47476 and 56664 bytes of data for 3380 and 3390, respectively
- Capacity in terms of number of tracks or gigabytes
- Device address (*device number*), which is a thread onto which I/O operations are serialized by the operating system

Although the physical devices 3380 and 3390 will eventually no longer be used, the logical view—with the three characteristics of track size, capacity, and addressing—continues to exist in the new concept of *logical volume* or *logical device*.

9.1.2 Arrays

An *array* is the combination of two or more physical disk storage devices in a single logical device or multiple logical devices. *Redundant array of independent disks (RAID)* distributes data redundantly across an array of disks. The objective is to achieve continuous data availability in the face of various hard drive failures through the use of disk mirroring, parity data generation and recording, hot sparing, and dynamic reconstruction of data from a failed disk to a spare disk. RAID technology provides the disk I/O system with high availability. RAID types have been categorized into five levels: RAID 1 through 5. Some new definitions have been developed to address new implementations or updated views of the RAID concept.

Each RAID level has some basic characteristics, but all of them have a *fixed mapping* between logical devices (or logical volumes) and physical drives. Currently admitted definitions (see *IBM RAMAC Array Subsystem Introduction*, GC26-7004) are:

- RAID 0: data striping without parity

- RAID 1: mirroring
- RAID 2: synchronized access with separate error correction disks
- RAID 3: synchronized access with fixed parity disk
- RAID 4: independent access with fixed parity disk
- RAID 5: independent access with rotating parity
- RAID 6: dual redundancy with rotating parity

Note that we still closely associate the terms *volume* and *device* because the mapping is fixed. A logical device now consists of those storage facility resources required to manage the data that is accessible to an ESCON device. This definition can also be extended to a SCSI logical unit in a disk data-sharing environment. The definition of the mapping between logical volumes and physical arrays of disks can be done by configuration tools at the level of the storage server by implementing the fixed mapping tables. Figure 22 on page 86 shows an example of RAID mapping: eight logical volumes onto four physical *head disk assemblies (HDAs)* in a RAMAC 3 drawer. Note that, while the logical volume view is still *Extended Count Key Data (ECKD)* architecture, the physical HDAs have a *fixed block architecture (FBA)*. This flexible association disconnects the technology from the architectural implementation.



Figure 22. RAMAC3 Drawer Logical Volume Mapping

9.1.3 Log Structured File and SnapShot

The physical disk space is considered as a never-ending sequential space, called a *log*. New or updated data is placed at the end of the log, in the free area. Therefore, data is never updated in place; it is always written to a new place. Only the most recent copy of the data is valid, and a directory indicates the position of this copy. The track image is the update unit.

Every functional volume is defined as the *most recent* set of tracks. This concept offers two advantages:

- *Timed evolutionary view of the data*

- Only one set of write operations to disk in continuous physical sequence (instead of a set of random writes), which is *the most optimized* write mode for RAID technology

Figure 23 on page 87 and Figure 24 on page 87 illustrate the LSF concept.

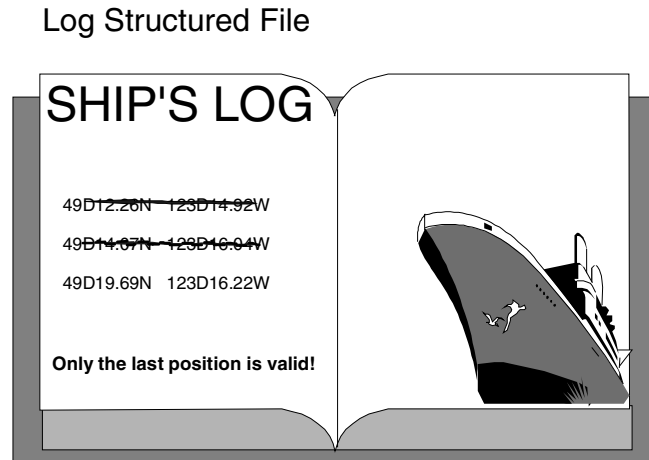


Figure 23. LSF Concept 1

The main challenge of an LSF architecture is managing the free space. Because the LSF log has to be *never-ending*, an LSF system must always have enough free space for writing new data. Over time, old copies of data begin to fragment the log, so to reclaim free space, some type of automatic cleaning routine must be hardware implemented to defragment the log. This cleaning is often referred to as *garbage* or *free space collection*. The benefits of LSF outweigh the overhead of free space collection.

Log Structured File...

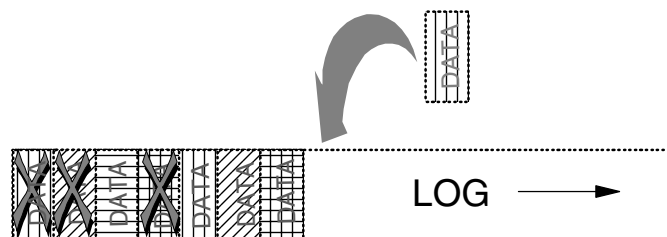


Figure 24. LSF Concept 2

The timed view of a volume, through an instantaneous duplication of the table representation of a volume, allows two independent views of the same physical data *without any data move process*. So each view can do independent updates that are separately recorded, while the common unchanged part is still shared. Figure 25 on page 88 shows an overview of snapping a volume with SnapShot.

Volume SnapShot

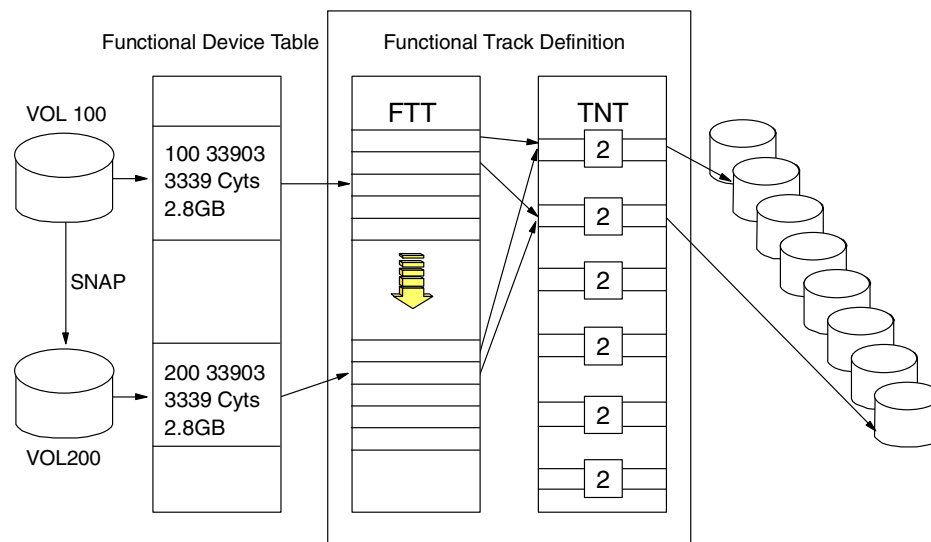


Figure 25. Snapshot Overview

SnapShot, as it "copies" from a *source* object to a *target* object (in compliance with MVS definitions):

- Defines instantaneously the target object
- Allows instantaneous access to both source and target objects (no physical copy lead time to wait for)
- Shares the physical data on disks at copy time (no double space occupancy to manage).

SnapShot is a virtual data duplicator, at volume and data set level, that exploits the architecture of the RVA to create copies of data almost instantaneously. SnapShot produces copies without data movement because it logically manipulates pointers within the RVA. Because there is no actual movement of data, snapping can take seconds rather than minutes or hours, and host processor and channels are not involved because there is no data transfer. As far as the operating system is concerned, the snap is a real copy of the data; as far as the RVA hardware is concerned, the snap is a virtual copy of data.

For more information about LSF and SnapShot concepts, refer to *IBM RAMAC Virtual Array*, SG24-4835. For implementation of SnapShot facilities, we recommend using it implicitly through the DFSMSdss interface, as described in *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268. This approach requires the minimum changes in JCL. For specific examples in business intelligence applications, see *Using RVA and SnapShot for Business Intelligence Applications with OS/390 and DB2*, SG24-5333.

9.1.4 Virtual Volumes

A higher level of flexibility in organization is accomplished when there is no fixed physical-to-logical mapping. The control unit dynamically maps *functional*

volumes to physical disks. A functional volume is a logical volume still defined by track size, capacity, and address. This mapping structure is contained in a series of *tables* stored in the control unit. These tables are updated at each write on functional volume, and have to be maintained when previously used space is released.

Data from all functional volumes could reside on one array device or many array devices. Functional volumes can be defined and configured non-disruptively through dynamic activation and utilities, such as *IBM Extended Facilities Product (IXFP)* for the RAMAC Virtual Array (RVA). Because more arrays can be installed and defined non-disruptively, increasing the capacity of such a control unit is easy.

Defining a logical volume by tables brings capabilities such as easy and almost instantaneous volume duplication when both source and target volumes are controlled by the same set of tables inside the same storage server. However the actual data copy has yet to take place. Either background storage server tasks or system utilities implement the movement of data. Virtual volume definition by tables brings another function: it allows the instant volume duplication by creating two independent host access views of the same data, simply sharing the same data with no replication. The IBM RVA SnapShot enables instantaneous duplication with no physical space utilization at duplication time. This advantage comes from the other architecture improvement, the LSF concept, on which the virtual volume architecture is based.

9.2 Disk Control Units

DB2 uses VSAM Media Manager for its I/O operations. Like any access method, VSAM Media Manager builds for every I/O a channel program and sends a request to the I/O supervisor. The I/O supervisor enqueues this request on a device number for the channel subsystem.

The channel program consists of standard commands, described in the ECKD disk architecture, that specify I/O demand to the control unit. The control unit executes these commands, propagates them and controls their requests to logical volumes and physical devices. It also manages data delivery to the channel subsystem.

The channel subsystem manages the transfer of channel commands and of data through links to the control unit. This linking can be complex and involves ESCON Directors, channel extenders, and even telecommunication devices for remote I/Os.

There are two views of a disk control unit. Physically it is a *storage server* to which disk drives are attached and the channel links from hosts are connected. The storage server contains all the facilities to perform the I/O operations. Logically the disk control unit is an aggregate of subunits known as *logical control units* (LCUs) or *control unit images*, doing the I/O operations.

9.2.1 Storage Server

The *storage server* contains all shared resources and processes to support LCU activities. It often consists of two or more *clusters*. A cluster can take over the

processing of any other failing cluster. Let us briefly review the storage server subcomponent relationships:

- *Host adapters* attach channel links and allow them to communicate with either cluster-processor complex. Practically, statistics at this level deal with what is called *upper interface* busy percentage.
- *Device adapters* provide storage device interfaces. Statistics captured at this level, very often indirectly measured, are called *lower interface* busy percentage.
- The *cluster-processor complex* provides the management functions for the storage facility. It consists of cluster processors, cluster memory, cache, nonvolatile storage (NVS), and related logic.

9.2.2 Storage Devices

Storage devices provide the primary nonvolatile storage medium for any host data stored within the storage facility. Storage devices are grouped in *arrays* (or *ranks*) and are managed by the storage server as a common resource.

9.2.3 Logical Control Unit

Each LCU has an associated set of devices. Each device has a unique device address on the LCU. All LCUs are accessible over any installed host adapter. Host traffic and performance are controlled at the LCU level.

For OS/390 architecture, an LCU views up to 256 *logical volumes* (or *device numbers*); it is physically identified by a *subsystem identifier (SSID)* at installation time, but dynamically referred to by a *LCU number* determined at initialization time. As an example of implementation, an IBM RVA Turbo 2 storage server is viewed as four LCUs. Each LCU currently contains 64 functional volumes, to be increased to 256 when the 1024 addresses support is delivered.

9.3 Cache Management

Cache is a storage server memory resource used to buffer data for reuse and a faster access by channel. Cache masks many of the mechanical actions from the I/O access and improves the service time when the data is accessed from cache rather than from the disk. A cache hit (when the required record is found in the cache) comprises the data transfer time plus a small protocol time for both reads and writes. Read misses and write misses have the same response time characteristics as if they were uncached.

Cache performance depends on:

- Locality of reference, the likelihood of references to other records in the same track
- Frequency of reuse, the likelihood of referencing again (re-referencing) the same record or track

Locality of reference and re-referencing are results of the access pattern to the data, which in turn is related to the application. Fast I/O response times usually rely on a high cache hit rate, minimizing the number of accesses to disk.

In a relational database environment, the physical separation of logically related data results in little locality of reference. Data in memory techniques also minimize the re-referencing of data on disk, as this is ideally accomplished in processor memory.

Write caching requires that data integrity be preserved. Applications assume that an update written to disk is safe. When cache memory is used to improve the performance of writes, an additional level of protection is provided by either the NVS, which has battery protection, or by battery protection of the cache itself. In the first case, updates are written to both cache and NVS before I/O is signaled complete to the application. The copy in the NVS is marked as available for overwriting once the update is destaged to disk. The function of caching writes is called *DASD fast write (DFW)*.

To maximize the efficiency of the cache, storage servers have a variety of caching algorithms to use the cache for data with good caching characteristics, but prevent poor cache candidates from swamping the cache. These caching algorithms are either invoked directly from the software or determined by the server itself.

Cache is managed on a *least recently used (LRU)* algorithm, where the oldest data is made available to be overwritten by new data. Large cache improves the residence time for a cache-unfriendly application. Caching is controlled by the hardware at the volume level or extent level. It is controlled at the subsystem level (through the IDCAMS SETCACHE command), and at the volume or the data set level through software.

9.3.1 Track Caching

Track caching assumes that once a record is accessed on a track, another will be accessed soon. This is the unique algorithm used by RVA.

When a track is accessed on disk, either the required record is passed back to the application and simultaneously copied into the cache and the remainder of the track is staged from the disk, or, for RVA, the whole compressed and compacted track is staged in the cache.

Good performance for track caching depends on good locality of reference. Random workloads often result in poor cache hits, that is, data is staged into cache but never re-referenced. Unproductive staging results in:

- Keeping the disk busy while the track is staged into cache
- Keeping the paths busy while staging
- Using up space in the cache

A poor cache hit rate is likely to be less than 50-60% for reads. To gain the benefits of DFW, data with a poor cache hit rate will require a large cache.

9.3.2 Read Record Caching

Read record caching is suitable for data that has a poor cache hit rate and is therefore subject to unproductive staging. Where read record caching algorithms are invoked, the required record is returned to the application and copied into the cache, but the remainder of the track is not. Record caching avoids adversely impacting the performance of good cache candidates.

9.3.3 Write Record Caching (Quickwrite)

Write record caching, called *quickwrite*, extends the benefits of DFW to data that does not have a read-before-update access pattern (which is currently required to have a DFW hit) and for data with a poor cache hit rate. Data with a predictable record format, such as VSAM records, can benefit from this algorithm.

9.3.4 Sequential Caching

Sequential access to data is managed by a sequential caching algorithm that prestages tracks ahead of the record requested by the application. Once accessed, the space occupied by those tracks is marked as available for reuse rather than being subjected to the LRU algorithm (the exception is the 3990 controller). DB2, through the Media Manager, by requesting sequential caching, optimizes the cache management for better performance.

9.3.5 No Caching—Bypass Cache

Applications that do not benefit from caching can specify the bypass cache option in the Define Extent command. This setting is accomplished by the access methods. Most of the storage servers implement bypass cache by caching the tracks of data anyway, but also managing LRU algorithms for a faster reutilization of the cache space memory those tracks occupy.

9.3.6 No Caching—Inhibit Cache Load

A variant of bypass cache is the inhibit cache load (ICL) command. This command specifies that if the data is found in the cache, it can be read from the cache, but if not, it should not to be staged into the cache. This may be of benefit when the same data is accessed in several different modes, for example, read by a sequential prefetch operation as well as a random read operation.

9.3.7 DB2 Cache Parameters (DSNTIPE)

DB2 interfaces I/O through the VSAM Media Manager and uses the ICL command to optimize the sequential processes. The setting is done at DB2 installation time in the DSNTIPE panel. For the best large cache utilization, we recommend the SEQUENTIAL CACHE parameter being specified to SEQ (instead of default BYPASS) for DB2 prefetch. The UTILITY CACHE OPTION parameter should be set to YES (instead of default NO).

9.3.8 Dynamic Cache Management Enhancement

Dynamic Cache Management Enhanced (DCME) is an interactive cache resource management algorithm between System Managed Storage (SMS) and storage server Licenced Internal Code (LIC). SMS specifies for each data set the level of performance required:

- Whether the data set should be cached (must cache)
- Whether the data set should be excluded from caching (never cache)
- Whether caching should be used only if it is suitable to do so (may cache)

The recommendation is to define all data sets requiring performance as *must cache* (which is accomplished when an ACS routine sets them in a Storage Class defined with a low response time). This means that while all data sets use caching, the *must cache* data sets have an optimized LRU algorithm, which

allows them a longer stay in cache. Other data sets should be set in *may cache* Storage Classes, defined with intermediate response time values.

9.4 Paths and Bandwidth Evolution

A path is a logical concept that lies on the physical web of links (cables) existing between hosts and storage servers. This topology can be highly diversified and complex. An important factor of this interface layer is the potential number of parallel activities a specific DB2 subsystem on a specific host can sustain with a given storage server: this is the *number of paths*.

For any host-LCU association, paths are defined in the IOCP:

- CHPID PATH defines the set of host physical links or channels to be used.
- CNTLUNIT CUNUMBR...PATH enumerates, for each LCU the usable paths from previous set (please note that the name is not related to the SSID).

Similarly, CESTPATH establishes paths between a pair of LCUs for *peer-to-peer remote copy (PPRC)* by defining the physical links to be used.

The bandwidth between two nodes in the network, such as the host and the LCU, is the maximum number of MB/sec that can instantaneously flow from one node to another. The actual bandwidth represents the possible sustained transfer activity. At building time, Storage Groups should be designed with pathing considerations to control the potential parallel access they offer to allocations, specifically when several control units are merged in a large disk storage server.

Sequential data striping exploits this parallel topology. Sequential data striping is provided as a capability of DFSMS/MVS and it is available only for DFSMS-managed data sets.

The *OS/390 Fiber Connection (FICON) Architecture* improves from 17 half-duplex to 100 full-duplex MB/sec bandwidth on a physical link. While the number of CHPIDs per channel is still 256 per processor, likewise the number of device addresses per LCU also remains at 256, there is a 16-fold increase (from 16 to 256) in the number of LCUs a physical link can address (and thus in device addresses, or device numbers). A statement of direction exists for S/390 FICON control units. Disk storage servers with native FICON attachments are not yet on the market, but FICON links can already attach to ESCON Directors and split into up to eight 17 MB/sec ESCON physical links to reach unmodified current disk storage servers. So path parallel capabilities should dramatically improve in the next several years.

9.5 Capabilities

Most disk storage server capabilities deal with availability, performance, or sparing space resource. This section reviews the most popular capabilities from a DB2 utilization point of view.

9.5.1 Dual Copy

Dual copy of a volume (called *primary volume*) is an availability option that triggers duplication of any write onto a shadow volume (*secondary volume*) of the *same physical control unit*. It is also referred as RAID 1 because there is a

one-to-one redundancy. Its purpose is I/O automatic switching to secondary when unattended outage occurs on primary.

Virtual volume, RAID 5, and RAID 6 have made the concept of dual copy practically obsolete.

9.5.2 Concurrent Copy

Concurrent copy is a function the disk storage server controls in conjunction with DFSMS software. Concurrent copy enables taking backup copies of data while minimally impacting the application data access. Concurrent copy delivers a consistent point-in-time copy of a table space while allowing minimum interruption to data access. Table spaces remain available for processing during almost the entire backup process. They must be switched into read mode for only a short time during concurrent copy initialization.

Initiating a concurrent copy operation allows DFSMSdss to build a map of the data to be copied. To create this map in the case of update I/Os, DFSMSdss needs to obtain serialization on the data while the application I/O is suspended. This process establishes the concurrent copy session (or logical copy); then application access to the data can be resumed, and the copy is available for DFSMSdss DUMP or COPY functions.

With concurrent copy, consistent copies of DB2 data objects can be taken almost instantaneously, thus significantly reducing the length of system outage required for backup. The elapsed time for a backup is reduced from the time taken to physically back up the data to a minimum time to establish the concurrent copy session. Either the database must be quiesced, or the table locked in read-only access for the data to be serialized. More backups can be captured and therefore reduce the length of time taken for forward recovery in the event of a recovery. Figure 26 on page 94 shows the difference between a traditional backup and a concurrent copy backup.

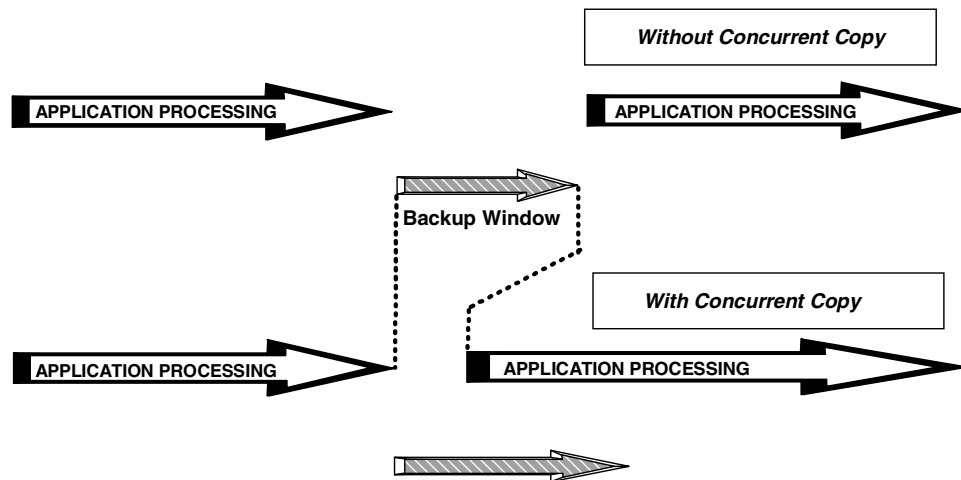


Figure 26. Schema of a Backup with Concurrent Copy

DB2 fully integrates concurrent copy into DB2 recovery. The CONCURRENT option on the DB2 COPY utility reduces disruption and automatically manages the copies used for recovery, to ensure consistent data. This option invokes the concurrent copy function of DFSMSdss, and records the resulting image copies in the SYSCOPY table of the DB2 catalog. “Image Copy Options” on page 20 has more information about DB2 use of concurrent copy.

Concurrent copy is called through the *DFSMSdss standard API*. DB2 COPY with the CONCURRENT keyword calls this API for full image copies. DB2 RECOVER recognizes that type of image copy. Other callers of concurrent copy are IMS, CICS (backup while open), and DFSMSshm.

9.5.3 Virtual Concurrent Copy

Virtual concurrent copy extends the benefits of concurrent copy to users who have RVA installed with SnapShot. When the CONCURRENT keyword is specified on a DFSMSdss COPY or DUMP statement, the software can detect whether you have a 3990 storage control or an RVA. If you have an RVA, the virtual concurrent copy function is invoked. If all the criteria are met for DFSMSdss SnapShot, a DFSMSdss SnapShot will be performed in preference to a concurrent copy.

The logical completion of the point-in-time copy occurs when the source data is snapped into an interim data set called the working space data set (WSDS). The physical completion occurs when the data is moved by DFSMSdss to the target tape or disk data set. Once copy is logically complete, the data can be made available for application updates. Figure 27 on page 95 explains the four steps of a virtual concurrent copy operation.

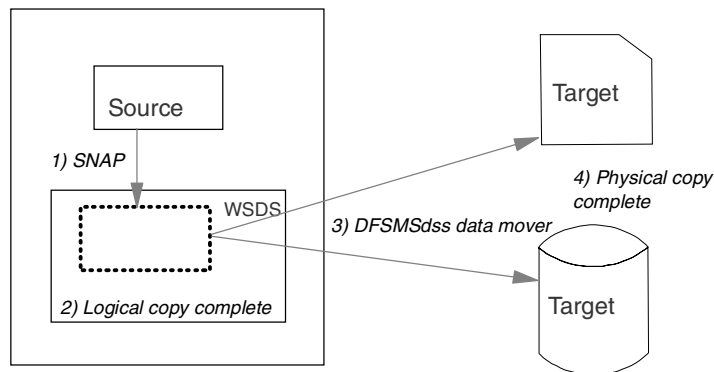


Figure 27. Virtual Concurrent Copy Operation Steps

When concurrent copy is already in use, it is not necessary to change the JCL to use virtual concurrent copy. As concurrent copy support is incorporated into the backup and recovery utilities of DB2, IMS, and CICS, virtual concurrent copy can take advantage of this support immediately and without any change.

9.5.4 Remote Copy

Remote copy continuously duplicates on a remote (secondary) storage server any update done on a local (primary) storage server. The objective is to provide an application independent disaster recovery solution. The problem with

traditional disaster recovery, is that each software subsystem (CICS, IMS, DB2, VSAM, and others) has its own recovery technique. Because an application is typically made up of multiple software subsystems, it is impossible to get a time-consistent backup across all subsystems unless the application is stopped, which impacts availability. Please note that backups are still required in a remote copy environment.

Duplication can be done on the secondary server either *synchronously* or *asynchronously* with the primary server update. The IBM 3990 open extended architecture defines peer-to-peer remote copy (*PPRC*) for synchronous environments and *extended remote copy (XRC)* for asynchronous environments. To provide an operational disaster recovery solution, data consistency is mandatory for secondary remote copy volumes should any event occur to the primary, the secondary, or to links between primary and secondary. Continuous availability of the primary site is also mandatory when secondary site outage occurs. For consistency reasons, we recommend choosing only one remote copy technique, synchronous or asynchronous, for a given environment.

9.5.4.1 PPRC

PPRC allows two disk storage servers to directly communicate with each other through ESCON links. The storage servers can be sited up to 43 km apart. The remote copies are established between two disk volumes. Once the pairs are *synchronized*, the storage servers maintain the copies by applying all updates to both volumes. Updates must be received at both storage servers before the I/O is posted as complete to the application making PPRC operation *synchronous*. Figure 28 on page 96 shows the PPRC data flow (where SP stands for Storage Path).

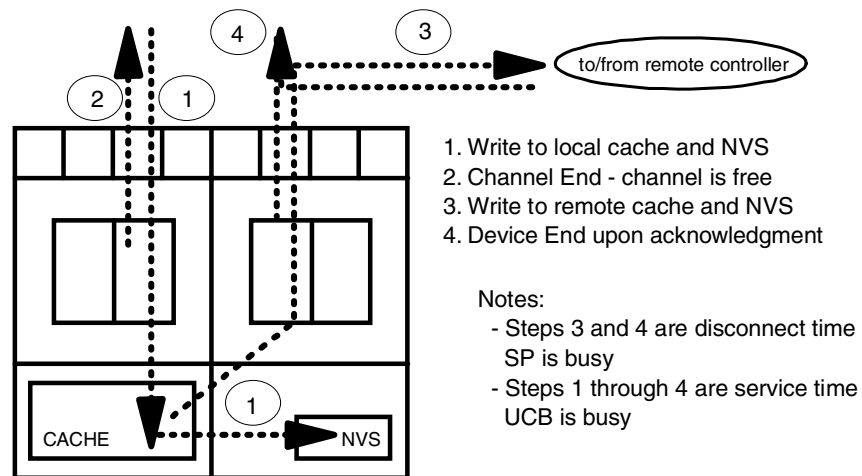


Figure 28. Profile of a PPRC Write

PPRC operations are entirely at the disk volume level. *Write sequence consistency* is preserved by the updates being propagated to the second site in real time. Databases that are spread across multiple volumes may be unrecoverable if a *rolling disaster* causes the secondary volumes to be at an inconsistent level of updates. A rolling disaster is one where various components fail in sequence. For example, if a data volume failed to update its secondary, yet the corresponding log update was copied to the secondary, this would result in a secondary copy of the data that is inconsistent with the primary copy. The

database would be corrupted and would have to be recovered from image copies and log data. In all cases notification of this miss must be known at secondary. When that happens for hundreds of volumes, without a clear notification of status of impacted secondary volumes, recovery can be extremely long. For more information on this topic, please refer to *RAMAC Virtual Array: Implementing Peer-to-Peer Remote Copy*, SG24-5338.

Figure 29 on page 97 shows time-sequenced I/O writes in a synchronous remote copy environment.

The Need for Time-Consistency

- ★ Many examples where the start of one write is time dependent on the completion of a previous write
 - Data base & log
 - Catalogs, VTOCs
 - Index & data components
- ★ Time sequence could be exposed in remote copy
- ★ To be managed through
 - PPRC Critical attribute
 - Automation / Freeze function

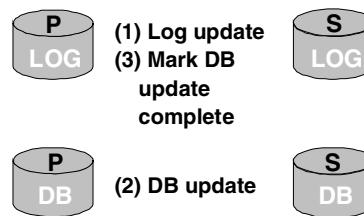


Figure 29. Time Sequenced I/Os

9.5.4.2 Geographically Dispersed Parallel Sysplex

Currently some System/390 platform users have set up a sysplex over multiple sites for availability, capacity, and/or workload balancing reasons. However, these configurations provide reduced continuous application availability because, if a disaster occurs at the site where the data resides, the surviving portion of the sysplex will be down until lengthy data recovery actions can be completed. Moreover, data recovery can be expected to be incomplete and may lag actual production status by up to 24 hours.

A *geographically dispersed parallel sysplex (GDPS)* is a multisite availability solution that merges sysplex and remote copy technologies. GDPS provides an integrated disaster survival capability that addresses the system, the network, and the data parts of an application environment.

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

In the event of a site failure (including disasters), the surviving site will continue to function and absorb the work of the failed site. In the event of a planned site outage, the workload executing in the site undergoing a planned outage will be quiesced and restarted at the other site. Current experience indicates that for large operational sites a *planned switch* from one site to the other takes less than 60 minutes (including networking), and *site unplanned outage recovery* takes less than 45 minutes. Only a single keystroke is required to invoke a GDPS action.

This replaces a manual site switch process that could require more than 20 people to be present to perform their specialized tasks. Figure 30 on page 98 shows the global GDPS architecture.

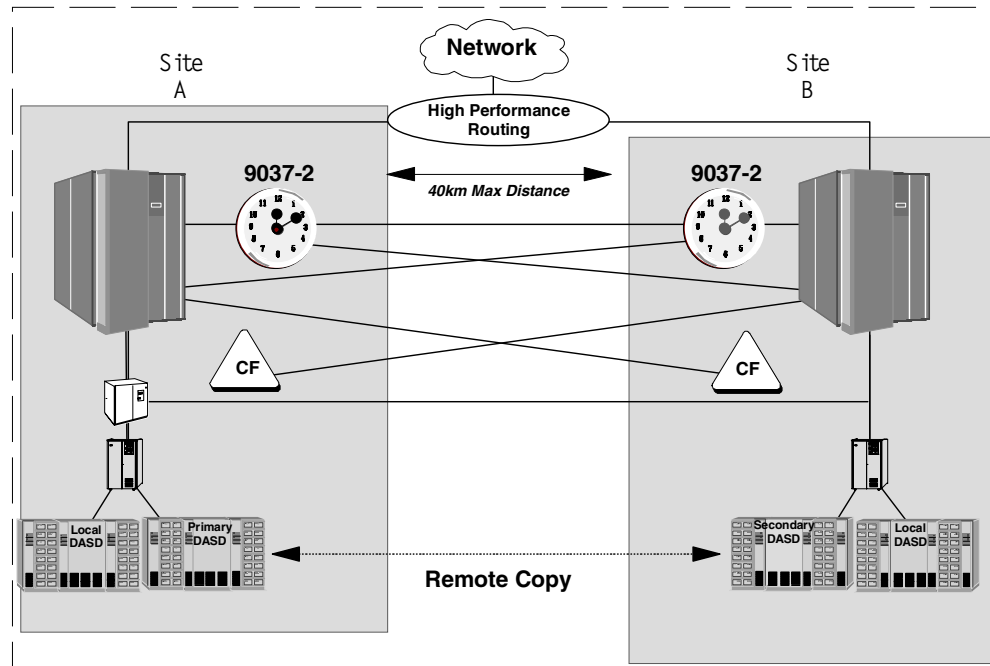


Figure 30. GDPS Architecture

Implementation

GDPS is being implemented as an *automation solution*, using standard sysplex and IBM 3990 Storage Control Unit functions. The base for the implementation of a GDPS is a sysplex spread across two sites, securing diagnostic and control capability in case of a site failure. The two sites may be up to 40 km apart.

The sysplex must be configured to be fault tolerant: this applies to the sysplex control data sets and to the Sysplex Timer and Coupling Facilities (if used). A fault-tolerant Sysplex Timer configuration consists of two interconnected timers, properly connected to all processor complexes that are part of the sysplex.

All data required for an application restart must be DASD resident. All data that is part of the same group of applications must be in one site, and PPRC must be used to maintain a *synchronous* copy of the data in the backup location. Spare processor capacity and/or expendable workload must be available in the secondary site so that enough capacity is available to resume the critical workload in the backup location.

Processing

GDPS processing is initialized based on a GDPS *configuration database* that contains site and configuration details. This allows GDPS to support and automate the routine PPRC configuration management tasks such as setting up links and pairs, to perform an interval driven check on the *current* status of the configuration and compare it against the *target* configuration status.

During normal operations GDPS continuously monitors all systems and specifically looks for messages indicating that PPRC volume pairs are being suspended. At the occurrence of a *suspend*, GDPS immediately *freezes* the image of the secondary disk configuration, to ensure *restartability* of the applications in the backup location.

The next step is to analyze the reason for the suspend, because each cause can have different levels of effect. For instance, if the suspend was caused by a secondary equipment problem, it makes sense not to interrupt primary application processing. However, if the suspend was caused by a primary equipment failure, then GDPS, after having stopped the secondary device updates, will either allow applications to continue or force them to stop. The choice is driven by an *installation policy*. If the failure is part of a disaster unfolding in the primary location, workload restartability is ensured by freezing the secondary data image. If the primary site applications were stopped, no data loss will ever occur.

Automation taking control at a suspend event is possible because the storage server starts an *automation window* when a suspend condition is detected. The write operation that forced the condition to surface is not completed until the automation has taken specific action. The automation window is essential in taking control at the right moment and ensuring data consistency in the backup location.

If there is a need to make the switch to the backup facility, GDPS executes all the mechanics of removing the failed site systems from the sysplex, changing the status of the former secondary disk configuration to bring the primary back up, switching the network to the backup location, reconfiguring processor capacity in the surviving site as required to support the fallback mode of operation, and finally restarting the application.

9.5.4.3 Extended Remote Copy

XRC is the *asynchronous* implementation of remote copy. Copies are also established by disk volume, but there is the concept of *session*, which relates a number of disk volumes that may be associated with the same application. The remote copies are managed by session, and *write sequence* consistency is maintained across all disk volumes. Although the *data currency* at the secondary may *lag* behind the primary by some seconds or minutes, the consistency of the data is preserved even where data is spread across multiple LCUs or storage servers. Figure 31 on page 100 describes the XRC data flow.

Preservation of the write sequence consistency enables easier recovery of any database management system at the secondary site in the event of the disaster. XRC is implemented through the System Data Mover (SDM) function of DFSMS.

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

The ability to perform a DB2 restart means that recovery at the secondary site may be as quick as a recovery from a failure on the production system. The only drawback to an asynchronous implementation of remote copy is that the currency of the data may lag behind the primary system. This may result in some transactions having to be manually reentered after recovery at the secondary

site. XRC externalizes a *timestamp* of the recovered system so that manual recovery is possible from a specified time. The time lag between the primary and the secondary sites can be minimized by performance tuning actions.

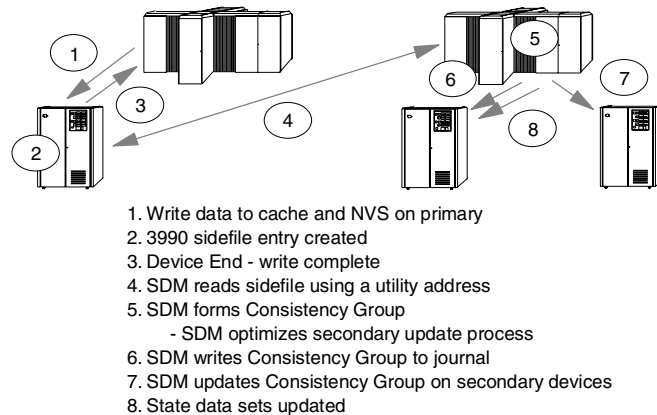


Figure 31. XRC Data Flow

9.5.5 Compression

Host compression techniques are commonly used to reduce the amount of auxiliary storage required. As a general result, not only is storage space saved, but also disk I/O; the data occupies less space; and fewer operations are required to access and transfer the data on channels and networks. The cost is extra CPU cycles needed at the host to compress the data before destaging to storage servers and to decompress the data after it has been retrieved.

DB2 uses host compression and keeps the data compressed in the buffer pools as well, effectively increasing their size, and decompressing only the rows needed by the application programs. DB2 provides utilities that estimate the compression values for your data, and therefore can help when evaluating the trade off between DASD savings and CPU overhead.

Some disk storage servers, like RVA, store the user data in compressed form. In such cases compression and decompression are independent of the host. So the question arises about the usability of both levels of compression. Are they compatible?

The answer is yes: *both can be used!* Obviously, when you use both, the effectiveness of the compression ratio between host data and stored data will be considerably less than the 3.6 general value for traditional data, probably in the range between 1.5 and 2.5, but still greater than 1. The RVA also implements compaction and replaces the traditional device control information (such as gaps and headers) with other techniques. In general, when capacity planning for large storage occupancy, and the real amount of compressed data is not well defined, consider some preliminary analysis of your RVA solution. There are tools available to the IBM storage specialists in order to determine the specific compression ratio by sampling the data of a given environment.

Please refer to *IBM RAMAC Virtual Array, SG24-4951*, and to *DB2 for OS/390 and Data Compression, SG24-5261*, for details on RVA and DB2 compression.

9.5.6 Sequential Data Striping

Sequential data striping provides the opportunity for significant improvement in sequential processing performance by allowing data to be spread across multiple devices that are accessed concurrently *transparently to the applications*. With sequential data striping, the data transfer rate may be substantially higher than the individual device is capable of sustaining.

Sequential data striping is provided as a capability of DFSMS/MVS and it is available only for DFSMS-managed data sets.

Sequential data striping is currently a capability for BSAM and QSAM. Sequential data striping should be considered for the sequential processing component of critical batch jobs, as it may well provide a reduction in elapsed time for those jobs, and it may spread workload onto several paths: it smoothes the intensive sequential I/O activity.

DFSMS 1.5.0 extends striping to the data component of VSAM KSDSs, ESDSs, RRDSs, and VRRDSs (no striping for LDSs yet).

Allocating such VSAM data sets and requesting extended format in the Data Class as striping with sustained data rate in MB/sec in the Storage Class allows striped I/O to the data component. VSAM creates the data component in writing the next control interval to the next stripe in a wraparound way and interspersed among a stripe group. When sequentially reading, VSAM creates and drives as many I/O operations independent of each other as the number of available stripes. Therefore the data aggregate rate for a single data component is determined by multiplying the transfer rate between control unit and CPU by the number of stripes.

Sequential data striping works as described above for sequential processing and when the data is processed for nonshared resources (NSR). Striped I/O does not apply for direct processing. Direct processing reads only one CI at a time, as do local shared resource (LSR) and global shared resource (GSR). Record level sharing (RLS) is also excluded from striped I/O processing. Neither KSDSs with key ranges nor the IMBED attribute qualify for striping.

The maximum number of stripes is 16. If more than one stripe is forced to use the same storage path, I/O processing has to complete before the next I/O is started for another stripe.

When dealing with very large DB2 table spaces to be scanned sequentially, performance can be greatly improved by utilizing partitions and query parallelism. In DB2 environments, sequential data striping is today of some interest for log archiving onto disk, or could be considered for some utility work file activity, or to improve performance for some large standard image copies. A more widespread usage can be envisioned (for instance to improve the bandwidth on active logs) as soon as striping becomes applicable to VSAM LDS and DB2 introduces its exploitation.

Chapter 10. DB2 I/O Operations

The information shown here is extracted and modified from different sections of the *DB2 UDB for OS/390 V6 Administration Guide*, SC26-8957. This information is provided to give storage administrators an understanding of the I/O operations performed by DB2.

The two most important I/O operations performed by DB2 are the data read I/O and the log write I/O. The data read I/O has direct impact on the response time of any SQL query. The log write I/O has an important impact on online transaction response time. This chapter will describe the following I/O operations in detail:

- Data I/O: read and write accesses to DB2 table spaces and index spaces
- Log I/O: read and write accesses to the active log.

Other I/O operations are performed by DB2, like access to image copies, archive logs, and BSDSs. These I/O operations are not described in this chapter.

10.1 Avoiding I/O Operations

One of the basic principles of DB2 design is to avoid I/O operations if at all possible. As Roger Miller, DB2 Lead Architect, often says: "*The best I/O is the one which is avoided completely, the second best is the one that does not go to the disk - just to the cache.*"

DB2 tries to achieve this, using a hierarchy of buffer pools to keep data in memory. Modern disk devices complement this by having large caches which give an additional level of intermediate data storage. This storage hierarchy is illustrated in Figure 32 on page 104.

DB2 uses virtual buffer pools to store the data pages. The virtual buffer pools are optionally backed up by hiper pools. When data sharing is used, group buffer pools in the coupling facility store updated pages before these are cast out to disk. Pages in the group buffer pool can be accessed from any member of the data sharing group.

In addition to the caching done by DB2, the storage controller also uses a cache for data. The controller has algorithms to determine the convenience of pre-staging the data to the cache. For example, if several sequential reads are detected, the controller reads tracks ahead of the requests in order to improve the cache hit ratio.

With DB2 V6, the hierarchy shown in Figure 32 is extended to include virtual buffer pools in data spaces. For more details, refer to *DB2 UDB for OS/390 Version 6 Performance Topics*, SG24-5351.

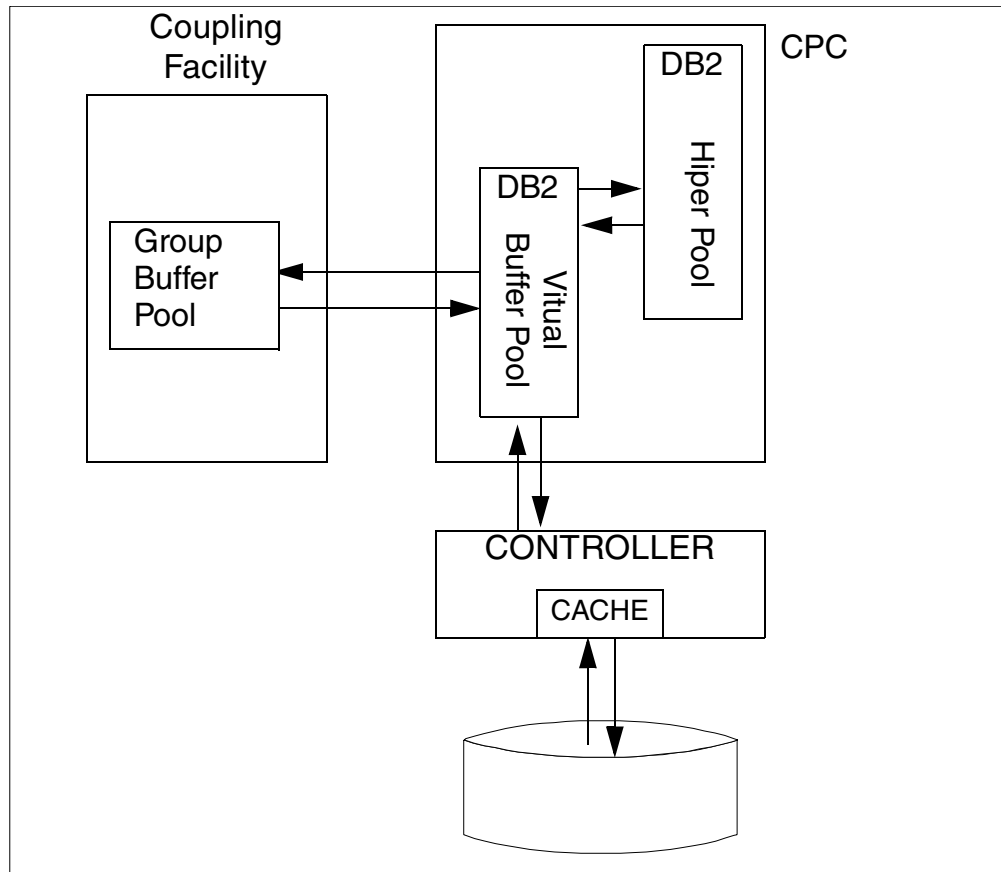


Figure 32. Storage Hierarchy

10.2 Data Read Operations

DB2 uses four read mechanisms to get data pages from disk into the virtual bufferpool:

- Normal read (or synchronous read)
- Sequential prefetch
- Dynamic prefetch
- List sequential prefetch

10.2.1 Normal Read

Normal read is used when just one or a few consecutive pages are retrieved. The unit of transfer for a normal read is one page. This read is referred to in DB2 PM reports as synchronous read. See **A** in Figure 33 on page 109.

10.2.2 Sequential Prefetch

When the optimizer chooses sequential prefetch as access path, sequential prefetch is performed concurrently with other operations of the originating application program. It brings pages into the virtual buffer pool before they are required and reads several pages with a single I/O operation. Because this read executes concurrently and independently of the application program, it is referred to in DB2 PM reports as asynchronous read. See **B** in Figure 33 on page 109. Of course, not all asynchronous I/O can be performed concurrently; there can be

instances that do not have total overlap, in which wait times will still appear in the accounting records.

Sequential prefetch can be used to read data pages, by table space scans or index scans with clustered data reference. It can also be used to read index pages in an index scan. Sequential prefetch allows CP and I/O operations to be overlapped.

Because sequential prefetch reads multiple pages in one I/O operation, it has an important performance advantage over the normal read for applications that process multiple sequential pages. The DB2 virtual buffer pool must be large enough to avoid situations in which prefetched pages are being stolen by another application before they are referenced.

10.2.3 Dynamic Prefetch

Standard sequential prefetch is established at bind time, when the optimizer establishes a sequential access path to the data. Sequential prefetch can also be initiated at execution time. DB2 uses a '*sequential detection*' algorithm to determine that pages are accessed in a sequential access pattern, and which activates sequential prefetch. This type of sequential prefetch is called dynamic prefetch. An algorithm is also used to disable the dynamic prefetch.

Dynamic prefetch occurs when the optimizer establishes a non-sequential access path to the data (for example: `SELECT ... WHERE KEY = :variable`); and the keys provided to the program are many, and in sequential order, or very nearly so. It provides the same advantages as sequential prefetch.

Dynamic prefetch requests are detailed in DB2 PM reports. For an example, see **D** in Figure 33 on page 109.

10.2.4 List Prefetch

List prefetch is used to prefetch data pages that are not contiguous (such as through non-clustered indexes). List prefetch reads a set of data pages that are determined by a list of RIDs taken from an index. Before the read is performed, the RIDs are sorted in sequential order, allowing clustered accesses. List prefetch can also be used by incremental image copy. List prefetch requests are detailed in DB2PM reports. For an example, see **C** in Figure 33 on page 109.

10.2.5 Prefetch Quantity

The sequential, dynamic, and list prefetch operations each read a set of pages. The maximum number of pages read by a request issued from an application program is determined by the size of the buffer pool used.

When the virtual buffer pool is very small, sequential prefetch is disabled. Prefetch is also disabled if the sequential prefetch thresholds is reached. This is explained in 10.2.7, "Sequential Prefetch Threshold" on page 107.

Figure 24 shows the prefetch quantity as a function of the page size and the buffer pool size. For certain utilities (REORG, RECOVER), the prefetch quantity can be twice as much.

Table 24. Number of Pages Read Asynchronously in One Prefetch Request

Page Size	Bufferpool Size (buffers)	Prefetch Quantity (pages/request)
4K	<224	8
	224-999	16
	>=1000	32
8K	<113	4
	113-499	8
	>=500	16
16K	<57	2
	57-249	4
	>=250	8
32K	<100	2
	>=100	4

From the DB2 PM accounting trace, the average number of pages read in prefetch operations by an application program can be calculated. The average number of pages read is the total number of pages read in prefetch operations (**E** in Figure 33 on page 109) divided by the sum of prefetch operations (**B**, **C**, **D** in Figure 33), that is:

$$\text{Average pages read by one prefetch operation} = \mathbf{E} / (\mathbf{B} + \mathbf{C} + \mathbf{D}).$$

The DB2 PM statistics report calculates these values for every type of prefetch operation. Figure 34 on page 110 shows the average pages read by sequential prefetch (**K**), by list prefetch (**L**) and by dynamic prefetch (**M**). These numbers apply to the whole DB2 subsystem, while the accounting report numbers normally refer to one plan.

10.2.6 Data Management Threshold

The data management threshold (DMTH) is set by DB2 at 95% of each virtual buffer pool. The DMTH is maintained and checked independently for each individual virtual buffer pool.

This threshold is checked before a page is read or updated. If the threshold has not been exceeded, DB2 accesses the page in the virtual buffer pool once for each page, no matter how many rows are retrieved or updated in that page. If the threshold has been exceeded, DB2 accesses the page in the virtual buffer pool once for each row that is retrieved or updated in that page. Reaching this threshold has a significant effect on processor usage and performance.

10.2.7 Sequential Prefetch Threshold

The sequential prefetch threshold (SPTH) is set by DB2 at 90% of each virtual buffer pool. This threshold is checked at two different times:

- Before scheduling a prefetch operation. If the threshold has been exceeded, the prefetch is not scheduled.
- During buffer allocation for an already-scheduled prefetch operation. If the threshold has been exceeded, the prefetch is canceled.

When the sequential prefetch threshold is reached, sequential prefetch is disabled until more buffers become available. When this occurs, the performance of operations that use sequential prefetch is adversely affected.

10.3 Data Write Operations

When an application updates data, the updated pages are kept in the virtual buffer pool. Eventually, the updated data pages in the virtual bufferpool have to be written to disk. Write operations can be either asynchronous or synchronous, with the execution of the unit of work.

10.3.1 Asynchronous Writes

Most DB2 writes are done asynchronously from the application program and chained whenever possible. This helps performance and implies that the application may have long since finished, by the time its data updates are written to disk. Updated pages are kept in the virtual bufferpool for possible reuse. The reuse ratio can be obtained from the DB2PM statistics report. See **J** in Figure 35 on page 110, for an example.

Updated pages are written asynchronously, when:

- A checkpoint is taken, which happens whenever:
 - The DB2 parameter LOGLOAD limit is reached.
 - An active log is switched.
 - The DB2 subsystem stops executing normally.
- The percentage of updated pages in a virtual buffer pool for a single data set exceeds a preset limit called the vertical deferred write threshold (VDWQT).
- The percentage of unavailable pages in a virtual buffer pool exceeds a preset limit called the deferred write threshold (DWQT).

Because these operations are independent from the application program, the DB2 accounting trace cannot show these writes. The DB2 PM statistics report is required to see the asynchronous writes. This is shown in **H** in Figure 35 on page 110.

10.3.2 Synchronous Writes

Synchronous writes occur exceptionally, when:

- The virtual buffer pool is too small and the immediate write threshold (IWTH, see 10.3.3, “Immediate Write Threshold” on page 108) is exceeded.
- More than two DB2 checkpoints have been taken during the execution of a unit of work, and an updated page has not been written out to disk.

When the conditions for synchronous write occur, the updated page is written to disk as soon as the update completes. The write is synchronous with the application program SQL request; that is, the application program waits until the write has been completed. These writes are shown in the DB2 accounting trace (See **F** in Figure 33 on page 109) and in the DB2PM statistics report (see **G** in Figure 35 on page 110).

10.3.3 Immediate Write Threshold

The immediate write threshold is set when 97.5% of all pages in the virtual buffer pool are unavailable, and cannot be changed. Monitoring buffer pool usage includes checking how often this threshold is reached. Generally, you want to set virtual buffer pool sizes large enough to avoid reaching this threshold.

Reaching this threshold has a significant effect on processor usage and I/O resource consumption. For example, updating three rows per page in 10 sequential pages ordinarily requires one or two asynchronous write operations. When IWTH is exceeded, the updates require 30 synchronous writes.

10.3.4 Write Quantity

DB2 writes a variable number of pages in each I/O operation. Table 25 on page 108 shows the maximum pages DB2 can write in a single asynchronous I/O operation. Some utilities can write twice the amount shown in this figure. The actual number of pages written in a time interval can be obtained from the DB2 PM statistics report. For an example, see **I** in Figure 35 on page 110.

Table 25. Maximum Pages in One Write Operation

Page Size	Maximum Pages
4K	32
8K	16
16K	8
32K	4

10.3.5 Tuning Write Frequency

Large virtual buffer pools benefit DB2 by keeping data pages longer in storage, thus avoiding an I/O operation. With large buffer pools and very high write thresholds, DB2 can write large amounts of data at system checkpoint time and impact performance.

The DB2 administrators can tune virtual buffer pool parameters to cause more frequent writes to disk and reduce the impact of the writes at system checkpoint. The tuning parameters are the DWQT and the VDWQT. The DWQT works at virtual buffer pool level, while the VDWQT works at data set level.

Table spaces containing pages which are frequently reread and updated should have a high threshold, placing them in a virtual buffer pool with a high DWQT, or high VDWQT. This ensures that pages are reused in storage. The reference value **J** in Figure 35 on page 110 shows the rate of updates per each write. The higher this rate, the better the page reuse for write is in this virtual buffer pool.

Large table spaces, where updates are very scattered and page reuse is infrequent or improbable, can have their threshold set low, even to zero. A zero threshold means that updated pages are written to disk very frequently. In this case, the probability of finding the update page still on the disk cache is higher (cache hit) helping with disk performance. A low threshold also reduces the write impact at checkpoint time.

TOT4K	TOTAL	
-----	-----	
BPOOL HIT RATIO (%)	2	
GETPAGES	6135875	
BUFFER UPDATES	48	
SYNCHRONOUS WRITE	0	F
SYNCHRONOUS READ	19559	A
SEQ. PREFETCH REQS	164649	B
LIST PREFETCH REQS	0	C
DYN. PREFETCH REQS	26065	D
PAGES READ ASYNCHR.	5943947	E
HPOOL WRITES	0	
HPOOL WRITES-FAILED	0	
PAGES READ ASYN-HPOOL	0	
HPOOL READS	0	
HPOOL READS-FAILED	0	

Figure 33. DB2PM Accounting Trace Buffer Pool Report Extract

Care must be taken if trying to tune the write efficiency with the LOGLOAD value.

DB2 checkpoint performance can be adversely impacted by the LOGLOAD value set too high. LOGLOAD is the installation parameter that establishes the number of LOG control intervals generated before taking a checkpoint. If this value is excessive, a large amount of disk writing takes place at checkpoint, and the DB2 restart time in case of failure is also impacted. With DB2 V6 the LOGLOAD value can be dynamically changed to reflect changes in the workload. See *DB2 UDB for OS/390 Version 6 Performance Topics*, SG24-5351, for more information.

BP4	READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
-----		-----	-----	-----	-----
	BPOOL HIT RATIO (%)	55.12			
	GETPAGE REQUEST	221.8K	6534.43	N/C	110.9K
	GETPAGE REQUEST-SEQUENTIAL	18427.00	542.99	N/C	9213.50
	GETPAGE REQUEST-RANDOM	203.3K	5991.43	N/C	101.7K
	SYNCHRONOUS READS	613.00	18.06	N/C	306.50
	SYNCHRON. READS-SEQUENTIAL	64.00	1.89	N/C	32.00
	SYNCHRON. READS-RANDOM	549.00	16.18	N/C	274.50
	GETPAGE PER SYN.READ-RANDOM	370.36			
	SEQUENTIAL PREFETCH REQUEST	577.00	17.00	N/C	288.50
	SEQUENTIAL PREFETCH READS	577.00	17.00	N/C	288.50
	PAGES READ VIA SEQ.PREFETCH	18440.00	543.38	N/C	9220.00
	S.PRF.PAGES READ/S.PRF.READ	31.96	K		
	LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
	LIST PREFETCH READS	0.00	0.00	N/C	0.00
	PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
	L.PRF.PAGES READ/L.PRF.READ	N/C	L		
	DYNAMIC PREFETCH REQUESTED	2515.00	74.11	N/C	1257.50
	DYNAMIC PREFETCH READS	2515.00	74.11	N/C	1257.50
	PAGES READ VIA DYN.PREFETCH	80470.00	2371.23	N/C	40.2K
	D.PRF.PAGES READ/D.PRF.READ	32.00	M		

Figure 34. DB2 PM Statistic Report Buffer Pool Reads

BP1	WRITE OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
-----		-----	-----	-----	-----
	BUFFER UPDATES	15179.00	1517.84	410.24	0.67
	PAGES WRITTEN	4608.00	0.00	N/C	N/C
	BUFF.UPDATES/PAGES WRITTEN	J	3.29		
	SYNCHRONOUS WRITES	G	0.00	0.00	N/C
	ASYNCHRONOUS WRITES	H	187.00	18.70	5.05
	PAGES WRITTEN PER WRITE I/O	I	24.64		
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00	N/C	N/C
	VERTI.DEF.WRITE THRESHOLD	0.00	0.00	N/C	N/C
	DM CRITICAL THRESHOLD	0.00	0.00	N/C	N/C
	WRITE ENGINE NOT AVAILABLE	0.00	0.00	N/C	N/C

Figure 35. DB2 PM Statistic Report Buffer Pool Writes

```

DSNB450I =DB2Z TABLESPACE = DSND06.SYSCOPY, USE COUNT = 0, GBP-DEP = N
DSNB452I =DB2Z STATISTICS FOR DATASET 1 -
DSNB453I =DB2Z VP CACHED PAGES -
                CURRENT      =    64 MAX                =    64
                CHANGED      =     0 MAX                =     0
DSNB455I =DB2Z SYNCHRONOUS I/O DELAYS -
                AVERAGE DELAY =     9 MAXIMUM DELAY    =    22
                TOTAL PAGES   =     3
DSNB456I =DB2Z ASYNCHRONOUS I/O DELAYS -
                AVERAGE DELAY =     1 MAXIMUM DELAY    =     1
                TOTAL PAGES   =    61 TOTAL I/O COUNT =     2

```

Figure 36. Display Buffer Pool Data Set Statistics

10.4 Log Writes

Log records are created by application programs when data is updated. Each data update requires two log records, one with the data before the update, and another with the data after the update, generally combined into one physical record.

The application program uses two methods (see Figure 37 on page 113) to move log records to the log output buffer:

- NO WAIT
- FORCE

NO WAIT

Most log records are moved to the log output buffer, and control is immediately returned to the application program. These moves are the most common. If no log buffer is available, the application must wait for one to become available. Log records moved into the output buffer by an application program appear in a DB2 PM statistics report as the number of NOWAIT requests. See **C** in Figure 41 on page 118.

FORCE

At commit time, the application must wait to ensure that all changes have been written to the log. In this case, the application forces a write of the current and previous unwritten buffers to disk. Because the application waits for this to be completed, it is also called a synchronous write.

Physical Writes

Figure 37 on page 113 also shows the physical writes to disk. The log records in the log output buffer are written from the output buffer to disk. DB2 uses two types of log writes: asynchronous and synchronous, which will be explained further. These writes to the active log data set are shown by DB2 PM as **E** in Figure 41 on page 118.

10.4.1 Asynchronous Writes

DB2 writes the log records (the control intervals) from the output buffer to the active log data set when the number of log buffers used reaches the value the installation set for the WRITE THRESHOLD field of installation panel DSNTIPL; see Figure 40 on page 115. The application is not aware of these writes.

10.4.2 Synchronous Writes

Synchronous writes usually occur at commit time when an application has updated data. This write is called *forcing* the log, because the application must wait for DB2 to write the log buffers to disk before control is returned to the application. If the log data set is not busy, all log buffers are written to disk. If the log data set is busy, the requests are queued until it is freed.

10.4.3 Writing to Two Logs

If there are two logs (recommended for availability), the write to the first log, in general, must complete before the write to the second log begins. The first time a log control interval is written to disk, the write I/Os to the log data sets are done in parallel. However, if the same 4 KB log control interval is again written to disk, then the write I/Os to the log data sets must be done serially to prevent any possibility of losing log data in case of I/O errors occurring on both copies simultaneously. This method improves system integrity. I/O overlap in dual logging occurs whenever multiple log control intervals have to be written; for example, when the WRITE THRESHOLD value is reached, or when log records accumulate because of a log device busy condition

10.4.4 Two-Phase Commit Log Writes

IMS applications with DB2, and CICS and RRS applications with additional resources besides DB2 to manage, use two-phase commit protocol. Because they use two-phase commit, these applications force writes to the log twice, as shown in Figure 38 on page 113. The first write forces all the log records of changes to be written (if they have not been written previously because of the write threshold being reached). The second write writes a log record that takes the unit of recovery into an in-commit state.

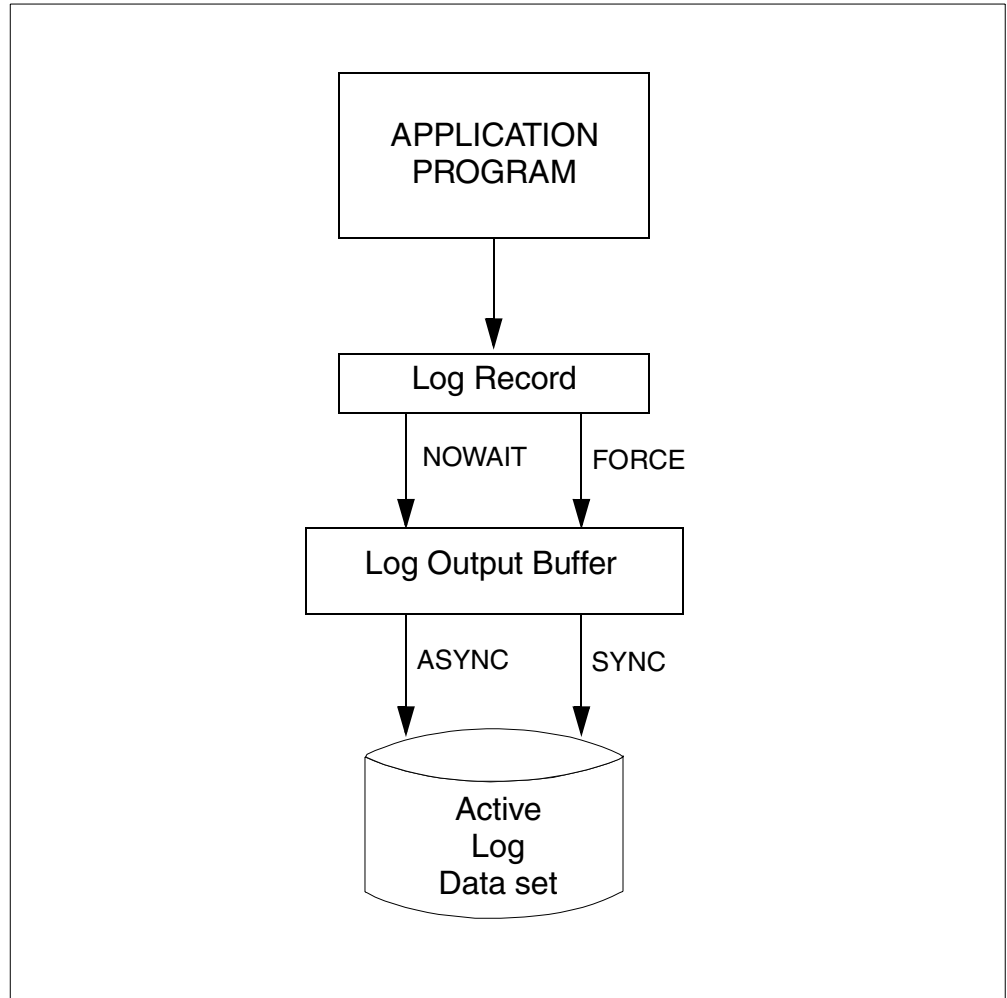


Figure 37. Log Record Path to Disk

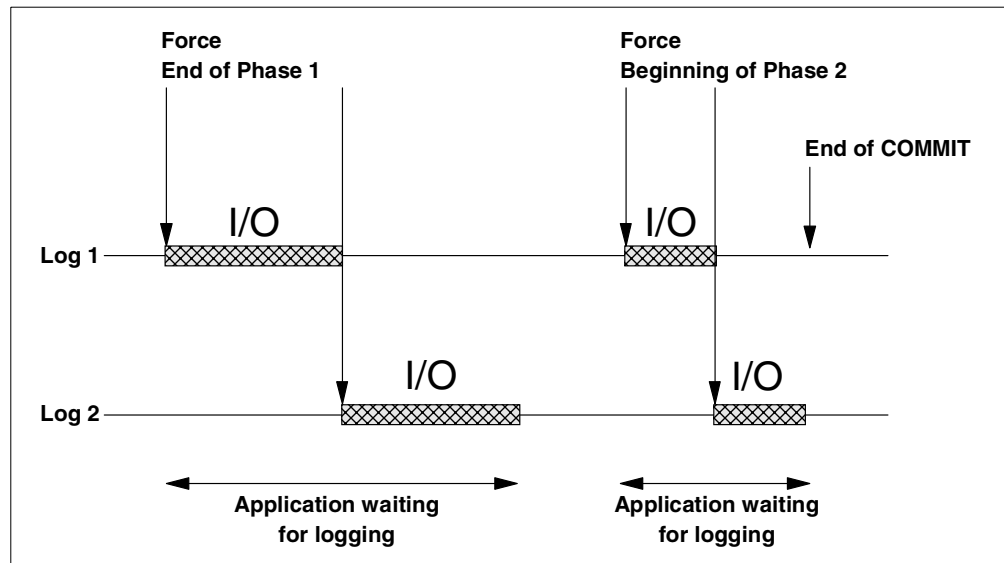


Figure 38. Two-Phase Commit with Dual Active Logs

10.4.5 Improving Log Write Performance

In this section we present some considerations on choices to improve log write performance.

LOG OUTPUT BUFFER Size

The OUTPUT BUFFER field of installation panel DSNTIPL lets the system administrator specify the size of the output buffer used for writing active log data sets. This field is shown in Figure 40 on page 115. With DB2 V6, the maximum size of this buffer (OUTBUFF) is 400000 KB. Choose as large a size as the MVS system can support without incurring additional paging. A large buffer size will improve both log read and log write performance. If the DB2 PM statistics report shows a non-zero value for **B** in Figure 41 on page 118, the log output buffer is too small.

WRITE THRESHOLD

The WRITE THRESHOLD field of installation panel DSNTIPL (see Figure 40 on page 115) indicates the number of contiguous 4KB output buffer pages that are allowed to fill before data is written to the active log data set. The default is 20 buffers, and this is recommended. Never choose a value that is greater than 20% of the number of buffers in the output buffer.

Devices for Log Data Sets

The devices assigned to the active log data sets must be fast. In a transactional environment, the DB2 log may have a very high write I/O rate and will have direct impact on the transaction response time. In general, log data sets can make effective use of the DASD Fast Write feature of IBM's 3990 cache.

Avoid Device Contention

To avoid contention on the disks containing active log data sets, place the data sets so that the following objectives are achieved:

- Define log data set on dedicated volumes
- If dual logging is used, separate the access path for primary and secondary log data sets
- Separate the access path of the primary log data sets from the next log data set pair

Do not place any other data sets on disks containing active log data sets. Place the copy of the bootstrap data set and, if using dual active logging, the copy of the active log data sets, on volumes that are accessible on a path different from that of their primary counterparts. Place sequential sets of active log data sets on different access paths to avoid contention while archiving. To achieve all this, a minimum of three volumes on separate access paths is required for the log data sets. A simple example is illustrated in Figure 39 on page 115.

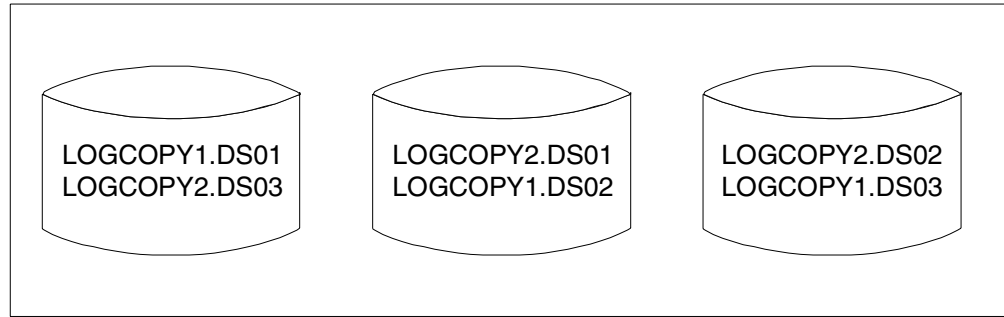


Figure 39. Minimum Active Log Data Set Distribution

Preformat New Active Log Data Sets

The system administrator, when allocating new active log data sets, can preformat them using the DSNJLOGF utility described in Section 3 of *DB2 for OS/390 Utility Guide and Reference*, SC26-8967. This avoids the overhead of preformatting the log, which normally occurs at unpredictable times.

```

DSNTIPL          UPDATE DB2 - ACTIVE LOG DATA SET PARAMETERS
====>

Enter data below:

 1 NUMBER OF LOGS   ====> 3          Data sets per active log copy (2-31)
 2 INPUT BUFFER     ====> 60K         Size in bytes (28K-60K)
 3 OUTPUT BUFFER    ====> 4000K       Size in bytes (40K-400000K)
 4 WRITE THRESHOLD  ====> 20         Buffers filled before write (1-256)
 5 ARCHIVE LOG FREQ  ====> 24         Hours per archive run
 6 UPDATE RATE      ====> 3600        Updates, inserts, and deletes per hour
 7 LOG APPLY STORAGE ====> 0M         Maximum ssmDEMI storage in MB for
                                       fast log apply (0-100M)

F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT   F11=RIGHT   F12=RETRIEVE

```

Figure 40. Installation Panel DSNTIPL

10.5 Log Reads

It is during a rollback, restart, and database recovery that the performance impact of log reads becomes evident. DB2 must read from the log and apply changes to the data on disk. Every process that requests a log read has an input buffer dedicated to that process. DB2 optimizes the log reads searching for log records in the following order:

1. Log output buffer
2. Active log data set
3. Archive log data set

If the log records are in the output buffer, DB2 reads the records directly from that buffer. If the log records are in the active or archive log, DB2 moves those log

records into the input buffer used by the reading process (such as a recovery job or a rollback).

From a performance point of view, it is always best for DB2 to obtain the log records from the output buffer. These accesses are reported by DB2 PM; see **F** in Figure 41 on page 118. The next fastest access for DB2 is the active log; see **G** in Figure 41. Access to the archive log is not desirable; it can be delayed for a considerable length of time. For example, tape drives may not be available, or a tape mount can be required. A zero value for **A** in Figure 41 indicates that the active logs are sized adequately.

10.5.1 Improving Log Read Performance

In this section we present some considerations on choices to improve log read performance.

Active Log Size

Active logs should be large enough to avoid reading the archives, especially during restart, rollback, and recovery. When data is backed out, performance is optimal if the data is available from the output buffer or from the active log. If the data is no longer available from the active log, the active log is probably too small. For information about sizing the active log data sets, see 10.5.2, “Active Log Size” on page 117.

Log Input Buffer

The default size for the input buffer is 60 KB. It is specified in the INPUT BUFFER field of installation panel DSNTIPL (see Figure 40 on page 115). The default value is recommended.

Avoid Device Contention

Avoid device contention on the log data sets. See the recommendation made in 10.4.5, “Improving Log Write Performance” on page 114.

Archive to Disk or Tape

If the archive log data set resides on disk, it can be shared by many log readers. In contrast, an archive on tape cannot be shared among log readers. Although it is always best to avoid reading archives altogether, if a process must read the archive, that process is serialized with anyone else who must read the archive tape volume. For example, every rollback that accesses the archive log must wait for any previous rollback work that accesses the same archive tape volume to complete.

Archiving to disk offers several advantages:

- Recovery times can be reduced by eliminating tape mounts and rewind time for archive logs kept on tape.
- Multiple RECOVER utilities can be run in parallel.
- DB2 log data can span a greater length of time than what is currently kept in your active log data sets.
- Need for tape drives during DB2 archive log creation is eliminated. If DB2 needs to obtain a tape drive on which to create the archive logs and it cannot allocate one, all activity will stop until DB2 can create the archive log data sets.

If you allow DB2 to create the archive log data sets on RVA disks, you can take advantage of the compression capability offered by the device.

Depending on the type of application data DB2 is processing and storing in the log data sets, you could obtain a very good reduction in DASD occupancy with RVA and achieve good recoverability at a reasonable price. This is explained in more detail in *DB2 for OS/390 and Data Compression*, SG24-5261.

Archive to Disk and Tape

DB2 V5 has introduced the option to archive one copy of the log to disk and the other one to tape. This allows more flexibility than when archiving only to tapes and disk space savings when compared to archiving only to disk.

- In case of unavailability of tape units, you can, in fact, cancel the request for allocation (having previously set the WRITE TO OPER parameter to YES in the Archive Log Installation Panel reported in Figure 21 on page 70) and let DB2 continue with a single archiving.
- Disk space utilization is improved by reducing the number of data sets for the dual copy of active logs to one copy of the archive log data set on disk and one on tape.

10.5.2 Active Log Size

The capacity the system administrator specifies for the active log can affect DB2 performance significantly. If the capacity is too small, DB2 might need to access data in the archive log during rollback, restart, and recovery. Accessing an archive log generally takes longer than accessing an active log. An active log which is too small is shown by a non-zero value in **A** in Figure 41 on page 118.

Log Sizing Parameters

The following DB2 parameters affect the capacity of the active log. In each case, increasing the value the system administrator specifies for the parameter increases the capacity of the active log. See Section 2 of the *DB2 Installation Guide*, for more information on updating the active log parameters. The parameters are:

The NUMBER OF LOGS field on the installation panel DSNTIPL (see Figure 40 on page 115) controls the number of active log data sets.

The ARCHIVE LOG FREQ field on the installation panel DSNTIPL (see Figure 40) controls how often active log data sets are copied to the archive log.

The UPDATE RATE on the installation panel DSNTIPL (see Figure 40) is an estimate of how many database changes (inserts, update, and deletes) are expected per hour.

The CHECKPOINT FREQ on the installation panel DSNTIPN specifies the number of log records that DB2 writes between checkpoints.

The DB2 installation CLIST uses UPDATE RATE and ARCHIVE LOG FREQ to calculate the data set size of each active log data set.

Calculating Average Log Record Size

One way to determine how much log volume is needed is to calculate the average size in bytes of log records written. To do this, the DB2 system administrator

needs values from the statistics report shown in Figure 41: the NOWAIT counter **C**, and the number of control intervals created in the active log, counter **D**. Use the following formula:

$$\text{avg size of log record in bytes} = \mathbf{D} * 4096 / \mathbf{C}$$

Using this value to estimate logging needs, plus considering the available device sizes, the DB2 system administrator can update the output of the installation CLIST to modify the calculated values for active log data set sizes.

LOG ACTIVITY	QUANTITY	/MINUTE	/THREAD	/COMMIT
READS SATISFIED-OUTPUT BUFF	15756.00	F 11.21	2.82	0.07
READS SATISFIED-OUTP.BUF(%)	100.00			
READS SATISFIED-ACTIVE LOG	0.00	G 0.00	0.00	0.00
READS SATISFIED-ACTV.LOG(%)	0.00			
READS SATISFIED-ARCHIVE LOG	0.00	A 0.00	0.00	0.00
READS SATISFIED-ARCH.LOG(%)	0.00			
TAPE VOLUME CONTENTION WAIT	0.00	0.00	0.00	0.00
WRITE-NOWAIT	C 2019.6K	1437.45	361.15	8.70
WRITE OUTPUT LOG BUFFERS	E 250.3K	178.17	44.76	1.08
BSDS ACCESS REQUESTS	2041.00	1.45	0.36	0.01
UNAVAILABLE OUTPUT LOG BUFF	B 0.00	0.00	0.00	0.00
CONTR. INTERV. CREATED-ACTIVE	59442.00	D 42.31	10.63	0.26
ARCHIVE LOG READ ALLOCATION	0.00	0.00	0.00	0.00
ARCHIVE LOG WRITE ALLOCAT.	2.00	0.00	0.00	0.00
CONTR. INTERV. OFFLOADED-ARCH	65023.00	46.28	11.63	0.28
READ DELAYED-UNAVAIL.RESOUR	0.00	0.00	0.00	0.00
LOOK-AHEAD MOUNT ATTEMPTED	0.00	0.00	0.00	0.00
LOOK-AHEAD MOUNT SUCCESSFUL	0.00	0.00	0.00	0.00

Figure 41. Log Statistics in a Sample DB2 PM Statistics Report

Chapter 11. I/O Performance and Monitoring Tools

This chapter addresses I/O performance reporting and monitoring tools in relation to storage management in a DB2 environment. The following tools are described:

- DB2 Performance Monitor (DB2 PM)
- Resource Monitoring Facility (RMF)
- IBM Extended Facilities Product (IXFP) for RVA monitoring

Figure 42 on page 119 illustrates the scope of these tools.

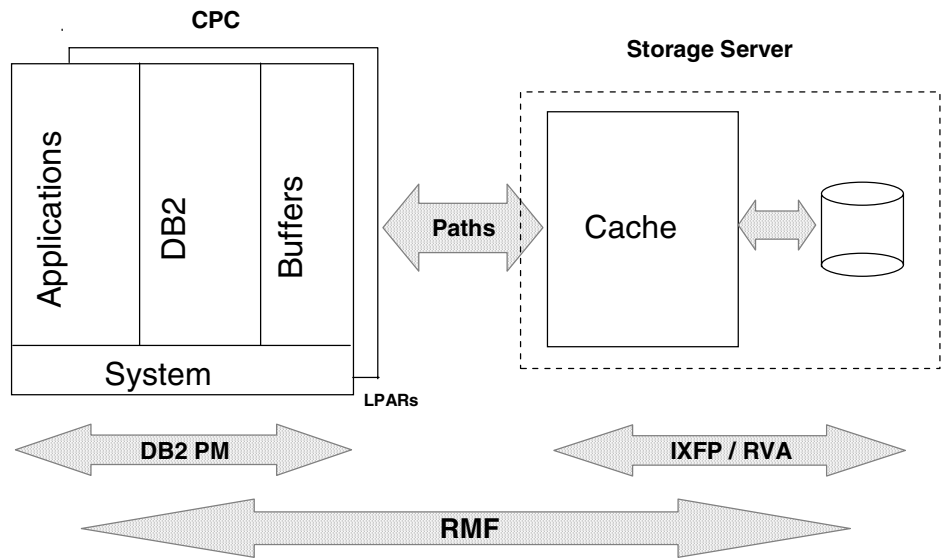


Figure 42. Scope of Performance Analysis Tools

11.1 DB2 PM Overview

DB2 generates data about its own performance, called instrumentation data, but it has no reporting facility to analyze this data. The entries in the DB2 installation panel DSNTIPN, reported in Figure 43 on page 120, activate audit, global, accounting, and monitoring traces as well as checkpoint frequency. See Section 5 of the *DB2 for OS/390 Administration Guide*, SC26-8957, for more information on the trace categories.

DB2 PM provides the capability to gather, analyze, and report on DB2 instrumentation data. DB2 PM can report performance information online and in batch.

The DB2 instrumentation data creates several types of trace records. The traces relevant to I/O analysis are:

- Accounting Trace
- Statistics Trace
- Performance Trace

Statistics and accounting traces are collected in most installations. A performance trace is collected when a specific problem has to be investigated. Activating the performance trace has a significant impact on DB2 subsystem performance. The user can cause more or less information to be collected by these traces, by specifying trace classes to be activated.

An accounting trace provides information at an identifier level. Examples of identifiers are plans, packages, users, or connection types. Accounting information can be a summary of multiple executions (Accounting Report), or it can be a detail of every execution (Accounting Trace).

A statistics trace provides information at DB2 subsystem level. It can be a summary of multiple statistic intervals (Statistics Report) or it can be a listing of each interval (Statistics Trace).

The performance trace can generate detailed information on all DB2 subsystem activity. When this trace has been started with the appropriate classes, DB2 PM can generate an I/O activity report.

```

DSNTIIPN          UPDATE DB2 - TRACING AND CHECKPOINT PARAMETERS
====>

Enter data below:

  1  AUDIT TRACE          ==> NO          Audit classes to start. NO,YES,list
  2  TRACE AUTO START    ==> NO          Global classes to start. YES, NO, list
  3  TRACE SIZE          ==> 64K        Trace table size in bytes. 4K-396K
  4  SMF ACCOUNTING      ==> 1          Accounting classes to start. NO,YES,list
  5  SMF STATISTICS      ==> YES        Statistics classes to start. NO,YES,list
  6  STATISTICS TIME     ==> 30        Time interval in minutes. 1-1440
  7  DATASET STATS TIME ==> 5          Time interval in minutes. 1-1440
  8  MONITOR TRACE       ==> NO          Monitor classes to start. NO, YES, list
  9  MONITOR SIZE        ==> 8K        Default monitor buffer size. 8K-1M
 10  CHECKPOINT FREQ     ==> 50000      Number of log records per checkpoint
 11  UR CHECK FREQ       ==> 0          Checkpoints to enable UR check. 0-255
 12  LIMIT BACKOUT       ==> AUTO      Limit backout processing. AUTO,YES,NO
 13  BACKOUT DURATION    ==> 5          Checkpoints processed during backout if
                                         LIMIT BACKOUT = AUTO or YES. 0-255.
 14  RO SWITCH CHKPTS    ==> 5          Checkpoints to read-only switch. 1-32767
 15  RO SWITCH TIME      ==> 10        Minutes to read-only switch. 0-32767
 16  LEVELID UPDATE FREQ ==> 5          Checkpoints between updates. 0-32767
    F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND   F6=RCHANGE
    F7=UP        F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT  F12=RETRIEVE

```

Figure 43. Installation Panel DSNTIIPN

11.1.1 Accounting I/O Information

11.1.1.1 I/O Operations

The data I/O operations initiated by DB2 on behalf of an application are detailed in the buffer pool sections of the DB2 PM accounting report. Each bufferpool is shown independently. For an example, see Figure 44 on page 121. Every read access is reported as a getpage. Every write is reported as a buffer update. The getpages and buffer updates may initiate an I/O. These I/Os are either synchronous or asynchronous. The asynchronous reads are reported in the three prefetch fields (SEQ, LIST, DYN).

BP4	TOTAL
-----	-----
BPOOL HIT RATIO (%)	50
GETPAGES	300350
BUFFER UPDATES	0
SYNCHRONOUS WRITE	0
SYNCHRONOUS READ	754
SEQ. PREFETCH REQS	702
LIST PREFETCH REQS	0
DYN. PREFETCH REQS	3944
PAGES READ ASYNCHR.	148634
HPOOL WRITES	0
HPOOL WRITES-FAILED	0
PAGES READ ASYN-HPOOL	0
HPOOL READS	0
HPOOL READS-FAILED	0

Figure 44. DB2 PM Accounting, Buffer Pool Section

11.1.1.2 I/O Suspensions

If accounting trace class 3 is activated, the DB2 accounting reports show a summary of the wait times of a DB2 application. This summary shows when the DB2 application is suspended (waiting) for a DB2 system task to complete. These suspensions include the waits for I/O operations.

Figure 64 on page 145 shows an example of class 3 suspend times. The values shown for **A**, **B**, **C** are total elapsed times for I/O and the number of occurrences of each type of I/O (events).

- A: SYNCHRON. I/O** The total elapsed time due to synchronous I/O and the total number of synchronous I/O suspensions.
- B: OTHER READ I/O** The total waiting time due to asynchronous read I/O and the total number of suspensions due to asynchronous read I/O.
- C: OTHER WRTE I/O** The total elapsed time due to asynchronous write I/O and the total number of asynchronous write I/O suspensions.

Note: Another example of *other write I/O* is the case in which a transaction is waiting because the page that the transaction wants to update is currently being written out by a write engine. In this case, the wait time is captured in other write I/O.

Note: Wait time for force at commit time is included in SYNC I/O WAIT with V5 and LOG FORCE WRITE WAIT with V6.

11.1.2 Statistics I/O Information

11.1.2.1 Data I/O Operations

The statistics report provides detailed information on I/O operations for each buffer pool at the DB2 subsystem level. This means that this report summarizes the information of all applications executing during the interval that statistics are gathered. The information provided is much more detailed than the information in

the accounting report. Some additional information is calculated in this report, for example, in Figure 45 it shows the average number of pages for each type of prefetch read.

BP4	READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT

	BPOOL HIT RATIO (%)	55.12			
	GETPAGE REQUEST	221.8K	6534.43	N/C	110.9K
	GETPAGE REQUEST-SEQUENTIAL	18427.00	542.99	N/C	9213.50
	GETPAGE REQUEST-RANDOM	203.3K	5991.43	N/C	101.7K
	SYNCHRONOUS READS	613.00	18.06	N/C	306.50
	SYNCHRON. READS-SEQUENTIAL	64.00	1.89	N/C	32.00
	SYNCHRON. READS-RANDOM	549.00	16.18	N/C	274.50
	GETPAGE PER SYN.READ-RANDOM	370.36			
	SEQUENTIAL PREFETCH REQUEST	577.00	17.00	N/C	288.50
	SEQUENTIAL PREFETCH READS	577.00	17.00	N/C	288.50
	PAGES READ VIA SEQ.PREFETCH	18440.00	543.38	N/C	9220.00
	S.PRF.PAGES READ/S.PRF.READ	31.96			
	LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
	LIST PREFETCH READS	0.00	0.00	N/C	0.00
	PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
	L.PRF.PAGES READ/L.PRF.READ	N/C			
	DYNAMIC PREFETCH REQUESTED	2515.00	74.11	N/C	1257.50
	DYNAMIC PREFETCH READS	2515.00	74.11	N/C	1257.50
	PAGES READ VIA DYN.PREFETCH	80470.00	2371.23	N/C	40.2K
	D.PRF.PAGES READ/D.PRF.READ	32.00			
	PREF.DISABLED-NO BUFFER	0.00	0.00	N/C	0.00
	PREF.DISABLED-NO READ ENG	0.00	0.00	N/C	0.00
	SYNC.HPOOL READ	0.00	0.00	N/C	0.00
	ASYN.HPOOL READ	0.00	0.00	N/C	0.00
	HPOOL READ FAILED	0.00	0.00	N/C	0.00
	ASYN.DA.MOVER HPOOL READ-S	0.00	0.00	N/C	0.00
	ASYN.DA.MOVER HPOOL READ-F	0.00	0.00	N/C	0.00
	PAGE-INS REQUIRED FOR READ	59.00	1.74	N/C	29.50

Figure 45. DB2 PM Statistics, Buffer Pool Read Operations Section

11.1.2.2 Log Activity

The log activity report gives detailed information about log I/O. Figure 46 on page 123 shows a log activity section from a statistics report.

The block of lines identified by **A** in Figure 46 indicate read access. The different values indicate statistics for each possible source of log records. For performance reasons, most of the reads should be satisfied from the output buffer: the archive log should only be used for exceptional circumstances. Please refer to 10.5, “Log Reads” on page 115 for details.

The block of lines identified by **B** in Figure 46 indicates write access. This is explained in 10.4, “Log Writes” on page 111.

Line **C** in Figure 46 shows BSDS accesses. Just like the active log accesses, these accesses are mainly writes.

The block of lines starting with **D** shows volume of records created in the active log and offloaded by the archiving process.

The block of lines starting with **E** shows archive volume mounting information.

LOG ACTIVITY		QUANTITY	/MINUTE	/THREAD	/COMMIT
-----		-----	-----	-----	-----
READS SATISFIED-OUTPUT BUFF	A	0.00	0.00	N/C	0.00
READS SATISFIED-OUTP.BUF(%)		N/C			
READS SATISFIED-ACTIVE LOG		0.00	0.00	N/C	0.00
READS SATISFIED-ACTV.LOG(%)		N/C			
READS SATISFIED-ARCHIVE LOG		0.00	0.00	N/C	0.00
READS SATISFIED-ARCH.LOG(%)		N/C			
TAPE VOLUME CONTENTION WAIT		0.00	0.00	N/C	0.00
WRITE-NOWAIT	B	0.00	0.00	N/C	0.00
WRITE OUTPUT LOG BUFFERS		509.00	15.00	N/C	254.50
BSDS ACCESS REQUESTS	C	0.00	0.00	N/C	0.00
UNAVAILABLE OUTPUT LOG BUFF		0.00	0.00	N/C	0.00
CONTR.INTERV.CREATED-ACTIVE	D	5.00	0.15	N/C	2.50
ARCHIVE LOG READ ALLOCATION		0.00	0.00	N/C	0.00
ARCHIVE LOG WRITE ALLOCAT.		0.00	0.00	N/C	0.00
CONTR.INTERV.OFFLOADED-ARCH		0.00	0.00	N/C	0.00
READ DELAYED-UNAVAIL.RESOUR	E	0.00	0.00	N/C	0.00
LOOK-AHEAD MOUNT ATTEMPTED		0.00	0.00	N/C	0.00
LOOK-AHEAD MOUNT SUCCESSFUL		0.00	0.00	N/C	0.00

Figure 46. DB2 PM Statistics, Log Activity Section

11.1.3 Performance I/O Information and I/O Activity

DB2 PM can create several I/O activity reports for different types of data sets. The buffer pool report shows the activity of one identifier (for example: plan, user, buffer pool id) against the data sets in one virtual buffer pool.

Before the report can be generated, the appropriate trace must be started. It is recommended to limit the trace to reduce the impact on system performance. This can be done by specifying class and instrumentation facility component identifiers (IFCIDs) in the START TRACE command. Asynchronous I/O activity is not collected when user (AUTHID) or application identifiers (PLAN) are specified in the START TRACE command unless the user identifier of the DBM1 address space is also listed. Table 26 on page 124 shows the requirements to generate the trace information for the different I/O activity reports. Figure 47 on page 124 shows an extract from a summary I/O activity report. A detail report allows an analysis at the identifier level. For example, it can be used for a detailed analysis of the accesses of one application to one table space.

Table 26. Trace Requirement for the I/O Activity Reports

I/O Activity Report	DB2 Trace	Class	IFCID
Buffer Pool	Performance	4	6, 7, 8, 9, 10, 105, 107
EDM Pool	Performance	4	29, 30, 105, 107
Active Log	Performance	5	34, 35, 36, 37, 38, 39
Archive Log/BSDS	Performance	5	34, 35, 36, 37, 40, 41, 114, 115, 116, 119, 120
Cross Invalidation	Performance	21	105, 107, 255

BUFFER POOL	TOTALS	AET
-----	-----	-----
TOTAL I/O REQUESTS	51	0.019885
TOTAL READ I/O REQUESTS	51	0.019885
NON-PREFETCH READS	51	
PREFETCH READS		
WITHOUT I/O	0	
WITH I/O	0	
PAGES READ	0	
PAGES READ / SUCC READ	0.00	
TOTAL WRITE REQUESTS	0	
SYNCHRONOUS WRITES	0	N/C
COUPLING FACILITY CASTOUTS	0	N/C
PAGES WRITTEN PER WRITE	0.00	
ASYNCHRONOUS WRITES	0	N/C
COUPLING FACILITY CASTOUTS	0	N/C
PAGES WRITTEN PER WRITE	0.00	

Figure 47. Buffer Pool Section from I/O Activity Summary Report

11.2 RMF Monitoring

From a DB2 point of view, I/O flows between disk storage and buffer pools. I/O activity can be modeled as a three-step process involving pathing from hosts to storage servers, caching of storage server, and internal activity between cache and physical disk storage. Therefore, to monitor I/O one must ask the following questions:

- Which paths do I/Os use, and what is their workload?
- How efficient are the storage server cache algorithms for this workload?
- What is the performance, service time, and/or response time, offered by the storage server?

Most of the RMF reports are issued either at the central processor complex (CPC) level for a global view, or at the logical partition (LPAR) level for each MVS image view.

The following two processes can be used to extract relevant data from the various RMF reports:

1. Determine which fields, from which reports, are useful for a DB2 performance analyst. Most of this data is also required by IBM storage specialists for disk evaluation.
2. Determine how to aggregate and handle the raw data to get *resource level* occupancy information.

11.2.1 RMF Report Analysis

The data required for analyzing DB2 disk activity comes from four RMF reports which are produced by RMF Monitor I or Monitor III reporting (see Chapter 5, "Long-Term Overview Reporting with the Postprocessor" in the *OS/390 RMF Report Analysis*, SC28-1950). The two main RMF reports are cache and device:

- Cache Subsystem Activity - Cache Reports

The cache reports provide cache statistics on an LCU level referring to the LCU by its subsystem identifier (SSID). The accounting methodology for the number of I/Os is different from the methodology used for the device report. So, use a percentage instead of a value when you want to establish correlations between the two reports.

- Direct Access Device Activity - Device Report

The device report triggers path analysis through I/O queuing and provides information for all disk devices per LCU (referred by LCU number). The easiest way to associate this device information with complementary information in the cache reports is to use *device number ranges*. The device report is also an anchor point for measuring path activity.

To obtain path activity information, you can use use:

1. I/O Queuing Activity - IOQ Report

The IOQ report determines, for each LCU, the channel path identifiers (IDs) used. It also allows analysis of potential pending time issues due to channel subsystem.

2. Channel Path Activity - CHAN Report

The CHAN report gives channel link effective activity.

11.2.1.1 Cache Subsystem Activity Reports

RMF Monitor I gathers data for cache subsystem activity reports in SMF record type 74.5 as a default option. To produce the reports, specify:

```
REPORTS ( CACHE ( SUBSYS ) )
```

Note: The SUMMARY and DEVICE options must not be used.

There are three Cache Subsystem Activity reports:

- *Cache Subsystem Status*

This report gives the amount of cache storage and nonvolatile storage (NVS) installed, as well as the current status of the cache.

- *Cache Subsystem Overview*

This report gives the number of I/O requests sent to the control unit and their resolution in the cache (hits).

- *Cache Subsystem Device Overview*

This report gives, for all online volumes attached to the subsystem, the specific utilization of the cache. It also consolidates this information at the LCU level. This information is often correlated with the LCU view of the DEVICE report.

These statistics reports are *generated by each disk storage server* from IDCAMS LISTDATA command requests to each storage server LCU. Therefore, subsystem identification is by SSID or a control unit identifier (CU-ID). CU-ID is the lowest online device number attached to the LCU. This pinpointing is used to establish correlations between cache and device activity reports. Moreover, all reported values are *consolidated data* from all LPARs sharing the LCU. Figure 48 on page 127 shows cache subsystem status and overview reports, and Figure 49 on page 127 shows the cache subsystem device overview report.

Cache Subsystem Status

CACHING must be ACTIVE.

Cache Subsystem Overview

This report consists of consolidated sets of information:

- READ I/O REQUESTS, WRITE I/O REQUESTS, and CACHE MISSES have roughly the same structure. SEQUENTIAL lines report a workload measurement of asynchronous activities. NORMAL lines report synchronous activities.
- Under MISC, the DFW BYPASS field reports NVS overuse. *ASync (TRKS) displays the data flow between cache and physical disks.* A high value of ASync I/Os with a BYPASS=0 is an indicator of a heavy workload, but the NVS buffer is adequate.
- Under NON-CACHE I/O, ICL is at zero when DCME is not used.
- The CKD STATISTICS column reports the existence of old channel programs, which can cause performance degradation if they are still used. Some system-related tools can still use those channel programs.
- Under CACHE MISSES there are four sets of data:
 - NORMAL and SEQUENTIAL lines show respectively synchronous and asynchronous I/O misses.
 - The TRACKS and RATE columns display staging activity from physical disks to cache. In particular, the *sequential prefetch* activity is accounted by number of tracks read and by rate of the read at the end of the SEQUENTIAL line.
 - CFW DATA is positive when DFSORT uses cache sortwork files.
 - TOTAL covers read and write columns only.

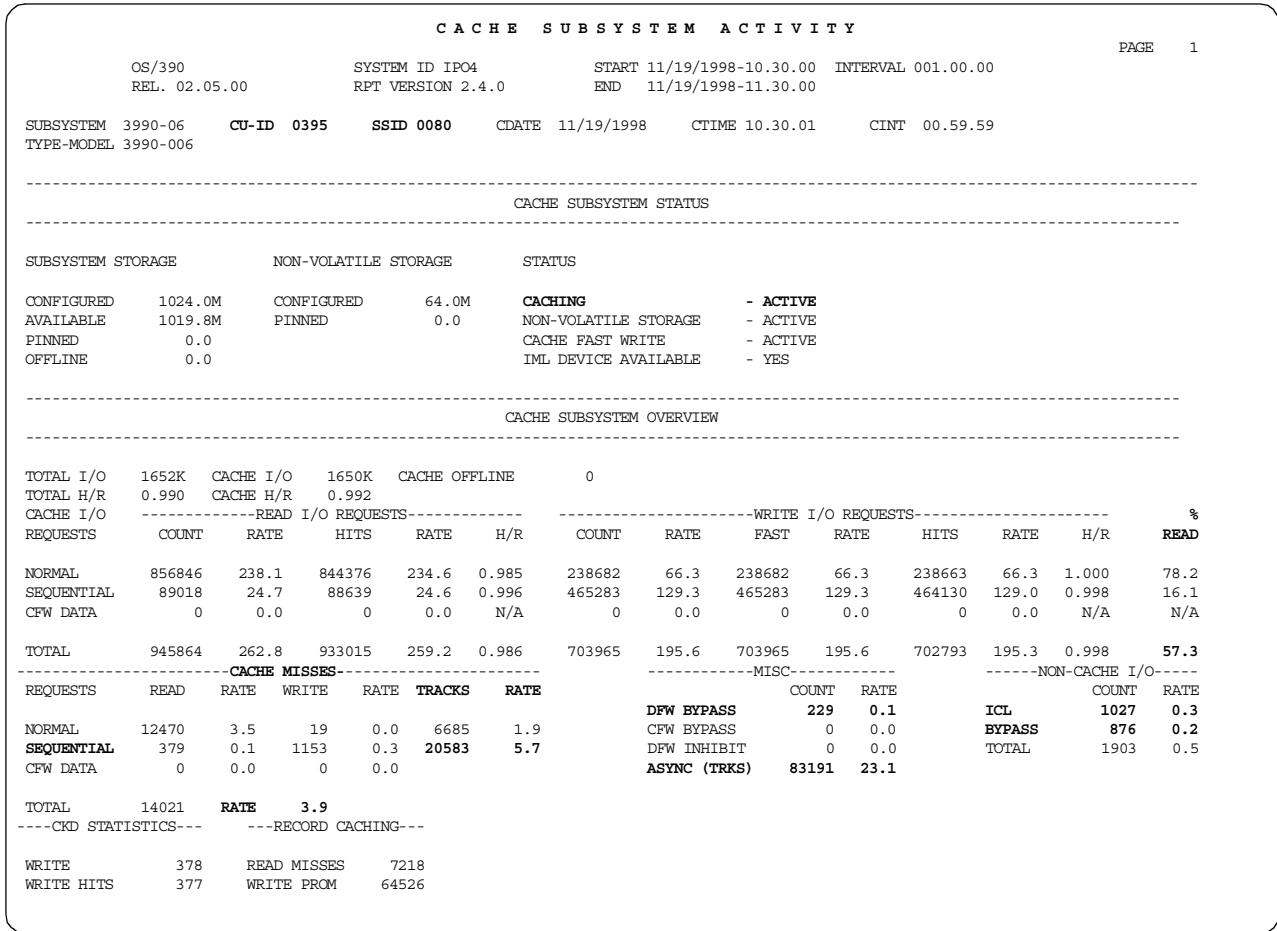


Figure 48. Cache Subsystem Activity Status and Overview Reports

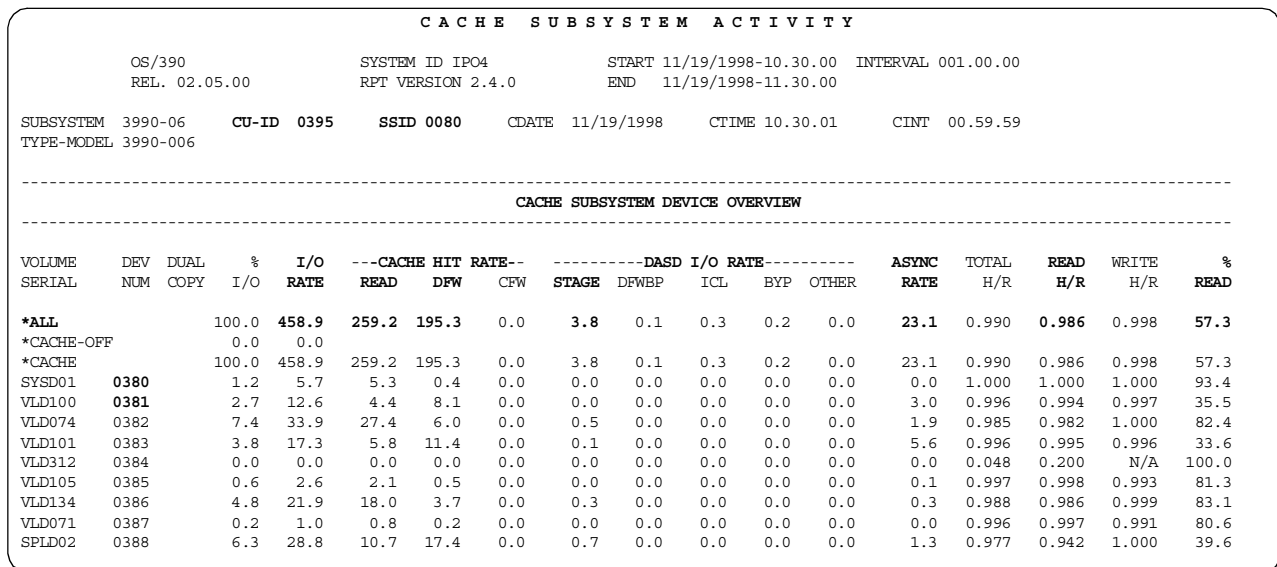


Figure 49. Cache Subsystem Activity Device Overview Report

Cache Subsystem Device Overview

This report lists the devices known by the subsystem at the beginning of the interval. Each line displays statistics for a specific (functional) volume. The I/O rate, divided into two groups (CACHE HIT and DASD I/O), shows the different types of I/O activity in each group. The *ALL line consolidates values at the subsystem level. The fields to review are, in decreasing order of importance:

- I/O RATE, number of I/O requests per second.
- % READ, percentage of read requests compared to all read plus write requests. When combined with DEVICE ACTIVITY RATE of LCU level DEVICE report, this value enables you get to the WRITE ACTIVITY RATE by:
 $(100 - \%READ) * DEVICE ACTIVITY RATE / 100$
- The write activity rate value is an important factor for performance evaluation in a remote copy environment.
- READ H/R, read hit ratio.
- WRITE H/R, write hit ratio.
- DFW, rate of DFW requests.
- STAGE, rate of any (normal or sequential and read or write) I/O requests with cache miss.
- ASYNC RATE, number of tracks asynchronously destaged from cache to disk as a natural consequence of least recently used cache management algorithms.
- ICL, rate of inhibit cache load requests should be at zero. See 9.3.6, “No Caching—Inhibit Cache Load” on page 92.

11.2.1.2 Direct Access Device Activity Report

This report can be produced at either the LCU level (standard) or the Storage Group level for each LPAR. *Both reporting levels should be used.* Although the LCU report is relevant with CACHE reports, Storage Group level reporting automatically consolidates information in a global view of I/O activity consistent with the installation organization defined to SMS. With appropriate SMS definitions, such Storage Group level reporting should map the applications point of view.

To get standard LCU reporting, specify:

```
REPORTS ( DEVICE ( DASD ) )
```

To get Storage Group reporting, specify in the RMF Monitor III postprocessor:

```
REPORTS ( DEVICE ( SG ( storage-group-name ) ) )
```

Figure 50 on page 129 shows the Direct Access Device Activity Report at the Storage Group (SG) level.

The response time of a specific volume in a given LPAR consists of *service time* and volume thread *queuing time*. Service time splits into:

- *Pending time*, which covers the channel subsystem and connections delays to the LCU due to ESCON Director switching, and, in ESCON multiple image facility (EMIF), channel sharing between LPARs

- *Disconnect time*, which covers all internal LCU delays primarily due to a prerequisite process in the storage server; for instance, staging activity for cache misses, or PPRC propagation of the I/O to the secondary site
- *Connect time*, which covers effective data transfer activity

Queuing time is called *I/O supervisor queuing (IOSQ) time* in RMF reports, and covers delays involved by aggregate application interactions on the same volume for that LPAR.

Several LPARs (weighted by activity) sharing the same storage servers must be consolidated to establish value correlations with CACHE reporting.

DIRECT ACCESS DEVICE ACTIVITY																			
OS/390				SYSTEM ID IPO4				START 11/19/1998-10.30.00				INTERVAL 001.00.00							
REL. 02.05.00				RPT VERSION 2.4.0				END 11/19/1998-11.30.00				CYCLE 1.000 SECONDS							
TOTAL SAMPLES = 3,600		IODF = 40		CR-DATE: 11/05/98				CR-TIME: 16.56.05				ACT: POR							
STORAGE GROUP	DEV	DEVICE	VOLUME	LCU	ACTIVITY RATE	AVG	AVG	AVG	AVG	AVG	AVG	AVG	AVG	%	%	%	AVG	%	%
NUM	TYPE	SERIAL			RATE	TIME	TIME	TIME	TIME	TIME	TIME	TIME	TIME	CONN	UTIL	RESV	ALLOC	ALLOC	PEND
SGDB2TS	0103	33903	VLD273	0026	0.053	20	0	0.0	0.0	0.0	0.6	13.0	6.7	0.04	0.10	0.0	64.9	100.0	0.0
SGDB2TS	0105	33903	VLD274	0026	0.102	4	0	0.0	0.0	0.0	0.6	1.8	1.5	0.02	0.03	0.0	22.0	100.0	0.0
SGDB2TS	0107	33903	VLD275	0026	0.706	15	0	0.0	0.0	0.0	0.5	12.3	2.0	0.14	1.01	0.0	31.0	100.0	0.0
SGDB2TS	0109	33903	VLD276	0026	0.224	12	0	0.0	0.0	0.0	0.5	9.8	1.4	0.03	0.25	0.0	22.0	100.0	0.0
SGDB2TS	013A	33903	VLD270	0026	0.003	5	0	0.0	0.0	0.0	0.3	3.6	0.7	0.00	0.00	0.0	0.0	100.0	0.0
SGDB2TS	013B	33903	VLD301	0026	0.109	8	0	0.0	0.0	0.0	0.6	6.2	1.4	0.02	0.08	0.0	62.0	100.0	0.0
SGDB2TS	0384	33903	VLD312	0031	0.008	9	0	0.0	0.0	0.0	0.7	5.1	3.6	0.00	0.01	0.0	17.0	100.0	0.0
SGDB2TS	0396	33903	VLD286	0031	0.036	7	0	0.0	0.0	0.0	0.9	5.1	1.3	0.00	0.02	0.0	60.0	100.0	0.0
SGDB2TS	03BF	33903	VLD219	0031	0.065	21	9	0.0	0.0	0.1	0.9	6.4	4.9	0.03	0.07	0.0	29.0	100.0	0.0
SGDB2TS	0D0B	33903	VLD265	0063	0.131	2	0	0.0	0.0	0.0	0.4	0.6	1.5	0.02	0.03	0.0	14.0	100.0	0.0
SGDB2TS	0D13	33903	VLD285	0063	0.018	4	0	0.0	0.0	0.0	0.4	2.9	1.0	0.00	0.01	0.0	20.7	100.0	0.0
SGDB2TS	0D14	33903	VLD307	0063	0.062	2	0	0.0	0.0	0.0	0.4	0.5	1.0	0.01	0.01	0.0	20.9	100.0	0.0
SGDB2TS	0D22	33903	VLD310	0063	0.011	6	0	0.0	0.0	0.0	0.5	4.4	1.1	0.00	0.01	0.0	32.0	100.0	0.0
SGDB2TS	4066	33903	VLD538	0076	0.675	20	4	0.0	0.0	0.0	0.5	11.4	4.1	0.28	1.05	0.0	103	100.0	0.0
SGDB2TS	4067	33903	VLD539	0076	3.926	31	15	0.0	0.0	0.0	0.3	8.9	6.5	2.54	6.05	0.0	107	100.0	0.0
SGDB2TS	4068	33903	VLD540	0076	2.526	20	5	0.0	0.0	0.0	0.3	5.4	9.5	2.40	3.75	0.0	120	100.0	0.0
SGDB2TS	407F	33903	VLD563	0076	1.076	12	2	0.0	0.0	0.0	0.4	7.2	2.5	0.27	1.05	0.0	112	100.0	0.0
SGDB2TS	4080	3390	VLD564	0077	1.276	13	3	0.0	0.0	0.0	0.4	7.1	2.4	0.31	1.21	0.0	87.7	100.0	0.0
SGDB2TS	4081	3390	VLD565	0077	1.456	16	2	0.0	0.0	0.0	0.4	7.1	5.8	0.84	1.87	0.0	90.9	100.0	0.0
SGDB2TS	4095	3390	VLD585	0077	0.179	15	0	0.0	0.0	0.0	0.9	10.4	3.9	0.07	0.26	0.0	54.0	100.0	0.0
SGDB2TS	4096	3390	VLD586	0077	1.797	12	1	0.0	0.0	0.0	0.3	6.9	3.8	0.68	1.93	0.0	119	100.0	0.0
SGDB2TS		SG			107.550	17	5	0.0	0.0	0.0	0.4	7.4	3.9	0.43	1.27	0.0	6879	100.0	0.0

Figure 50. Direct Access Device Activity Report

The Direct Access Device Activity report provides detailed information for each (functional) volume and consolidates the information at the LCU (or Storage Group when required) level.

The fields to review are:

- LCU, reference number used to locate which CHAN report data to analyze
- DEVICE ACTIVITY RATE, rate per second at which start subchannel (SSCH) instructions to the device completed successfully
- AVG RESP TIME, response time in milliseconds
- AVG IOSQ TIME, queuing time in IOSQ on the device
- AVG PEND TIME, pending time
- AVG DISC TIME, disconnect time

- AVG CONN TIME, connect time mainly for data transfer. To estimate path percentage utilization demand, calculate:

$$\text{AVG CONN TIME} * \text{DEVICE ACTIVITY RATE}/1000 * 100$$

- As an example, an average connect of 4.5 ms with 1200 I/O/sec gives 540%, which means a minimum of six paths is required for this workload level. Checking the channel path activity reports for the different LPARS sharing this LCU enables you to determine the balance of the activity demand (540%) over the current defined path configuration. Intermediate consolidations may be required in complex configurations.
- % DEV UTIL, percentage of device utilization shows the percentage of times RMF has found this device busy. This is a good indicator of demand contention on this volume.

I/O Queuing Activity Report

The I/O Queuing Activity report (see Figure 51 on page 131) is used to analyze the pathing behavior. To get this report, specify:

REPORTS (IOQ)

Use only two fields, LCU and CHAN PATHS, which list the physical paths to review later in the channel path activity report.

Channel Path Activity Report

The Channel Path Activity report (Figure 52 on page 132) identifies performance contentions associated with the channel paths. To produce this report, specify:

REPORTS (CHAN)

Review the following fields:

- CHANNEL ID is the hexadecimal number of the channel path identifier (CHPID).
- PATH SHR; a value of Y indicates that the ESCON channel link (physical channel) is shared between one or more LPARs.
- PARTITION UTILIZATION (%) is the percentage of physical channel path utilization by the LPAR.
- TOTAL UTILIZATION (%) is the percentage of physical channel path utilization that all LPARS of this CPC use. This is the aggregate view of channel utilization.

I/O QUEUING ACTIVITY									
OS/390		SYSTEM ID IPO4			START 11/19/1998-10.30.00 INTERVAL				
001.00.00	REL. 02.05.00	RPT VERSION 2.4.0			END 11/19/1998-11.30.00 CYCLE 1.000				
SECONDS									
TOTAL SAMPLES = 3600 IOP ACTIVITY RATE AVG Q LENGH IODF = 40 CR-DATE: 11/05/98 CR-TIME:									
16.56.05 ACT: POR									
		00	1615.962	0.07					
-									
LCU	CONTENTION RATE	DELAY Q LENGH	% ALL CH PATH BUSY	CONTROL UNITS	CHAN PATHS	CHPID TAKEN	% DP BUSY	% CU BUSY	
0026	0.534	0.39	0.21	0026	A6	82.864	0.00	9.97	
				0027	A2	84.151	0.00	9.49	
					DA	81.694	0.00	10.65	
					E9	83.505	0.00	9.89	
0031	0.617	0.57	2.06	002E	A5	117.30	0.00	13.01	
					9F	110.59	0.00	14.97	
				002F	EA	106.98	0.00	13.26	
					B5	113.08	0.00	13.37	
0063	0.106	0.02	0.01	0034	EA	37.748	0.00	1.98	
					EB	60.421	0.00	1.24	
				0035	B4	57.141	0.00	5.16	
					9E	59.232	0.00	3.06	
0071	2.206	0.02	0.05	0030	E4	52.818	9.33	3.80	
					B2	17.382	42.67	0.58	
					82	16.741	45.13	0.60	
					98	15.184	50.52	0.52	
					9C	16.727	44.92	0.53	
0076	0.000	0.00	0.00	0041	BB	13.857	0.20	0.45	
					BA	13.856	0.26	0.58	

Figure 51. I/O Queuing Activity Report

CHANNEL PATH ACTIVITY														PAGE
OS/390		SYSTEM ID IPO4				START 11/19/1998-10.30.00		INTERVAL 001.00.00				PAGE		1
REL. 02.05.00		RPT VERSION 2.4.0				END 11/19/1998-11.30.00		CYCLE 1.000 SECONDS						
IODF = 40		CR-DATE: 11/05/98		CR-TIME: 16.56.05		ACT: POR		MODE: LPAR		CPMF: AVAILABLE				
CHANNEL PATH		UTILIZATION(%)			CHANNEL PATH		UTILIZATION(%)			CHANNEL PATH		UTILIZATION(%)		
ID	TYPE	SHR	PARTITION	TOTAL	ID	TYPE	SHR	PARTITION	TOTAL	ID	TYPE	SHR	PARTITION	TOTAL
08	OS	Y	0.00	0.00	19	CN D	Y	0.16	3.43	85	IS		OFFLINE	
0C	IS		OFFLINE		1A	CN D	Y	0.00	0.00	88	BY			0.00
0D	IS		OFFLINE		1B	CN D	Y	1.28	1.73	89	BL		OFFLINE	
14	CN D	Y	8.28	8.62	80	CC	Y	0.52	1.48	8A	BL		0.00	0.00
15	CN D	Y	7.84	8.58	81	CN	Y	0.33	0.33	8C	BL		0.00	0.00
16	CN D	Y	8.84	9.22	82	CN D	Y	9.11	15.32	8D	BL		0.17	0.17
17	CN D	Y	8.28	8.51	83	CN D	Y	0.00	0.00	8E	BL		0.00	0.00
18	CN D	Y	0.00	0.00	84	IS		OFFLINE		90	BL		0.14	0.14
91	BL		OFFLINE		9B	CN D	Y	0.00	0.00	A3	CN D	Y	0.78	1.07
92	BL		0.00	0.00	9C	CN D	Y	9.15	15.25	A4	CN D	Y	1.29	1.58
94	BL		0.00	0.00	9D	CN	Y	0.01	0.01	A5	CN D	Y	36.18	36.93
95	BL		OFFLINE		9E	CN D	Y	11.73	12.30	A6	CN D	Y	21.39	21.50
96	BL		OFFLINE		9F	CN D	Y	35.47	35.74	A7	CN D	Y	1.32	1.71
98	CN D	Y	8.43	14.62	A0	CN	Y	0.00	0.00	B0	CN D	Y	0.00	0.00
99	CN D	Y	0.00	0.00	A1	CN D	Y	0.00	0.00	B1	CN D	Y	0.00	0.00
9A	CN D	Y	0.00	0.00	A2	CN D	Y	20.98	21.06	B2	CN D	Y	9.41	15.65
B3	CN D	Y	0.03	0.04	BB	CN D	Y	7.79	7.70	DC	IS		OFFLINE	
B4	CN D	Y	11.62	11.63	BC	BL		OFFLINE		DD	IS		OFFLINE	
B5	CN D	Y	36.13	36.74	BD	BL		OFFLINE		E0	CV		OFFLINE	
B6	CN D	Y	0.00	0.00	BE			OFFLINE		E1	CV		0.02	0.02
B7	CN D	Y	0.80	1.05	D8	CN	Y	0.74	1.37	E2	CN D	Y	0.13	0.16
B8	CN D	Y	8.02	7.67	D9	CC	Y	0.36	0.41	E3	CN D	Y	0.00	0.00
B9	CN D	Y	0.78	1.11	DA	CN D	Y	21.31	21.70	E4	CN D	Y	10.27	10.32
BA	CN D	Y	7.68	7.89	DB	CN D	Y	0.00	0.00	E5	CN D	Y	2.33	5.74
E6	CN D	Y	0.15	0.20										
E7	CN D	Y	0.79	1.14										
E8	CN	Y	0.00	0.00										
E9	CN D	Y	21.01	21.62										
EA	CN D	Y	41.52	42.42										
EB	CN D	Y	12.38	12.37										

Figure 52. Channel Path Activity Report: LPAR Mode

11.2.2 Using RMF Reports

In a performance monitoring context, there are several heterogeneous considerations to be aware of before using RMF reporting. When doing a resource level analysis, what is the reason for using SMs for DB2, and why get reports at the Storage Group level? One may also search for more in-depth analysis tools than RMF. Moreover, as RMF produces a very large number of pages in its reports, spreadsheet automatic data capture and manipulation may be viewed with interest. Finally, the common approach of DB2 and storage specialists on the same performance study cases requires definition of how different tools report the same DB2 I/O.

11.2.2.1 Resource Level Analysis

Some RMF reports must be consolidated to get a global view of contention at the resource level.

For channel utilization, the relevant information is TOTAL UTILIZATION %, which consolidates any EMIF channel utilization on one CPC. When several CPCs share an LCU, pathing analysis must be extended to include all LPAR activities to this LCU over all CPCs.

Cache utilization reports consolidate all different LPAR activities for each LCU. So, to get the whole storage server view, a manual consolidation of LCUs is mandatory. For device activity, RMF data capture is done at the LCU level, so two levels of weighted consolidation are required: first, between all LPARs, to get the

same view as cache reports for each LCU: and second, between all LCUs to get the whole storage server view. Some tools, such as IXFP, offer consolidated data. In the case study activities, there is only one active LPAR, so only LCU level consolidation is done.

11.2.2.2 RMF Reporting at Storage Group Level

The RMF DEVICE report, when edited at the Storage Group level, shows the Storage Group's overall performance from which it is easy to deduce required parallelism. This example focuses only on required fields from DEVICE report at the Storage Group level for a query-intensive activity:

- DEVICE ACTIVITY RATE: 1,042.261
- AVG CONN TIME: 22.3

The required path occupancy (see 11.2.1.2, "Direct Access Device Activity Report" on page 128) for this workload is:

$$\left((1,042.261 \times 22.3) / 1000 \right) \times 100 = 2,324 \%$$

Therefore, there is a minimal path demand of 24 (23+1). It is wise to allocate such a workload over at least 32 paths, so this Storage Group should be spread over four RVAs with a multiple of eight volumes on each. When there are performance issues, and consolidated channel path performance data shows normal values, as the origin of those is in throughput demand flow (MB/sec), check for high disconnect and/or pending times.

11.2.2.3 Tools Providing More In-Depth Analysis than RMF

When RMF reports show some performance issues that require more in-depth analysis, a generalized trace facility (GTF) of channel command words (CCWs) should be used. Please refer to *OS/390 V2 R6.0 MVS Diagnosis: Tools and Service Aids*, SY28-1085, on how to customize the CCW trace. This trace is time-stamped, so storage specialists can control the channel programs issued and analyze their behavior. Some capacity planning information can only be derived at the trace level, in particular bandwidth information, such as number of MB/sec in read and/or in write activities. Such information requires knowledge of the data transmitted by each command. However, for the RVA, an IXFP report edits the global bandwidth, with reads and writes mixed. IXFP calls some internal RVA facilities that dynamically maintain these statistics.

11.2.2.4 Spreadsheet Tools for RMF Analysis

Two tools, RMF spreadsheet converter (RMF2SC) and RMF spreadsheet reporter (RMFPP), allow automatic data capture from standard RMF Monitor III printouts to most common spreadsheet tools. For RMFPP, the printouts should have been previously saved in EBCDIC format (preferably in fixed mode to allow high quality transmission) in the host before they are downloaded to a PC. These RMF tools are described in Part 6 of the *OS/390 V2 R6.0 RMF User's Guide*, SC28-1949.

RMF2SC takes output from RMF and converts it to spreadsheet formats. Working with RMF spreadsheets involves three steps:

1. Using RMF to generate the appropriate reports. The result can be in a data set, which you can download to the PC and process as a host data set or on the screen.

2. Starting RMF2SC on the PC, using appropriate options to select the reports to be converted.
3. Using your spreadsheet program to manipulate the spreadsheet data. Details of how to do this depend on which program you are using, but in all cases, the cells and ranges that you can reference are as described in the *OS/390 RMF Report Analysis*, SC28-1950.

RMF2SC is installed on the host along with the rest of the MVS components of RMF. The deliverable includes the RMF2SC program, sample RMF report files, macros, and converted RMF spreadsheets. The code of RMF2SC (in the self-extracting ZIP file ERB9R2S.EXE) is distributed as member ERB9R2S of the SERBPWS distribution library.

RMFPP allows you to convert RMF data to spreadsheet format and provides a practical approach to using spreadsheet macros for converted reports and overview records. The RMFPP is an extension of RMF2SC and enhances its capability and flexibility. RMFPP also extends the capability of the RMF Postprocessor by converting RMF report data to spreadsheet format. In addition, the function provides a set of sample spreadsheet macros, which you can use to process the converted RMF reports, and as a base for your own spreadsheet macros. The spreadsheet macros contained in the RMFPP are samples to demonstrate how you can use spreadsheets to process RMF data. Device trend and cache statistics macros are a good base for I/O monitoring from RMF.

Monitor the RMF home page on the Internet to find information about RMFPP. Go to the Tools page of this site:

<http://www.s390.ibm.com/rmf>

RMFPP currently supports Lotus 1-2-3 Version 5 (International English), Microsoft Excel Version 5 and Version 7, and Microsoft Excel 97.

11.2.2.5 Global View of a DB2 I/O by DB2 PM and RMF

Each time an application reads or writes data, DB2 requires or updates pages in buffer pools. DB2, synchronously or asynchronously, issues I/O requests, which can trigger synchronous or asynchronous staging and destaging operations between cache and physical disks. Figure 53 on page 135 displays the relationships between DB2 PM and RMF views of an I/O request from or to disk.

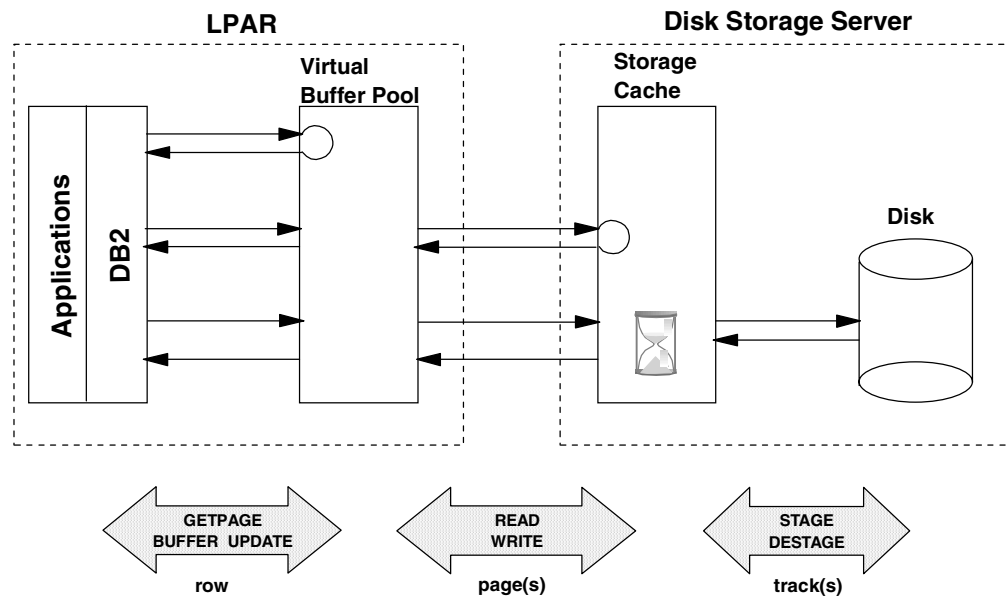


Figure 53. DB2 I/O

11.3 IXFP Monitoring

The IBM Extended Facilities Product (IXFP) is a host software that helps to manage the RVA. IXFP provides an additional level of cache control for RVA beyond that provided by the local operator panel. In addition, IXFP maximizes the benefits of RVA's virtual storage architecture by allowing interactive control of activities such as subsystem administration and reporting.

The IXFP facilities are described in more detail in the *RAMAC Virtual Array Storage Introduction*, GC26-7168, and *IXFP Configuration and Administration*, SC26-7178, manuals.

IXFP subsystem configuration facilities enable the user to control RVA's functional device configuration, as it appears to the host. In addition, the user can control subsystem storage director and channel path parameters or perform physical device operations such as forming arrays or draining devices. IXFP also reports on the physical and functional configuration of an RVA subsystem.

Besides its system reporting feature, IXFP provides extended operator control facilities and schedules the execution of deleted data space release (DDSR). IXFP allows the user to control the time and frequency of DDSR's execution to minimize possible interference with normal operations.

IXFP can continuously monitor and report on subsystem performance and capacity load statistics. It can provide detailed monitoring data for a subsystem. The data can then be used to understand the operation of the subsystem and to optimize RVA's performance and capacity utilization.

The IXFP Reporter facility component reports on functional device and cache performance and on space utilization. Several predefined reports and graphs of this data are available. The Reporter facility also provides the user with subsystem data in a flat file or in SMF, which can be manipulated by the user's

own report writer and graphics display tools. Refer to Chapter 10 of *IXFP Subsystem Reporting*, SC26-7184, as a reference manual for any RVA monitoring and reporting facilities. Standard IXFP reports require use of a SAS statistical environment from SAS Institute, Incorporated. IBM Storage Division specialists can also provide REXX programs with some basic reporting facilities.

There may be some differences between the data produced by IXFP reports and the data produced in the IDCAMS LISTDATA output that RMF uses to edit its CACHE reports. For compatibility reasons, the RAMAC Virtual Array counts certain I/Os as noncached in response to LISTDATA. Therefore, the IXFP reports reflect actual RVA performance more accurately.

IXFP edits three standard reports that consolidate activities at two levels: on all LPARs sharing the RVA, and on all four LCUs contained in the RVA. The standard reports are:

- Device Performance
- Cache Effectiveness
- Space Utilization

The Space Utilization report provides information related to LSF management of physical disk space. The Device Performance and Cache Effectiveness reports offer a complementary view of RMF and can be used to cross-check RMF information. From a DB2 application point of view, the system summary information indicates what to look for. Other detailed information either at the functional volume level or at RVA hardware subcomponent level are beyond the scope of this chapter.

11.3.1 Device Performance Reports

Figure 54 on page 137 shows the subsystem summary of the device performance report. Four fields of this report complement the RMF view. Other information for storage specialists is related to either functional volume statistics, disk array summaries, channel interface performance, or distribution of physical drive module utilization.

The fields to review are:

- I/O PER SEC is the average number of I/O operations per second for the subsystem.
- KBYTES PER SEC is the amount of data in kilobytes transferred per second between the host and the subsystem.
- I/O SERVICE TIME components show the average service time per I/O operation in milliseconds. This service time does not include host and channel subsystem queuing times (IOSQ and Pend time) as the RMF Device Activity report shows.
- DISC is the average time the subsystem was disconnected from the channel (in milliseconds) while processing an I/O operation.
- CONNECT is the average time the subsystem was connected to the channel (in milliseconds) while processing an I/O operation. This includes data transfer time and command parameter transfer time.

- WRITE PER SEC is the average number of write operations per second for the subsystem.
- I/O PER SEC is the average number of I/O operations per second for the subsystem. This field may not be equal to the sum of READ PER SEC and WRITE PER SEC, either because it includes other I/O operations, such as sense commands, or because there may be more than one read or write operation per channel program. Accounting of I/O per second is based on number of Locate Record CCWs met in the channel programs.
- READ HIT% is the percentage of read operations for which the referred track was present in cache storage.
- WRITE HIT % is the percentage of write operations for which the referred track was present in cache storage.
- STAGE PER SEC is the number of transfers of tracks of data from DASD storage to cache storage per second.
- HITS / STGE is the ratio of cache hits (in number of I/Os) to cache misses (in number of staged tracks).

XSA/REPORTER		IBM ITSO POKEEPSIE		17:30 Thursday, February 18,											
CACHE EFFECTIVENESS OVERALL SUMMARY				18FEB1999 17:32:04											
				SIBCEIO V2 R1 L1											
REPORT START DATE: 17FEB1999		REPORT END DATE: 17FEB1999													
REPORT START TIME: 15:16		REPORT END TIME: 15:54													
SUBSYSTEM NAME: 20395		(CACHE SIZE: 1024 MB		NVS SIZE: 8 MB)											
FDID	DEV	VOLSER	T/P	READ PER SEC	WRITE PER SEC	I/O PER SEC	READ RATIO	READ HIT %	WRITE HIT %	I/O HIT %	DFW CONSTR	STAGE PER SEC	HITS/STGE	LOW REF CT	
0000	2B00	RV2B00	P	4.9	0.0	4.2	11077	100.0	100.0	100.0	0.00	11.2	0.4	76.4	
0001	2B01	RV2B01	P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	
0002	2B02	RV2B02	P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	
00FE	2BFE	RV2BFE	P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	
00FF	2BFF	RV2BFF	P	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	
SUBSYSTEM SUMMARY				READ PER SEC	WRITE PER SEC	I/O PER SEC	READ RATIO	READ HIT %	WRITE HIT %	I/O HIT %	DFW CONSTR	STAGE PER SEC	HITS/STGE	LOW REF CT	TRACK OCCUP
PROD PARTITION				53.8	0.0	45.7	61329	99.3	100.0	99.3	0.0	113.2	0.5	73.7	
OVERALL TOTALS				53.8	0.0	45.7	61329	99.3	100.0	99.3	0.0	113.2	0.5	73.7	25050

Figure 55. IXFP Cache Effectiveness Subsystem Summary Report

11.3.3 Space Utilization Report

Figure 56 on page 139 shows the subsystem space utilization summary report. This information is mainly storage business oriented, but high net capacity load (NCL) ratios may trigger more intensive RVA background garbage collection processes, which can impact overall performance. Moreover, low compression ratios involve higher activity between cache and disk. An RVA is considered as balanced with a 3.6 compression ratio and with an NCL lower than 75%.

IBM ITSO POKEEPSIE										16:47 Wednesday, February 17, 1999					
XSA/REPORTER										SPACE UTILIZATION SUMMARY REPORT					
SUBSYSTEM 20395										17FEB1999 16:47:05					
(NUMBER OF FUNCTIONAL DEVICES: 256)										SIBSPUT V2 R1 L1					
0 FUNCTIONAL CAPACITY (MB) -- % FUNCTIONAL CAPACITY															
FDID	DEV	VOLSER	T/P	DEVICE	FUNCT	ALLOC	STORED	NOT	ALLOC	STORED	NOT	-- PHYSICAL CAP USED (MB) --			COMP
ADDR				TYPE	CAP (MB)			STORED			STORED	SHARED	UNIQUE	TOTAL	RATIO
0000	2B00	RV2B00	P	33903	2838.0	1708.3	1568.2	1269.8	60.2	55.3	44.7	0.0	950.0	950.0	1.7
0001	2B01	RV2B01	P	33903	2838.0	2552.3	2552.3	285.7	89.9	89.9	10.1	0.0	825.8	825.8	3.1
00FC	2BFC	RV2BFC	P	33903	2838.0	44.1	28.7	2809.3	1.6	1.0	99.0	0.0	8.6	8.6	3.3
00FD	2BFD	RV2BFD	P	33903	2838.0	852.4	177.6	2660.4	30.0	6.3	93.7	0.0	34.1	34.1	5.2
00FE	2BFE	RV2BFE	P	33903	2838.0	1296.9	910.5	1927.5	45.7	32.1	67.9	0.0	264.1	264.1	3.4
00FF	2BFF	RV2BFF	P	33903	2838.0	329.7	121.0	2717.0	11.6	4.3	95.7	0.0	49.5	49.5	2.4
SELECTED DEVICES SUMMARY															
				FUNCTIONAL CAPACITY (MB)				% FUNCT CAPACITY				DISK ARRAY			COMP
SELECTED DEVICES				TOTAL FUNCTIONAL	NOT		NOT		-- PHYSICAL CAP USED (MB) --			COMP			
				CAPACITY (MB)	STORED	STORED	STORED	STORED	SHARED	UNIQUE	TOTAL	RATIO			
PRODUCTION PARTITION:				256	726532.2	204036.4	522495.8	28.1	71.9	0.0	65964.1	65964.1	3.1		
TOTALS:				256	726532.2	204036.4	522495.8	28.1	71.9	0.0	65964.1	65964.1	3.1		
SUBSYSTEM 20395															
SPACE UTILIZATION SUMMARY															
NUMBER OF				DISK ARRAY			NET CAPACITY LOAD (%)			COLL FREE SPACE (%)			UNCOLL FREE SPACE (%)		
FUNCTIONAL DEVICES				CAPACITY (MB)			TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
-----				-----			-----	-----	-----	-----	-----	-----	-----	-----	-----
256				117880.2			0.0	56.4	56.4	0.0	42.4	42.4	0.0	1.3	1.3

Figure 56. IXFP Space Utilization Subsystem Report

The fields to review are as follows:

- NET CAPACITY LOAD (%) PROD is the percentage of back-end physical capacity that is used (not free) in the subsystem. This includes user data and the system areas needed to maintain the arrays. NCL does not include data in the cache until the data is written to the back-end disk storage.
- COMP RATIO is the approximate ratio of functional capacity stored to the physical capacity used (on a subsystem level).

Chapter 12. Case Study

The case study applies all previously described monitoring facilities in a common project from the DB2 and storage perspectives. This approach introduces some redundancy. The good news is that redundancy allows cross-checking information among various sources.

The environment is a very large DB2 query on partitioned table spaces over two RVA storage servers. Activity is exclusively read oriented. Only one DB2 LPAR accesses the data; there is no data sharing.

Reports generated by DB2 PM, RMF, and IXFP have been pruned to extract the relevant data and to focus on overall activity only. The reports are shown in the appendixes.

12.1 DB2 Case Study Analysis

From the DB2 point of view, the first step in analysis is to examine the accounting reports generated by DB2 PM to establish the elapsed and CPU times of the case study. The complete DB2 PM reports are shown in Appendix C, "DB2 PM Accounting Trace Report" on page 201 and Appendix D, "DB2 PM Statistics Report" on page 205.

12.1.1 General Analysis

12.1.1.1 Elapsed and CPU Time

For any application, the first analysis is the elapsed time and the CPU time. This information is obtained from the class 1 and class 2 times of the accounting report. This is shown in Figure 57 on page 142.

Line **A** shows the elapsed time of the application is 37 minutes and 41 seconds. The CPU time is 1 hour, 6 minutes and 21.58 seconds (**B**). The CPU time is much higher than the elapsed time, because multiple CPUs are being used in parallel. This is shown in the breakup of the CPU time, in TCB time (**C**), stored procedure time (TCB-STPROC) and parallel CPU time (PAR.TASKS, **D**).

12.1.1.2 SQL Statements

The CPU and elapsed values indicate a heavy process, either a batch calculation affecting many rows, or a CPU bound query. Figure 58 on page 142 helps to establish this. The number of SQL calls is small; it shows one dynamic SQL statement (**J** in Figure 58) which returns 100 rows (**K** in Figure 58). An extra FETCH is required to establish that there are no more rows. This example is a complex query.

TIMES/EVENTS	APPL (CLASS 1)	DB2 (CLASS 2)	
ELAPSED TIME	37:41.001054	37:40.386069	A
CPU TIME	1:06:21.580844	1:06:21.549125	B
TCB	14:02.087183	14:02.055513	C
TCB-STPROC	0.000000	0.000000	
PAR.TASKS	52:19.493661	52:19.493612	D
SUSPEND TIME	N/A	30:55.990291	E
TCB	N/A	3:48.273531	F
PAR.TASKS	N/A	27:07.716760	G
NOT ACCOUNT.	N/A	19:50.057025	H
DB2 ENT/EXIT	N/A	217	
EN/EX-STPROC	N/A	0	
DCAPT.DESCR.	N/A	N/A	
LOG EXTRACT.	N/A	N/A	

Figure 57. DB2 PM Accounting, Class 1 and Class 2 Sections

SQL DML	TOTAL	
SELECT	0	
INSERT	0	
UPDATE	0	
DELETE	0	
DESCRIBE	0	
DESC.TBL	0	
PREPARE	1	J
OPEN	1	
FETCH	101	K
CLOSE	1	
DML-ALL	104	

Figure 58. DB2 PM Accounting, SQL DML Section

The DB2 PM accounting report contains two more sections with SQL statements. One of these (SQL DCL) contains interesting information for this study. This section is shown in Figure 59 on page 143:

- L** One SET DEGREE statement
- M** One CONNECT of type 2

The SET DEGREE statement means that the user decided to enable parallelism for this query. This can be confirmed by examining the parallel query section of the DB2 PM accounting report in Figure 60 on page 143. A parallel degree of 5 (**N** in Figure 60) was established. One parallel group executed (**O** in Figure 60) and it executed with a degree of 5 (**P** in Figure 60).

The other DCL statement (**M** in Figure 59 on page 143) is a CONNECT of type 2. This could mean that distributed data was accessed. To confirm this, the DDF requester information in the accounting report is checked, this section is not present in the report. Statistics also collects DDF information, this is shown in Figure 61 on page 144. No information is shown. This means that the trace classes to collect DDF information were not started, or there was no DDF activity.

The explanation can be found in the driver program used to run the query. This program does a CONNECT RESET automatically after each query.

SQL DCL	TOTAL
-----	-----
LOCK TABLE	0
GRANT	0
REVOKE	0
SET SQLID	0
SET H.VAR.	0
SET DEGREE	1 L
SET RULES	0
CONNECT 1	0
CONNECT 2	1 M
SET CONNEC	0
RELEASE	0
CALL	0
ASSOC LOC.	0
ALLOC CUR.	0
DCL-ALL	2

Figure 59. DB2 PM Accounting, SQL DCL Section

QUERY PARALLEL.	TOTAL
-----	-----
MAXIMUM MEMBERS	N/P
MAXIMUM DEGREE	5 N
GROUPS EXECUTED	1 O
RAN AS PLANNED	1 P
RAN REDUCED	0
ONE DB2 COOR=N	0
ONE DB2 ISOLAT	0
SEQ - CURSOR	0
SEQ - NO ESA	0
SEQ - NO BUF	0
SEQ - ENCL.SER	0
MEMB SKIPPED (%)	0
DISABLED BY RLF	NO

Figure 60. DB2 PM Accounting, Parallel Query Section

12.1.1.3 Time Not Accounted

H in Figure 57 on page 142 shows 19 minutes 50 seconds of time not accounted for. This is the time the main TCB had to wait for all the parallel threads to finish.

Without query parallelism, the time not accounted for is defined as the difference between class 2 and class 3 times (class 2 - class 3). This formula works when there is only one TCB.

With query parallelism, this formula no longer works. Class 2 time is associated with the main TCB only, since it represents the DB2 time of a query. However, class 3 time is associated with each parallel task (SRB) plus the main TCB, and the sum of all the class 3 times can be much longer than the class 2 time. As a result, DB2 PM decides to report on the main TCB only. Again, the time not accounted for is still class 2 - class 3, but associated with the main TCB only. Since the main TCB can wait for a long time for all the parallel tasks to complete, as is the case where a query scans a large tablespace, the time not accounted for can be quite long.

GLOBAL DDF ACTIVITY	QUANTITY	/MINUTE	/THREAD	/COMMIT
DBAT QUEUED-MAXIMUM ACTIVE	N/P	N/P	N/P	N/A
CONV.DEALLOC-MAX.CONNECTED	N/P	N/P	N/P	N/A
INACTIVE DBATS - CURRENTLY	N/P	N/A	N/A	N/A
INACTIVE DBATS - HWM	N/P	N/A	N/A	N/A
ACTIVE DBATS - CURRENTLY	N/P	N/A	N/A	N/A
ACTIVE DBATS - HWM	N/P	N/A	N/A	N/A
TOTAL DBATS - HWM	N/P	N/A	N/A	N/A
COLD START CONNECTIONS	N/P	N/P	N/P	N/P
WARM START CONNECTIONS	N/P	N/P	N/P	N/P
RESYNCHRONIZATION ATTEMPTED	N/P	N/P	N/P	N/P
RESYNCHRONIZATION SUCCEEDED	N/P	N/P	N/P	N/P

Figure 61. DB2 PM Statistics, Global DDF Activity Section

12.1.2 Data Access

The next step of this analysis is the examination of the buffer pools to establish the pattern of data access. The accounting report shows that the following buffer pools are being used:

- BP0 System table and index spaces
- BP2 User table spaces
- BP4 User index spaces
- BP5 Work DB table spaces

BP0 contains little data which is all in the buffer pool; this buffer pool contains the DB2 Catalog and Directory. BP5 is another special case; it contains the work DB. Both BP2 and BP4 contain interesting data. The corresponding sections from the accounting report are in Figure 62 on page 144 and in Figure 63 on page 145. The query accesses 5.8 million pages in BP2 (A) and 0.3 million pages in BP4 (F). No data is updated in either buffer pool.

In buffer pool BP2, 18796 pages (B) are read synchronously and 5.8 million pages (E) are read with prefetch operations. The prefetch operations are sequential prefetch (C) and dynamic prefetch (D). The total is 186060 prefetches that read 5795249 pages, an average of 31.15 pages per prefetch.

BP2	TOTAL
BPOOL HIT RATIO (%)	0
GETPAGES	5835348 A
BUFFER UPDATES	0
SYNCHRONOUS WRITE	0
SYNCHRONOUS READ	18796 B
SEQ. PREFETCH REQS	163939 C
LIST PREFETCH REQS	0
DYN. PREFETCH REQS	22121 D
PAGES READ ASYNCHR.	5795249 E

Figure 62. DB2 PM Accounting, BP2 Section

BP4	TOTAL
-----	-----
BPOOL HIT RATIO (%)	50
GETPAGES	300350 F
BUFFER UPDATES	0
SYNCHRONOUS WRITE	0
SYNCHRONOUS READ	754
SEQ. PREFETCH REQS	702
LIST PREFETCH REQS	0
DYN. PREFETCH REQS	3944
PAGES READ ASYNCHR.	148634

Figure 63. DB2 PM Accounting, BP4 Section

12.1.3 Suspend Times

The class 3 section of the accounting report shows suspend (wait) times. This is shown in Figure 64 on page 145. These values are only shown if the accounting trace is started with class 3.

In this example, the wait times reported correspond to I/O. The application waited 56.77 seconds for synchronous reads (**A**) and 29 minutes and 59.16 seconds for prefetch reads (**B**). This time is not directly attributable to the elapsed time, because the I/O was performed in parallel. On Figure 57 on page 142 we have the timing for parallelism. Only 3 minutes and 48.3 seconds (**F** in Figure 57) were in the main task (TCB). The remainder, 27 minutes and 7.7 seconds (**G** in Figure 57) were performed in the five parallel tasks.

CLASS 3 SUSP.	ELAPSED TIME	EVENTS	
-----	-----	-----	
LOCK/LATCH	0.060799	2760	
SYNCHRON. I/O	56.770913	19559	A
OTHER READ I/O	29:59.155952	143669	B
OTHER WRITE I/O	0.000000	0	C
SER.TASK SWITCH	0.002627	2	
ARC.LOG (QUIES)	0.000000	0	
ARC.LOG READ	0.000000	0	
DRAIN LOCK	0.000000	0	
CLAIM RELEASE	0.000000	0	
PAGE LATCH	0.000000	0	
STORED PROC.	0.000000	0	
NOTIFY MSGS	0.000000	0	
GLOBAL CONT.	0.000000	0	
TOTAL CLASS 3	30:55.990291	165990	

Figure 64. DB2 PM Accounting Class 3 Times

12.1.3.1 Synchronous I/O

Line **A** in Figure 64 on page 145 shows 19559 synchronous reads. This value is also reported as the total synchronous reads made by this query, **F** in Figure 66 on page 147. The elapsed time for these reads is 56.770913 seconds. This gives an average of 2.9 milliseconds for each read. This is a good response time for a direct access to a page. This response time is also shown in the DB2 PM highlights section, **D** in Figure 65 on page 146.

```

HIGHLIGHTS
-----
THREAD TYPE   : ALLIED
TERM.CONDITION: NORMAL
INVOKE REASON : DEALLOC
COMMITTS      :          2
ROLLBACK      :          0
INCREM.BINDS  :          0
UPDATE/COMMIT :          0.00
SYNCH I/O AVG.: 0.002903 D
PROGRAMS      :          0
PARALLELISM   : CP

```

Figure 65. DB2 PM Accounting Highlights

12.1.3.2 Asynchronous Read I/O

B in Figure 64 on page 145 shows 143669 asynchronous reads. This corresponds to the sum of all prefetch operations (sequential, dynamic and list prefetch). The suspend time for these reads is 29 minutes and 59.155952 seconds (**B** in Figure 64). The total number of prefetch requests is not necessarily equal to the total number of prefetch I/Os, which is reported at DB2 subsystem level in the statistics record.

J in Figure 66 on page 147 shows that 5.94 million pages were read asynchronously. The suspend time for reading these pages is 29 minutes and 59.155952 seconds (**B** in Figure 64). This means that, on average, the program had to wait 0.3 milliseconds for each page. The low buffer hit ratio (**E** in Figure 66) means that most pages had to be read from disk (or cache).

The total number of prefetch requests made by the query is **G + H + I**. In this example, it is 190714 requests. Of these, 143669 caused a suspension; see **B** in Figure 64. This means that 47045 (190714-143669) prefetch requests did not cause the application program to wait; they have a response time of zero.

The average wait of all prefetch operations is the total wait time (**B** in Figure 64) divided by the total number of prefetch requests (**G + H + I**). This is 9.4 milliseconds.

The number of pages read in each prefetch request can be calculated as the total number of pages read asynchronously (**J** in Figure 66 on page 147) divided by the total number of prefetch requests (**G + H + I**). In this example, this gives 31.2 pages (5943947/190714). This is very close to the maximum (32 pages) for these buffer pools.

12.1.3.3 I/O Rate

The query did 19559 synchronous reads (**F** in Figure 66 on page 147) and 190714 prefetch requests (**G + H + I** in Figure 66). This gives a total of 210273 I/O requests. For an interval of 37 minutes 40.4 seconds (**A** in Figure 57 on page 142) this gives 93.0 I/O per second.

TOT4K	TOTAL	
-----	-----	
BPOOL HIT RATIO (%)	2	E
GETPAGES	6135875	
BUFFER UPDATES	48	
SYNCHRONOUS WRITE	0	
SYNCHRONOUS READ	19559	F
SEQ. PREFETCH REQS	164649	G
LIST PREFETCH REQS	0	H
DYN. PREFETCH REQS	26065	I
PAGES READ ASYNCHR.	5943947	J
HPOOL WRITES	0	
HPOOL WRITES-FAILED	0	
PAGES READ ASYN-HPOOL	0	
HPOOL READS	0	
HPOOL READS-FAILED	0	

Figure 66. DB2 PM Accounting Buffer Pool Summary

12.1.4 Conclusions

This example shows a very complex and heavy read-only query. Five-way parallelism has been used to reduce elapsed time.

The query is CPU bound with complex stage 2 type predicates, which access many pages but produce a small result set of 100 rows.

Sequential prefetch, dynamic prefetch, and sequential reads are executed. I/O response time is excellent.

The query could probably benefit from a higher degree of parallelism or from a faster central processor.

Summary of I/O operations

Sequential prefetch requests:	164649
Dynamic prefetch requests:	26065
Total prefetch requests:	190714
Total synchronous reads:	19559
Total I/O requests:	210273
Average I/O per second:	93

12.2 Storage Server Analysis

From the storage server analysis point of view, the first activity, in such a case study, is to eliminate unnecessary data from large RMF and IXFP reports. After the data has been discarded, you apply the methodology described in Chapter 11.2, “RMF Monitoring” on page 124, and Chapter 11.3, “IXFP Monitoring” on page 135 to access the relevant information.

12.2.1 RMF Views

RMF generates a report set for each LPAR. In each set, the cache subsystem activity report displays the same common view over all sharing LPARs at the LCU level. Therefore, the process of extracting data to analyze is mainly based on

discarding "foreign overhead" from the target study. However, values related to this foreign overhead must be preserved because interactions exist on resource access inside the same computing perimeter. The first step is an overall analysis. Moreover, this first overview allows locating some missing data and finding information from other sources. In this case, some input lacking from the RMF cache analysis reports was compensated by IXFP.

An efficient approach is to build a spreadsheet, doing some consolidation between the cache subsystem activity and device activity reports. In this case study, 46 pairs of LCU level RMF reports were reviewed before selecting 9 of them as relevant. There was no foreign overhead to take into account. Figure 67 on page 148 shows the result of this preliminary step. RMFPP was not used because we just needed to capture global activities at the LCU and at Storage Group levels. The report source line is either "crr" for cache activity, "da" for device activity source, or "crr/da". The fields to review are as follows:

- CUID/ADDR is the CU-ID field of the cache subsystem activity report header cross-checked with the DEV NUM of the direct access device activity report.
- SSID is the SSID field of the cache subsystem activity report header.
- LCU is the LCU field of the direct access device activity report.
- rmf_rate is the LCU total device activity rate from the direct access device activity report.
- crr_rate is the aggregate LCU I/O rate field from the cache subsystem device overview report. These values are higher than the values RMF captured because control units count at the command level (locate record command) rather than at the channel program level for RMF. Moreover, there is a missing value for the two last LCUs because of some operator interaction during the measurement interval. The fact the "crr" and the "rmf" values are close to each other shows there is no data sharing with other LPARs.

In most cases, a preknowledge of the environment shortens this pruning process.

reports field unit	REDUCTION				
	crr/da CUID/ADD	crr SSID	da LCU	da rate ssch /	crr rate sI/O / sec
RVA_1					
1st LCU	2B00	0088	0046	15.8	18.2
2nd LCU	2B40	0089	0047	8.8	10.1
3rd LCU	2B80	008A	0048	14.1	15.9
4th LCU	2BC0	008B	0049	7.9	missing
Tot	rva1			46.6	44.200
RVA_2					
1st LCU	2C00	2007	004A	3.7	4.2
2nd LCU	2C40	2008	004B	14.4	15.9
3rd LCU	2C80	2009	004C	14.5	16.8
4th LCU	2CC0	200A	004D	11.7	missing
Tot	rva2			44.4	36.900
System	71C0	603C	0055	7.0	22.1

Figure 67. Reducing the RMF Data to Analyze

12.2.1.1 Device Activity Report Analysis

The RMF report analysis is based on the LCU level and Storage Group level device activity reports. Figure 68 on page 149 shows the extracted summary lines. The Storage Group level line is the average combination of LCUs 46-48 and 4A-4D activities, as the detail volume level display shows (refer to Figure 49 on page 127 as an example). The other information to review is:

- DEVICE ACTIVITY RATE is spread across both LCUs and RVAs inside the Storage Group, which indicates an allocation controlled environment. The level of activity is 90.959.
- AVG RESP TIME is 31 ms.
- AVG IOSQ TIME is 1 ms, which also leads to a controlled environment.
- AVG PEND TIME is 0.2 ms, not an issue.
- AVG DISC TIME is 7.1 ms, which is quite high but lower than the connect time. This requires further analysis.
- AVG CONN TIME is 22.7 ms, which indicates heavy transfers. This information matches the DB2 PM overview of 32 pages, each consisting of one 4K CI, plus some overhead. The accuracy of this estimate is based on supposed homogeneous allocation parameters in the same Storage Group.

From the CONN time and ACTIVITY rate, the path demand is deduced thus:

$$((90.959 \times 22.7) / 1000) \times 100 = 206.5 \% \text{ of a path demand}$$

This Storage Group level device activity report is only for one LPAR. Several LPAR reports should be consolidated (with the spreadsheets) to get the global Storage Group demand in a data sharing environment. The case study is for only one LPAR. A more complex situation would be several LPARs spread across different CPCs, with EMIF shared channels.

DIRECT ACCESS DEVICE ACTIVITY																			
OS/390				SYSTEM ID QP02				START 02/12/1999-15.05.39				INTERVAL 000.38.01							
REL. 02.06.00				RPT VERSION 2.6.0				END 02/12/1999-15.43.41				CYCLE 1.000 SECONDS							
(RMF report extract)																			
STORAGE GROUP	DEV NUM	DEVICE TYPE	VOLUME SERIAL	LCU	DEVICE ACTIVITY RATE	AVG RESP TIME	AVG IOSQ TIME	AVG DPB DLY	AVG CUB DLY	AVG DB DLY	AVG PEND TIME	AVG DISC TIME	AVG CONN TIME	% DEV CONN	% DEV UTIL	% DEV RESV	AVG NUMBER ALLOC	% ANY ALLOC	% MT PEND
			LCU	0046	15.778	32	0	0.0	0.0	0.0	0.2	7.1	24.5	0.60	0.78	0.0	15.0	100.0	0.0
			LCU	0047	8.752	33	2	0.0	0.0	0.0	0.2	6.7	24.7	0.34	0.43	0.0	14.0	100.0	0.0
			LCU	0048	14.145	27	1	0.0	0.0	0.0	0.2	5.4	20.6	0.45	0.57	0.0	11.0	100.0	0.0
			LCU	0049	7.923	32	0	0.0	0.0	0.0	0.2	7.0	24.7	0.31	0.39	0.0	19.0	100.0	0.0
			LCU	004A	3.721	34	0	0.0	0.0	0.0	0.2	9.3	24.2	0.14	0.20	0.0	3.0	100.0	0.0
			LCU	004B	14.431	26	2	0.0	0.0	0.0	0.2	6.3	17.8	0.40	0.54	0.0	4.0	100.0	0.0
			LCU	004C	14.540	35	0	0.0	0.0	0.0	0.2	9.1	25.1	0.57	0.78	0.0	5.0	100.0	0.0
			LCU	004D	11.668	30	0	0.0	0.0	0.0	0.2	7.6	22.3	0.41	0.55	0.0	7.0	100.0	0.0
			LCU	0055	6.963	3	0	0.0	0.0	0.0	0.4	0.1	2.6	0.06	0.06	0.0	299	100.0	0.0
RVAL		SG			90.959	31	1	0.0	0.0	0.0	0.2	7.1	22.7	0.40	0.53	0.0	78.0	100.0	0.0

Figure 68. Case Study RMF Direct Access Device Activity Report Extract

12.2.1.2 I/O Queuing Activity Report Analysis

Figure 69 on page 150 displays an extract of this report, which enumerates the CHANNEL PATH hexadecimal identifications for each LCU:

1. For LCUs 0046-0049: 07, 08, 12, 8B, 91, 1E, C8, D0
2. For LCUs 004A-004D: 0A, 8C, 95, 15, 8F, C1, D2

This results in a total of 16 different paths. Look at their busy percentage in the channel path activity report.

I/O QUEUING ACTIVITY									
(RMF Report extract for first RVA)									
0046	0.000	0.00	0.00	2B00	07	1.988	0.07	0.15	
					08	1.980	0.11	0.11	
					12	1.985	0.04	0.02	
					8B	1.969	0.09	0.16	
				2B01	91	1.961	0.13	0.24	
					1E	1.967	0.04	0.31	
					C8	1.966	0.07	0.16	
					D0	1.964	0.07	0.24	
(RMF Report extract for second RVA)									
004A	0.000	0.00	0.00	2C00	0A	0.464	0.00	0.19	
					8C	0.456	0.10	0.19	
					95	0.463	0.09	0.19	
					C3	0.466	0.09	0.28	
				2C01	15	0.463	0.19	0.09	
					8F	0.471	0.18	0.46	
					C1	0.471	0.00	0.19	
					D2	0.466	0.09	0.19	

Figure 69. Case Study RMF I/O Queuing Activity Extract

12.2.1.3 Channel Path Activity Report Analysis

Figure 70 on page 150 shows an extract of this report, which establishes:

1. Those paths are not EMIF shared.
2. Their utilization is lower than 14% and well balanced.

So there is no path contention concern.

CHANNEL PATH ACTIVITY																	
CHANNEL PATH			UTILIZATION(%)			CHANNEL PATH			UTILIZATION(%)			CHANNEL PATH			UTILIZATION(%)		
ID	TYPE	SHR	PARTITION	TOTAL	ID	TYPE	SHR	PARTITION	TOTAL	ID	TYPE	SHR	PARTITION	TOTAL			
00	CNC_S			0.00	08	CNC_S			13.87	10	CNC_S			0.00			
01	CNC_S			0.02	09	CNC_S			0.02	11	CNC_S			0.02			
02	CNC_S			0.02	0A	CNC_S			12.30	12	CNC_S			13.88			
03	CNC_S			0.02	0B	CNC_S			0.05	13	CNC_S			0.03			
04	CNC_S			0.02	0C	CNC_S			0.00	14	CNC_S			0.02			
05	CNC_S			0.01	0D			OFFLINE		15	CNC_S			12.44			
06	CNC_S			0.00	0E	CNC_S			0.00	16	CNC_S			0.01			
07	CNC_S			13.75	0F	CNC_S			0.00	17	CNC_S			0.00			
1E	CNC_S			13.66	26	CNC_S			0.03	2E	CNC_S			0.03			
1F	CNC_S			0.03	27	CNC_S			0.02	2F	CNC_S			0.01			
34			OFFLINE		8D	CNC_S			0.02	95	CNC_S			12.24			
80			OFFLINE		8E	CNC_S			0.01	96	CNC_S			0.06			
84	OSA			0.02	8F	CNC_S			12.28	97	CNC_S			0.02			
88	CNC_S			0.00	90	CNC_S			0.03	98	CNC_S			0.03			
89	CNC_S			0.02	91	CNC_S			13.87	99	CNC_S			0.51			
8A	CNC_S			0.02	92	CNC_S			0.01	9A	CNC_S			0.00			
8B	CNC_S			13.63	93	CNC_S			0.02	9B	CNC_S			0.00			
8C	CNC_S			12.31	94	CNC_S			0.02	9C	CNC_S			0.04			
C0	CNC_S			0.02	C8	CNC_S			13.83	D0	CNC_S			13.63			
C1	CNC_S			12.34	C9	CNC_S			0.00	D1	CNC_S			0.03			
C2	CNC_S			0.03	CA	CNC_S			0.00	D2	CNC_S			12.24			
C3	CNC_S			12.33	CB	CNC_S			0.00	D3	CNC_S			0.01			

Figure 70. Case Study RMF Channel Path Activity Extract

12.2.1.4 Cache Subsystem Activity Reports Analysis

These reports address questions:

1. Is cache overloaded ?
2. What is the efficiency of staging and destaging processes ?

All %READ fields of reports show 100%, so the I/O activity is read-only.

Focus the analysis on LCU 0046, associated with the CU-ID 2B00; refer to Figure 71 on page 151. In the cache subsystem device overview report, the I/O RATE from the host is 18.2, of which 18.0 are reads and READ H/R is 0.991, which is good. Moreover, in the CACHE MISSES subset of the CACHE SUBSYSTEM OVERVIEW, the SEQUENTIAL LINE shows 93457 tracks staged for sequential prefetch at a rate of 41.1 per second between disk and cache. Those figures describe well the cache behaviour: intensive sequential read demand, satisfied by an intensive pre-staging activity with a good hit ratio. *The disconnect time observed in the device activity report comes from a probable sustained host demand slightly faster than pre-staging.*

This observation can be done on almost every RVA LCU. The CU-ID 2BC0 and 2CC0 show missing data because of a probable operator setting during the test; IXFP reports should complement it.

CACHE SUBSYSTEM ACTIVITY																	
SUBSYSTEM	3990-03	CU-ID	2B00	SSID	0088	CDATE	02/12/1999	CTIME	15.05.40	CINT	00.37.56						
TYPE-MODEL	9393-002																
(RMF report extract)																	

CACHE SUBSYSTEM OVERVIEW																	

TOTAL I/O	41324	CACHE I/O	41324	CACHE OFFLINE	0												
TOTAL H/R	0.991	CACHE H/R	0.991														
CACHE I/O	-----READ I/O REQUESTS-----			-----WRITE I/O REQUESTS-----									%				
REQUESTS	COUNT	RATE	HITS	RATE	H/R	COUNT	RATE	FAST	RATE	HITS	RATE	H/R	READ				
NORMAL	892	0.4	547	0.2	0.613	0	0.0	0	0.0	0	0.0	N/A	100.0				
SEQUENTIAL	40432	17.8	40392	17.7	0.999	0	0.0	0	0.0	0	0.0	N/A	100.0				
CFW DATA	0	0.0	0	0.0	N/A	0	0.0	0	0.0	0	0.0	N/A	N/A				
TOTAL	41324	18.2	40939	18.0	0.991	0	0.0	0	0.0	0	0.0	N/A	100.0				
-----CACHE MISSES-----				-----MISC-----				-----NON-CACHE I/O-----									
REQUESTS	READ	RATE	WRITE	RATE	TRACKS	RATE	COUNT		RATE	ICL		COUNT	RATE				
NORMAL	345	0.2	0	0.0	385	0.2	DFW BYPASS		0	0.0	BYPASS		0	0.0			
SEQUENTIAL	40	0.0	0	0.0	93457	41.1	CFW BYPASS		0	0.0	TOTAL		0	0.0			
CFW DATA	0	0.0	0	0.0			DFW INHIBIT		0	0.0			0	0.0			
							ASYNC (TRKS)		0	0.0							
TOTAL	385	RATE	0.2														
---CKD STATISTICS---				---RECORD CACHING---													
WRITE	0	READ MISSES	0														
WRITE HITS	0	WRITE PROM	0														

CACHE SUBSYSTEM DEVICE OVERVIEW																	

VOLUME	DEV	DUAL	%	I/O	---CACHE HIT RATE---			-----DASD I/O RATE-----					ASYNC	TOTAL	READ	WRITE	%
SERIAL	NUM	COPY	I/O	RATE	READ	DFW	CFW	STAGE	DFWBP	ICL	BYP	OTHER	RATE	H/R	H/R	H/R	READ
*ALL			100.0	18.2	18.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.991	0.991	N/A	100.0
*																	

Figure 71. Case Study RMF Cache Subsystem Activity Extracts

12.2.2 IXFP View

IXFP builds RVA statistics reports at the level of the storage server: it has a hardware standpoint, and so consolidates activities from sharing LPARs. This standpoint also gives to IXFP a good knowledge of data handed into each channel program. Information on functional volumes mapping to physical disk space is also available. For the case study, where only one unique query activity occurred in both the RVA Storage Groups, IXFP information complements the RMF view.

12.2.2.1 Device Performance Overall Summary

The Figure 72 on page 152 shows an extract of these reports, for both RVAs. Three items of information confirm that the storage servers are not at their utilization limit:

- *I/O activity and bandwidth.* For each RVA, there are roughly 45 I/Os per second. This generates a throughput from 5 MB to 5.5 MB per second on each RVA.
- *Service time.* For each RVA, the service time is roughly 30 milliseconds, of which 25% is disconnect time. Connect times are high, because of the 110-120 KB average size of data transferred per I/O.
- *Net capacity load.* The NCL is 56.4% for subsystem 20395 and 29.7% for subsystem 22897.

XSA/REPORTER					DEVICE PERFORMANCE OVERALL SUMMARY									
SUBSYSTEM 20395														
SUBSYSTEM SUMMARY														
	% DEV AVAIL	I/O PER SEC	KBYTES PER SEC	ACCESS DENSITY	-I/O TOTAL	SERVICE DISC	TIME (MS) CONNECT	% DEV UTIL	% DEV DISC	% DEV CONN				
PROD PARTITION	100.0	45.7	5481.4	0.1	30.5	7.3	23.3	0.5	0.1	0.4				
OVERALL TOTALS	100.0	45.7	5481.4	0.1	30.5	7.3	23.3	0.5	0.1	0.4				
SUBSYSTEM 20395 DISK ARRAY SUMMARY														
	AVG % DRIVE MODULE UTIL	COEFF OF VARIATION	NET TEST	CAPACITY PROD	LOAD % OVERALL	FREE SPACE TEST	COLLECTION PROD	LOAD OVERALL	COLL TEST	FREE PROD	SPC (%) OVERALL	UNCOLL TEST	FREE PROD	SPC (%) OVERALL
	10.6	78	0.0	56.4	56.4	0.0	0.0	0.0	0.0	42.4	42.4	0.0	1.2	1.2
SUBSYSTEM 22897														
SUBSYSTEM SUMMARY														
	% DEV AVAIL	I/O PER SEC	KBYTES PER SEC	ACCESS DENSITY	-I/O TOTAL	SERVICE DISC	TIME (MS) CONNECT	% DEV UTIL	% DEV DISC	% DEV CONN				
PROD PARTITION	100.0	44.5	5036.8	0.1	29.7	7.8	22.0	0.5	0.1	0.4				
OVERALL TOTALS	100.0	44.5	5036.8	0.1	29.7	7.8	22.0	0.5	0.1	0.4				
SUBSYSTEM 22897 DISK ARRAY SUMMARY														
	AVG % DRIVE MODULE UTIL	COEFF OF VARIATION	NET TEST	CAPACITY PROD	LOAD % OVERALL	FREE SPACE TEST	COLLECTION PROD	LOAD OVERALL	COLL TEST	FREE PROD	SPC (%) OVERALL	UNCOLL TEST	FREE PROD	SPC (%) OVERALL
	13.2	86	0.0	25.2	25.2	0.0	0.0	0.0	0.0	73.9	73.9	0.0	0.9	0.9

Figure 72. Case Study IXFP Device Performance Case Summary Extract

12.2.2.2 Cache effectiveness Overall Summary

Figure 73 on page 153 shows an extract of these reports for both RVAs that contains information similar to RMF, but with more details on caching algorithms, and also explains the origin of the observed disconnect times. The I/O demand the host submits to the RVAs and the staging effectiveness are analyzed:

- *Analysis of host I/Os.* Both RVAs receive a similar I/O activity demand which is read-only. That explains the high value of the READ RATIO field, which is the read-to-write ratio. Moreover, the sum (read-plus-write) is larger than number of I/Os. There are probably more than one read per I/O.
- *Hit ratios* are more than 99%. That is an indication that practically all reads find their addressed tracks into the cache. This situation is current when a sequential pre-staging algorithm anticipates the I/O read demand.
- *Staging activity* (measured in number of tracks per second) is 2.4 times the I/O activity:

$$(113.2 + 103) / (45.7 + 44.5) = 2.4$$

There are 2.4 tracks staged per I/O: *This intensive staging activity explains the observed disconnect time.*

XSA/REPORTER		CACHE EFFECTIVENESS OVERALL SUMMARY											18FEB1999 17:32:04	
SUBSYSTEM NAME: 20395		(CACHE SIZE: 1024 MB NVS SIZE: 8 MB)												
SUBSYSTEM SUMMARY	READ PER SEC	WRITE PER SEC	I/O PER SEC	READ RATIO	READ HIT %	WRITE HIT %	I/O HIT %	DFW CONSTR	STAGE PER SEC	HITS/STGE	LOW REF CT	TRACK OCCUP		
PROD PARTITION	53.8	0.0	45.7	61329	99.3	100.0	99.3	0.0	113.2	0.5	73.7			
OVERALL TOTALS	53.8	0.0	45.7	61329	99.3	100.0	99.3	0.0	113.2	0.5	73.7	25050		
SUBSYSTEM NAME: 22897		(CACHE SIZE: 1280 MB NVS SIZE: 8 MB)												
SUBSYSTEM SUMMARY	READ PER SEC	WRITE PER SEC	I/O PER SEC	READ RATIO	READ HIT %	WRITE HIT %	I/O HIT %	DFW CONSTR	STAGE PER SEC	HITS/STGE	LOW REF CT	TRACK OCCUP		
PROD PARTITION	50.1	0.0	44.5	57064	99.9	100.0	99.9	0.0	103.0	0.5	73.5			
OVERALL TOTALS	50.1	0.0	44.5	57064	99.9	100.0	99.9	0.0	103.0	0.5	73.5	22086		

Figure 73. Case Study IXFP Cache Effectiveness Overall Extract

12.2.2.3 Space Utilization Summary

Figure 74 on page 154 shows the extracts of space utilization reports for both RVAs. Table 27 on page 153 summarizes different space utilization by both RVAs. It also shows how, in this balanced configuration with an evenly distributed workload and with consistent service times, the NCLs are not homogenous.

Table 27. RVA Space Utilization Comparison

Subsystem	20395	22897
Net Capacity Load	56.4%	25.2%
% Functional Capacity Stored	28.1	5.4
Compression Ratio	3.1	1.9

The partitions of the table spaces into RVA 22897 contain data already compressed by CPU. From performance and transfer bandwidth points of view (KB per second), both RVAs have similar appearances.

XSA/REPORTER		SPACE UTILIZATION SUMMARY REPORT						17FEB1999		16:47:05	
SUBSYSTEM 20395		(NUMBER OF FUNCTIONAL DEVICES: 256)									
SELECTED DEVICES SUMMARY											
	SELECTED DEVICES	TOTAL FUNCTIONAL CAPACITY (MB)	FUNCTIONAL CAPACITY (MB)		% FUNCT CAPACITY		DISK ARRAY			COMP RATIO	
			NOT STORED	STORED	NOT STORED	STORED	SHARED	UNIQUE	TOTAL		
PRODUCTION PARTITION:	256	726532.2	204036.4	522495.8	28.1	71.9	0.0	65964.1	65964.1	3.1	
TOTALS:	256	726532.2	204036.4	522495.8	28.1	71.9	0.0	65964.1	65964.1	3.1	
SUBSYSTEM 20395 SPACE UTILIZATION SUMMARY											
	NUMBER OF FUNCTIONAL DEVICES	DISK ARRAY CAPACITY (MB)	NET CAPACITY LOAD (%)			COLL FREE SPACE (%)			UNCOLL FREE SPACE (%)		
			TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
	256	117880.2	0.0	56.4	56.4	0.0	42.4	42.4	0.0	1.3	1.3
SUBSYSTEM 22897 (NUMBER OF FUNCTIONAL DEVICES: 256)											
SELECTED DEVICES SUMMARY											
	SELECTED DEVICES	TOTAL FUNCTIONAL CAPACITY (MB)	FUNCTIONAL CAPACITY (MB)		% FUNCT CAPACITY		DISK ARRAY			COMP RATIO	
			NOT STORED	STORED	NOT STORED	STORED	SHARED	UNIQUE	TOTAL		
PRODUCTION PARTITION:	256	726532.2	39125.9	687406.3	5.4	94.6	0.0	20157.3	20157.3	1.9	
TOTALS:	256	726532.2	39125.9	687406.3	5.4	94.6	0.0	20157.3	20157.3	1.9	
SUBSYSTEM 22897 SPACE UTILIZATION SUMMARY											
	NUMBER OF FUNCTIONAL DEVICES	DISK ARRAY CAPACITY (MB)	NET CAPACITY LOAD (%)			COLL FREE SPACE (%)			UNCOLL FREE SPACE (%)		
			TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
	256	81609.4	0.0	25.2	25.2	0.0	73.9	73.9	0.0	0.9	0.9

Figure 74. Case Study IXFP Space Utilization Summary Extract

12.3 Case Study Summary

The summary is driven from the application point of view, because application generates the I/O demand. Then correlations are established with the RMF system and IXFP hardware performance monitors. Those correlations are established using spread sheets of which data are manually extracted from various reporting tools.

DB2 PM establishes an exclusively read oriented I/O profile of the activity. Figure 75 on page 155, built from DB2 PM data out of several areas, shows that, per second, there are:

- 84.37 prefetch reads of 32 pages each
- 8.65 synchronous reads of one page

The average wait on request is 8.8 millisecond.

RMF device activity report, see Figure 76 on page 155, shows consistent information for I/O rate of 91.0 and queuing. Queuing covers IOSQ, pending and disconnect. RMF cache activity report (see Figure 77 on page 156) has missing data for one LCU of each RVA.

IXFP report (see Figure 78 on page 156) affords the missing information on staging activity and gives a consistent view of activity (90.2 I/Os per second). The bandwidth I/O demand of 10.518 MB/s allows cross-checking of the average size of each I/O, which is around 117 KB.

Figure 79 on page 157 shows the I/O requests, and resulting staging activity, from the query getpage demand.


```

                                DB2 PM I/O SUMMARY
Elapsed (sec):                2260

BUFFER POOL Tot4K      requests           wait / request
                                per sec      ms per read

getpages                6135875      2714.52

synchronous prefetch    164649      72.84
dynamic prefetch       26065      11.53
total prefetch         190714      84.37      9.4
synchronous reads      19559      8.65      2.9

Total read I/O        210273      93.03      8.8

ACCOUNTING CLASS 3
                                elapsed sec
Synchronous I/O         56.77
Other Read I/O          1799.16

```

Figure 75. DB2PM I/O Summary

```

                                DEVICE ACTIVITY
CU-ID   rate  resp_t  iosq  pend  disc  conn  path  queing
        ssch/sec  ms    ms    ms    ms    ms    %    ms

RVA_1
1st LCU 2B00  15.778  31.8  0.0  0.2  7.1  24.5  38.7  7.3
2nd LCU 2B40  8.752  33.6  2.0  0.2  6.7  24.7  21.6  8.9
3rd LCU 2B80  14.145  27.2  1.0  0.2  5.4  20.6  29.1  6.6
4th LCU 2BC0  7.923  31.9  0.0  0.2  7.0  24.7  19.6  7.2
tot      46.598  30.8  0.7  0.2  6.5  23.4  109.0  7.4

RVA_2
1st LCU 2C00  3.721  33.7  0.0  0.2  9.3  24.2  9.0  9.5
2nd LCU 2C40  14.431  26.3  2.0  0.2  6.3  17.8  25.7  8.5
3rd LCU 2C80  14.540  34.4  0.0  0.2  9.1  25.1  36.5  9.3
4th LCU 22C0  11.668  30.1  0.0  0.2  7.6  22.3  26.0  7.8
tot      44.360  30.6  0.7  0.2  7.8  21.9  97.2  8.7

System 71C0  6.963  3.1  0.0  0.4  0.1  2.6  1.8  0.5

Device Activity with Storage Group Report
SG      91.0  31.0  1.0  0.2  7.1  22.7  206.5  8.3

```

Figure 76. Device Activity Summary

CACHE ACTIVITY									
field	rate	normal read	sequent read	normal staging	sequent staging	destage async	ic1	sequent read	
unit	io/s	io/s	io/s	trk/s	trk/s	io/s	io/s	hit ratio	
RVA_1									
1st LCU	18.2	0.4	17.8	0.2	41.1	0	0	0.999	
2nd LCU	10.1	0.1	10.0	0	23	0	0	1	
3rd LCU	15.9	2.5	13.5	0.1	30.3	0	0	1	
4th LCU	m	m	m	m	m	m	m	m	
tot	44.2	3.0	41.3	0.3	94.4	0,0	0.0		
RVA_2									
1st LCU	4.2	0.1	4.2	0	9.6	0	0	1	
2nd LCU	15.9	4.2	11.8	0.1	26.8	0	0	0.996	
3rd LCU	16.8	0.0	16.7	0	38.6	0	0	1	
4th LCU	m	m	m	m	m	m	m	m	
tot	36.9	4.3	32.7	0.1	75.0	0.0	0.0		
RVA	81.1	7.3	74.0	0.4	169.4	0.0	0.0		
Legend:									
ic1	Inhibit cache load = DB2 bypass cache								
m	missing data								

Figure 77. Cache Activity Summary

IXFP										
	io_rate	Kbyte/s	io serv	disc	conn	stage	hits/stg	r_hit	comp	ncl
	ssch/sec		ms	ms	ms	trk/s	ratio	%	ratio	%
rva 1	45.700	5481.4	30.6	7.3	23.3	113.2	0.5	99.3	3.1	56.4
rva 2	44.500	5036.8	29.8	7.8	22.0	103	0.5	99.9	1.9	25.2
rva 1+2	90.2	10518	30.2	7.5	22.7	216.2				
KB / I/O	116.61									

Figure 78. IXFP Summary

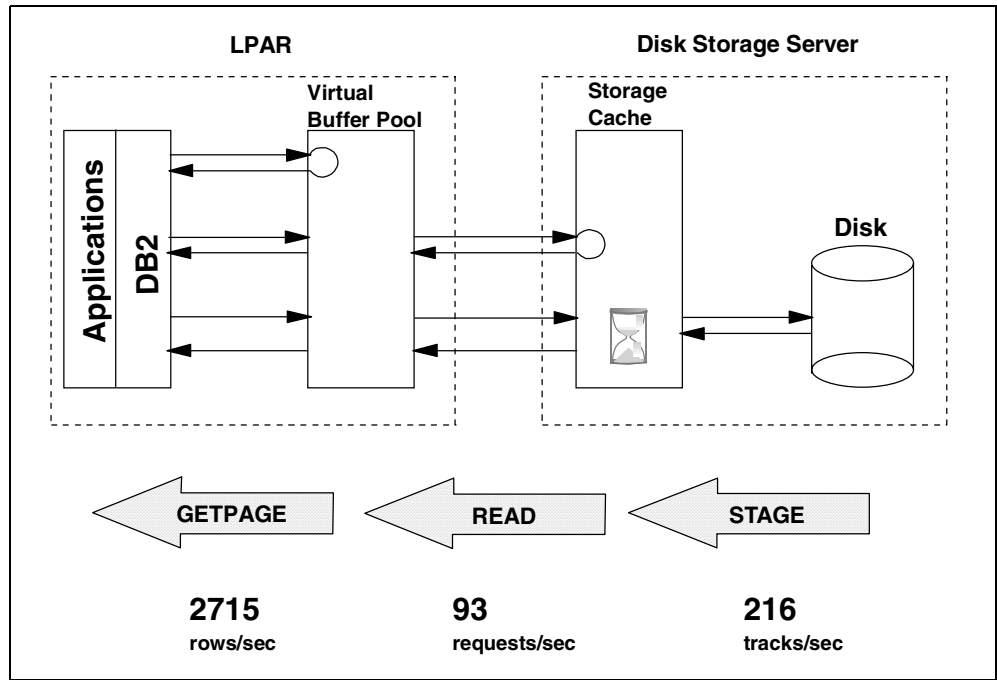


Figure 79. Case Study I/O Flows

Part 4. Appendixes

Appendix A. Test Cases for DB2 Table Space Data Sets

This appendix shows the different test cases generated during the writing of this publication, related to the allocation of table spaces (and indexes). The following tests are documented:

- TEST CASE 1 Appendix A.2, "Partitioned Table Space, DB2 Defined, Without SMS" on page 162
- TEST CASE 2 Appendix A.3, "Partitioned Table Space, User Defined, Without SMS" on page 164
- TEST CASE 3 Appendix A.4, "DB2 Table Spaces Using SMS, Existing Names" on page 165
- TEST CASE 4 Appendix A.5, "DB2 Table Spaces Using SMS, Coded Names" on page 174
- TEST CASE 5 Appendix A.6, "Partitioned Table Space Using SMS Distribution" on page 178
- TEST CASE 6 Appendix A.7, "Partitioned Table Spaces Using SMS, User Distribution" on page 181

All tests were performed under laboratory conditions, and are presented as examples of the options and methodology used to achieve the end results.

A.1 Test Environment

Figure 80 on page 161 shows the environment used for the tests. Three RVA units are available; each has four control units (CU0 - CU3). One volume on each of the control units is available for the tests.

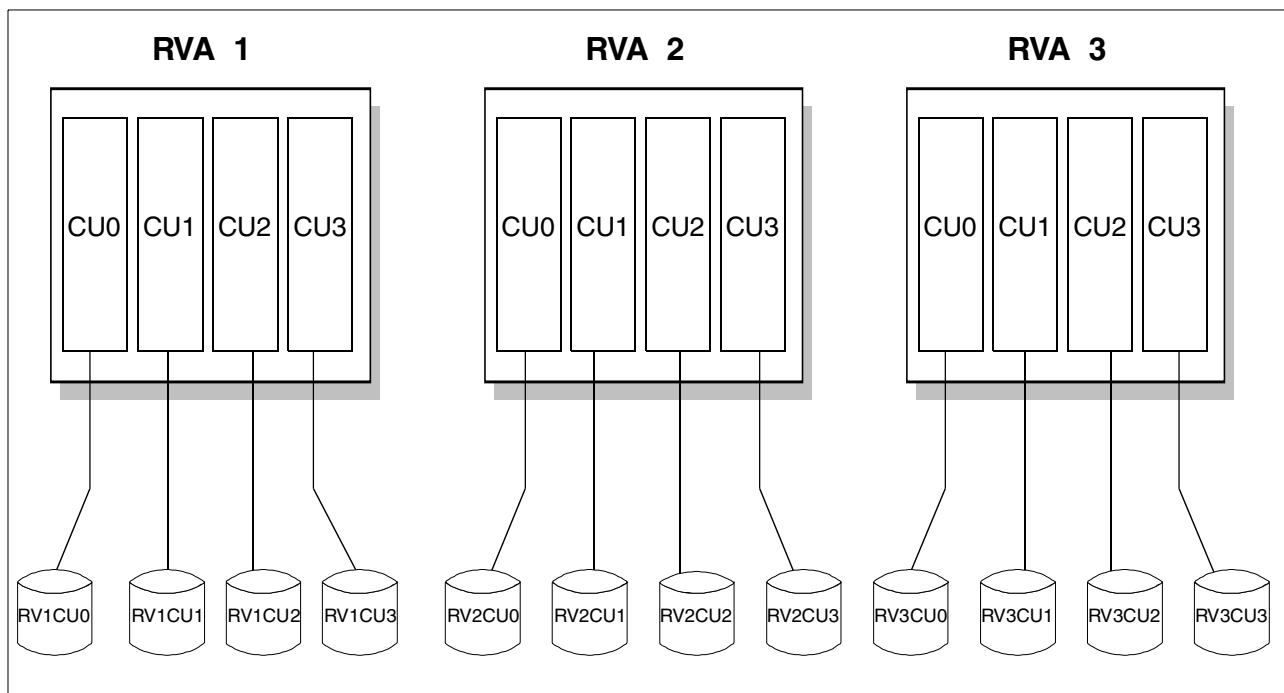


Figure 80. Disk Volume Configuration Used in the Test Environment

A.2 Partitioned Table Space, DB2 Defined, Without SMS

Test case 1 illustrates how to allocate a DB2 partitioned table space using DB2 defined data sets, without SMS.

The purpose is to distribute the different data sets across multiple volumes and access paths, in order to obtain maximum benefit from DB2 parallelism. This example allocates a partitioned table space with 16 partitions on eight volumes. Two partitions are placed on each volume.

A.2.1 Create Eight STOGROUPS

Figure 81 on page 162 shows an example of the CREATE STOGROUP statement. Eight similar statements are required to create all eight Storage Groups:

- SGRV1CU0
- SGRV1CU1
- SGRV1CU2
- SGRV2CU0
- SGRV2CU1
- SGRV3CU0
- SGRV3CU1
- SGRV3CU2

```
CREATE STOGROUP SGRV1CU0
VOLUMES ("RV1CU0")
VCAT DB2V610Z;
```

Figure 81. Test Case 1 - CREATE STOGROUP

A.2.2 Create the Database

The CREATE DATABASE statement is shown in Figure 82 on page 162. Any STOGROUP can be specified here, because it is overridden in the CREATE TABLESPACE statement.

```
CREATE DATABASE BPAOLOR1
STOGROUP SGRV1CU0
BUFFERPOOL BPO
CCSID EBCDIC
```

Figure 82. Test Case 1 - CREATE DATABASE

A.2.3 Create the Table Space

The CREATE TABLESPACE statement is shown in Figure 83 on page 163. In this statement, each partition is directed to a specific STOGROUP.

```
CREATE TABLESPACE PART1
  IN BPAOLR1
  USING STOGROUP SGRV1CU0
    PRIQTY 20
    SECQTY 20
    ERASE NO
  NUMPARTS 16

  (PART 1 USING STOGROUP SGRV1CU0
    PRIQTY 720
    SECQTY 720,

  PART 2 USING STOGROUP SGRV1CU1
    PRIQTY 720
    SECQTY 720,

  ....

  PART 16 USING STOGROUP SGRV3CU2
    PRIQTY 720
    SECQTY 720)

LOCKSIZE ANY LOCKMAX SYSTEM
BUFFERPOOL BP0
CLOSE NO
COMPRESS YES
CCSID EBCDIC
```

Figure 83. Test Case 1 - CREATE TABLESPACE

A.2.4 Display a Volume

Volumes can be displayed, to ensure that the allocation was done correctly. As an example, Figure 84 on page 163 shows the contents of volume RV1CU0. From this figure we can see that only two data sets are allocated on this volume. This is the expected result:

```
Menu Options View Utilities Compilers Help
-----
DSLISL - Data Sets on volume RV1CU0                               Row 1 of 4
Command ==>                                                       Scroll ==> CSR

Command - Enter "/" to select action                               Message                               Volume
-----
      DB2V610Z.DSNDBD.BPAOLR1.PART1.I0001.A001                    RV1CU0
      DB2V610Z.DSNDBD.BPAOLR1.PART1.I0001.A009                    RV1CU0
      SYS1.VTOCIX.RV1CU0                                           RV1CU0
      SYS1.VVDS.VRV1CU0                                           RV1CU0
***** End of Data Set list *****
```

Figure 84. Test Case 1 - Display of Volume RV1CU0

A.3 Partitioned Table Space, User Defined, Without SMS

Test case 2 illustrates how to allocate a DB2 partitioned table space using user defined data sets, without SMS. The objective and the table space are the same as for Appendix A, section A.2, "Partitioned Table Space, DB2 Defined, Without SMS" on page 162.

A.3.1 DEFINE CLUSTER for 16 Partitions

Sixteen DEFINE CLUSTER statements with explicit volume names, as shown in Figure 85 on page 164, are needed to create the 16 partitions:

```
DEFINE CLUSTER -
      ( NAME (DB2V610Z.DSNDBC.BPAOLOR1.PART2.I0001.A005) -
        LINEAR
      -
        REUSE -
        VOLUMES (RV2CU1) -
        RECORDS (180 180) -
        SHAREOPTIONS (3 3) ) -
      DATA -
```

Figure 85. Test Case 2 - DEFINE CLUSTER

A.3.2 CREATE STOGROUP

Only one STOGROUP is required for this test case. Figure 86 on page 164 shows the related CREATE STOGROUP statement.

```
CREATE STOGROUP SGRV1CU0
      VOLUMES ("RV1CU0", "RV1CU1", "RV1CU2", "RV1CU3",
              "RV2CU0", "RV2CU1", "RV2CU2", "RV2CU3",
              "RV3CU0", "RV3CU1", "RV3CU2", "RV3CU3")
      VCAT DB2V610Z;
```

Figure 86. Test Case 2 - CREATE STOGROUP

A.3.3 CREATE DATABASE

The database required for this test case is identical to that of test case 1 on Figure 82 on page 162.

A.3.4 CREATE TABLESPACE

Create the table space, using the high level qualifier (VCAT) to reference the clusters created in Appendix A, section A.3.1, "DEFINE CLUSTER for 16 Partitions" on page 164.

```

CREATE TABLESPACE PART2
  IN BPAOLOR1
  USING STOGROUP SGRV1CU0
  NUMPARTS 16
  (PART 1 USING VCAT DB2V610Z,
   PART 2 USING VCAT DB2V610Z,
   PART 3 USING VCAT DB2V610Z,
   PART 4 USING VCAT DB2V610Z,
   PART 5 USING VCAT DB2V610Z,
   .... ..)

```

Figure 87. Test Case 2 - CREATE TABLESPACE

A.3.5 Display a Volume

Volumes can be displayed, to ensure that the allocation was done correctly. As an example, Figure 88 on page 165 shows the contents of volume RV2CU1. From this figure we can see that only two data sets are allocated on this volume. This is the expected result:

```

Menu Options View Utilities Compilers Help
-----
DSLISL - Data Sets on volume RV2CU1                               Row 1 of 4
Command ==>                                                       Scroll ==> CSR

Command - Enter "/" to select action                               Message                               Volume
-----
DB2V610Z.DSNDBD.BPAOLOR1.PART2.I0001.A005                       RV2CU1
DB2V610Z.DSNDBD.BPAOLOR1.PART2.I0001.A013                       RV2CU1
SYS1.VTOCIX.RV2CU1                                               RV2CU1
SYS1.VVDS.VRV2CU1                                                RV2CU1
***** End of Data Set list *****

```

Figure 88. Test Case 2 - Display of Volume RV2CU1

A.4 DB2 Table Spaces Using SMS, Existing Names

Test case 3 illustrates a table space allocation using SMS. The purpose is to show a simple step-by-step process to achieve the placement of data under SMS control, and is based upon the following environment:

- Two existing DB2 subsystems; DB2D (development), and DB2P (production) are to be converted.
- The performance, availability, management, and volume location criteria of the data sets are shown in 6.1, “SMS Examples for DB2 Databases” on page 47.
- The existing naming convention for DB2 data sets uses the DB2 subsystem name as high level qualifier. There is no naming convention for databases and tablespaces.
- The following conditions are agreed upon by the DB2 administrator and the storage administrator for the conversion:
 - Most production databases will be allocated to volumes in the SGDB20 storage group. Table spaces in the critical database BACCTS are allocated on SGDBCRIT. Table spaces in the high performance

databases, BCUSTOMR, BSERVICE, BTRANS are allocated on SGDBFAST.

- All table spaces in the development system will be allocated on SGDBTEST.
- The development databases are subject to migration by HSM. Those databases with a name starting with B will get preferential treatment.

A.4.1 Storage Classes

Using ISMF, option 5.3, the following Storage Classes were defined:

- SCDBTEST
- SCDBFAST
- SCDBMED
- SCDBCRIT

Figure 89 on page 166 shows page 1 of the associated panel used by the storage administrator to build the definition for **SCDBTEST**.

```

                                STORAGE CLASS DEFINE                                Page 1 of 2
Command ==>>

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Class Name  : STORAGE CLASS FOR TEST DB2 TABLESPACES
To DEFINE Storage Class, Specify:
  Description ==>
  ==>
Performance Objectives
Direct Millisecond Response . . . . 20          (1 to 999 or blank)
Direct Bias . . . . .                (R, W or blank)
Sequential Millisecond Response . . 20          (1 to 999 or blank)
Sequential Bias . . . . .            (R, W or blank)
Initial Access Response Seconds . .          (0 to 9999 or blank)
Sustained Data Rate (MB/sec) . . . 10          (0 to 999 or blank)
Availability . . . . . STANDARD          (C, P ,S or N)
Accessibility . . . . . STANDARD        (C, P ,S or N)
Backup . . . . .                      (Y, N or Blank)
Versioning . . . . .                   (Y, N or Blank)

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

Figure 89. Test Case 3 - ISMF Storage Class Definition

Next, using ISMF, option 7.1 to edit the ACS routine source, code was written to assign Storage Classes for all Table space data sets with a HLQ of **DB2D** or **DB2P**. Figure 90 on page 167 displays the associated extract.

```

/*****/
/* STORAGE CLASS                               */
/* FILTLIST DEFINITIONS                       */
/*****/

FILTLIST SCDBMED  INCLUDE (DB2P.DSNDB%.**)
                  EXCLUDE (DB2P.DSNDB%.BCUSTOMR.**,
                           DB2P.DSNDB%.BSERVICE.**,
                           DB2P.DSNDB%.BTRANS.**,
                           DB2P.DSNDB%.BACCTS.**)

FILTLIST SCDBCRIT INCLUDE (DB2P.DSNDB%.BACCTS.**)

FILTLIST SCDBFAST INCLUDE (DB2P.DSNDB%.BCUSTOMR.**,
                           DB2P.DSNDB%.BSERVICE.**,
                           DB2P.DSNDB%.BTRANS.**)

FILTLIST SCDBTEST INCLUDE (DB2D.DSNDB%.**)

/*****/
/* SELECTION ROUTINE FOR DB2 TABLE SPACES   */
/*****/

SELECT
  WHEN (&DSN = &SCDBTEST)
    SET &STORCLAS = 'SCDBTEST'
  WHEN (&DSN = &SCDBFAST)
    SET &STORCLAS = 'SCDBFAST'
  WHEN (&DSN = &SCDBMED)
    SET &STORCLAS = 'SCDBMED'
  WHEN (&DSN = &SCDBCRIT)
    SET &STORCLAS = 'SCDBCRIT'

    OTHERWISE SET &STORCLAS = ''
END

```

Figure 90. Test Case 3 - Storage Class Routine Extract

A.4.2 Management Class

This exercise requires multiple management attributes for the Table spaces according to the space name specified in the data set name. Therefore, using ISMF, option 3.3, three Management Classes, **MCDB20**, **MCDB21**, and **MCDB22** were defined accordingly. Figure 91 on page 168 shows an example of page 2 of the associated panel required to achieve this:

Command ==>

CDS Name : SMS.SCD1.SCD1
 Management Class Name . . . : MCDB21

Partial Release : CONDITIONAL

Migration Attributes

Primary Days Non-usage . : 7
 Level 1 Days Date/Days . : 14
 Command or Auto Migrate . : BOTH

GDG Management Attributes

GDG Elements on Primary :
 Rolled-off GDS Action . . :

Figure 91. Test Case 3 - ISMF Management Class Definition

Next, using ISMF, option 7.1, the ACS routine code was edited to assign the appropriate Management Classes to all data sets with the corresponding naming convention. Figure 92 on page 168 displays the associated routine extract:

```

/*****
/* MANAGEMENT CLASS                               */
/* FILTLIST DEFINITIONS                           */
/*****

FILTLIST MCDB20    INCLUDE (DB2P.DSNDB%.**)

FILTLIST MCDB21   INCLUDE (DB2D.DSNDB%.B%.**)

FILTLIST MCDB22   INCLUDE (DB2D.DSNDB%.*%.**)

/*****
/* SELECTION ROUTINE FOR DB2 TABLE SPACES      */
/*****
SELECT
  IF &DSN EQ MCDB20
    THEN DO
      SET &MGMTCLAS = MCDB20
    EXIT
  END

  IF &DSN EQ MCDB21
    THEN DO
      SET &MGMTCLAS = MCDB21
    END
  ELSE DO
    SET &MGMTCLAS = 'MCDB22'
  END
END

```

Figure 92. Test Case 3 - Management Class Routine Extract

A.4.3 Storage Group

Four Storage Groups, **SGDBFAST**, **SGDB20**, **SGDBCRCIT**, and **SGDBTEST** were defined using ISMF, option 6.2. Figure 93 on page 169 shows the associated panel used by the storage administrator for the definition of POOL Storage Groups:

```
                                POOL STORAGE GROUP DEFINE
Command ==>>

SCDS Name . . . . . : SMS.SCD1.SCD1
Storage Group Name : SGDBTEST
To DEFINE Storage Group, Specify:
Description ==> STORAGE GROUP FOR DB2 TEST TABLE SPACES
==>
Auto Migrate . . Y (Y, N, I or P)   Migrate Sys/Sys Group Name . .
Auto Backup . . n (Y or N)         Backup Sys/Sys Group Name . .
Auto Dump . . . n (Y or N)        Dump Sys/Sys Group Name . . .

Dump Class . . .                   (1 to 8 characters)
Dump Class . . .                   Dump Class . . .
Dump Class . . .                   Dump Class . . .

Allocation/migration Threshold: High . . 60 (1-99)      Low . . 25 (0-99)
Guaranteed Backup Frequency . . . . . (1 to 9999 or NOLIMIT)

DEFINE SMS Storage Group Status . . . . . N (Y or N)
```

Figure 93. Test Case 3 - ISMF Pool Storage Group Definition

Next, using the ISMF edit facility, the code was added to the ACS source to allow the selection of the newly defined Storage Groups, based upon the Storage Class variable &STORCLAS. Figure 94 on page 169 shows the sample code used:

```
/******
/* STORAGE GROUP SELECTION ROUTINE */
/******

SELECT
  WHEN (&STORCLAS = 'SCDBTEST')
    SET &STORGRP = 'SGDBTEST'
  WHEN (&STORCLAS = 'SCDBFAST')
    SET &STORGRP = 'SGDBFAST'
  WHEN (&STORCLAS = 'SCDBCRCIT')
    SET &STORGRP = 'SGDBCRCIT'
  WHEN (&STORCLAS = 'SCDBMED')
    SET &STORGRP = 'SGDB20'
END
```

Figure 94. Test Case 3 - Storage Group Routine Extract

Three disk volumes were allocated to Storage Group SGDBTEST, one on each available RVA. Likewise, two volumes were assigned to each of the other Storage Groups, SGDBFAST, SGDBCRCIT and SGDB20. Figure 95 on page 170 shows an

example of the ISMF panel option 6.4 used by the storage administrator to define volumes.

Table 28. Test Case 3 - Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDBTEST	RV1CU3 RV2CU3 RV3CU3
SGDB20	RV1CU1
SGDBFAST	RV1CU0
SGDBCRT	RV2CU0

```

                                STORAGE GROUP VOLUME SELECTION
Command ==>>

CDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name : SGDBTEST
Storage Group Type : POOL

Select One of the following Options:
 2 1. Display - Display SMS Volume Statuses (Pool only)
   2. Define - Add Volumes to Volume Serial Number List
   3. Alter - Alter Volume Statuses (Pool only)
   4. Delete - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
Prefix From To Suffix Hex
====> RV1CU3 _____ ('X' in HEX field allows
====> RV2CU3 FROM - TO range to include
====> RV3CU3 hex values A through F.)
====>

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

```

Figure 95. Test Case 3 - ISMF Storage Group Volume Definition

DFSMSdss was used to convert these volumes to SMS management, by specifying the **CONVERTV** parameter. Figure 96 on page 170 shows the JCL used for this process:

```

//STEP1 EXEC PGM=ADDRSSU
//SYSPRINT DD SYSOUT=*
//DASD1 DD UNIT=3390,VOL=SER=RV1CU3,DISP=SHR
//DASD2 DD UNIT=3390,VOL=SER=RV2CU3,DISP=SHR
//DASD3 DD UNIT=3390,VOL=SER=RV3CU3,DISP=SHR
//SYSIN DD *
        CONVERTV -
          DDNAME(DASD1,DASD2,DASD3) -
          SMS
/*

```

Figure 96. Test Case 3 - DFSMSdss CONVERTV JCL

Figure 97 on page 171 shows the output from the executed batch job:

```

PAGE 0001      5695-DF175 DFSMSDSS V1R5.0 DATA SET SERVICES      1999.047 19:40
              CONVERTV -                                         00060000
              DDNAME (DASD1,DASD2,DASD3) -                       00070001
              SMS                                                00080001
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'CONVERTV'
ADR109I (R/I)-RI01 (01), 1999.047 19:40:33 INITIAL SCAN OF USER CONTROL STATEMENT
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 1999.047 19:40:33 EXECUTION BEGINS
ADR860I (001)-KVSMS(01), PROCESSING BEGINS ON VOLUME RV1CU3
ADR873I (001)-KVSMS(01), VOLUME RV1CU3 IN STORAGE GROUP SGBDBTEST IS ELIGIBLE FOR
                          CONVERSION TO SMS MANAGEMENT
ADR880I (001)-KVSMS(01), VOLUME RV1CU3 IS EMPTY. NO DATA SETS CONVERTED
ADR885I (001)-KVSMS(01), VOLUME RV1CU3 HAS BEEN SUCCESSFULLY CONVERTED TO SMS MANAGEMENT
ADR860I (001)-KVSMS(01), PROCESSING BEGINS ON VOLUME RV2CU3
ADR873I (001)-KVSMS(01), VOLUME RV2CU3 IN STORAGE GROUP SGBDBTEST IS ELIGIBLE FOR
                          CONVERSION TO SMS MANAGEMENT
ADR880I (001)-KVSMS(01), VOLUME RV2CU3 IS EMPTY. NO DATA SETS CONVERTED
ADR885I (001)-KVSMS(01), VOLUME RV2CU3 HAS BEEN SUCCESSFULLY CONVERTED TO SMS MANAGEMENT
ADR860I (001)-KVSMS(01), PROCESSING BEGINS ON VOLUME RV3CU3
ADR873I (001)-KVSMS(01), VOLUME RV3CU3 IN STORAGE GROUP SGBDBTEST IS ELIGIBLE FOR
                          CONVERSION TO SMS MANAGEMENT
ADR880I (001)-KVSMS(01), VOLUME RV3CU3 IS EMPTY. NO DATA SETS CONVERTED
ADR885I (001)-KVSMS(01), VOLUME RV3CU3 HAS BEEN SUCCESSFULLY CONVERTED TO SMS MANAGEMENT
PAGE 0002      5695-DF175 DFSMSDSS V1R5.0 DATA SET SERVICES      1999.047 19:40
ADR892I (001)-KVRPT(01), THE STATUS OF EACH VOLUME IS AS FOLLOWS
              VOLUME          FINAL STATUS          REASON FOR FAILURE
              -----          -
              RV1CU3 - CONVERTED          SMS
              RV2CU3 - CONVERTED          SMS
              RV3CU3 - CONVERTED          SMS
ADR006I (001)-STEND(02), 1999.047 19:40:39 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 1999.047 19:40:39 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 1999.047 19:40:39 DFSMSDSS PROCESSING COMPLETE. HIGHEST

```

Figure 97. Test Case 3 - DFSMSdss CONVERTV Output

A.4.4 ISMF Test Cases

Prior to permanently updating the SMS configuration, ISMF, option 7.4, was used to compare the active configuration (ACDS) against the updated source found in the SCDS. Figure 98 on page 171 shows the test case results for an existing Table space pattern name, **DB2D.DSNDBC.BTRANS.TEST1.I0001.A001**, against the ACDS. Here a *null* Storage Class is assigned, thus preventing the data set from becoming SMS managed:

```

                                ACS TESTING RESULTS
CDS NAME           : ACTIVE
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY  : PAOLOR3.JCL.CNTL

ACS TEST
MEMBER            EXIT CODE  RESULTS
-----          -
DESCRIPTION: DB2D.DSNDBC.BTRANS.TEST1.I0001.A001
SMSTEST1         0 DC = NULL VALUE ASSIGNED
                 0 SC = NULL VALUE ASSIGNED
NOTE: MC AND SG NOT EXECUTED WHEN ACS READ/WRITE VARIABLE STORCLAS = ''

ACS TESTING RC: 00

```

Figure 98. Test Case 3 - ISMF Test against the ACDS

Figure 99 on page 172 shows the test case results for the same pattern table space name against the updated SCDS. This time, with the relevant source code in place, the SMS qualification is successful, and Management Class and Storage Group attributes are also assigned:

```

                                ACS TESTING RESULTS

CDS NAME           : SMS.SCDS1.SCDS
ACS ROUTINE TYPES : DC SC MC SG
ACS TEST LIBRARY  : PAOLOR3.JCL.CNTL

ACS TEST
MEMBER            EXIT CODE  RESULTS
-----
DESCRIPTION: DB2D.DSNDBC.BTRANS.TEST1.I0001.A001
SMSTEST1          0 DC = NULL VALUE ASSIGNED
                  0 SC = SCDBTEST
                  0 MC = MCDB21
                  0 SG = SGDBTEST

ACS TESTING RC:   00

```

Figure 99. Test Case 3 - ISMF Test against the Updated SCDS

A.4.5 Updating the Active Configuration

For each update of an ACS routine or construct definition, a procedure of translating and validating the code must be performed by the storage administrator prior to activating the new configuration. This is executed using ISMF, option 7 panels. See *DFSMS/MVS V1R4 DFSMSdfp Storage Administration Reference*, SC26-4920, for further information on this process.

A.4.6 DB2 Definitions

To allow SMS to control the table space allocation, an appropriate SQL statement must be coded to reflect this in the definition of a DB2 STOGROUP. Figure 100 on page 172 shows the parameter, **VOLUMES("*)**, used to achieve this for one of the STOGROUPs.

```

CREATE STOGROUP SGRV1CU0

VOLUMES ("*")

VCAT DB2D;

```

Figure 100. Test Case 3 - CREATE STOGROUP

Figure 101 on page 173 shows an extract of the CREATE statement used to define the databases for the purpose of this test:

```

CREATE DATABASE BTRANS
      STOGROUP SGRV1CU0
      ...
CREATE DATABASE TESTDB
      STOGROUP SGRV1CU0
      ...

```

Figure 101. Test Case 3 - CREATE DATABASE Extract

Figure 102 on page 173 shows an extract of the CREATE TABLESPACE statements used for the purposes of this exercise:

```

CREATE TABLESPACE CHECKS
  IN BTRANS
  USING STOGROUP SGRV1CU0
        PRIQTY 1024000
        SECQTY 512000 ....
CREATE TABLESPACE TEST3
  IN BTRANS
  USING STOGROUP SGRV1CU0
        PRIQTY 1024000
        SECQTY 512000 ....
CREATE TABLESPACE TEST3
  IN TESTDB
  USING STOGROUP SGRV1CU0
        PRIQTY 1024000
        SECQTY 512000 ....

```

Figure 102. Test Case 3 - CREATE TABLESPACE Extract

A.4.7 Data Set Allocation Results

Figure 103 on page 173, using ISPF option 3.4, shows a data set list of the three table spaces defined in this exercise to the SGDBTEST Storage Group. Using standard volume selection criteria provided by SMS (see 5.3.4, "Storage Group" on page 43), the data sets were evenly distributed across the three assigned disk volumes.

```

Menu Options View Utilities Compilers Help
-----
DSLIST - Data Sets Matching DB2D                               Row 1 of 4
Command ==>                                                    Scroll ==> HALF

Command - Enter "/" to select action                          Message                               Volume
-----
      DB2D                                                       *ALIAS
      DB2D.DSNDBC.BTRANS.CHECKS.I0001.A001                    *VSAM*
      DB2D.DSNDBC.BTRANS.TEST3.I0001.A001                     *VSAM*
      DB2D.DSNDBC.TESTDB.TEST3.I0001.A001                     *VSAM*
      DB2D.DSNDBD.BTRANS.CHECKS.I0001.A001                    RV3CU3
      DB2D.DSNDBD.BTRANS.TEST3.I0001.A001                     RV1CU3
      DB2D.DSNDBD.TESTDB.TEST3.I0001.A001                     RV2CU3

```

Figure 103. Test Case 3 - ISPF Data Set List Display

Using the IDCAMS LISTCAT command, it can be seen from this extract in Figure 104 on page 174, that the catalog retains the SMS attributes assigned at data set allocation time, and shows the space attributes that were allocated in cylinders:

```

CLUSTER ----- DB2D.DSNDBC.BTRANS.CHECKS.I0001.A001
IN-CAT --- UCAT.VSBOX09
HISTORY
DATASET-OWNER-----HAIMO      CREATION-----1999.048
RELEASE-----2              EXPIRATION-----0000.000
SMSDATA
STORAGECLASS ---SCDBTEST      MANAGEMENTCLASS-MCDB21
DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
ALLOCATION
SPACE-TYPE-----CYLINDER     HI-A-RBA-----1049149440
SPACE-PRI-----1423          HI-U-RBA-----1474560
SPACE-SEC-----712
VOLUME
VOLSER-----RV3CU3          PHYREC-SIZE-----4096

```

Figure 104. Test Case 3 - IDCAMS LISTCAT Display Extract

A.5 DB2 Table Spaces Using SMS, Coded Names

Test case 4 illustrates a table space allocation using SMS with a formal naming convention policy. The following environment applies:

- There are three DB2 subsystems; DB2T, DB2D, and DB2P.
- The following conditions are agreed upon by the DB2 administrator and the storage administrator for the management of DB2 data sets:
 - The performance, availability, management and volume location criteria of the data sets are shown in 6.1, “SMS Examples for DB2 Databases” on page 47.
 - The naming convention used is shown in 6.1.8, “Table Space and Index Space Names for SMS” on page 56.
 - The following codes are used to establish the storage class:

```

T      SCDBTEST
M      SCDBMED
F      SCDBFAST
C      SCDBCRT

```

- The following codes are used to establish the management class:

```

0      MCDB0
1      MCDB1
2      MCDB2

```

A.5.1 Storage Class

The Storage Class routine was coded to incorporate the previously defined Storage Classes with the new naming convention. Figure 105 on page 176 shows an extract of the code used.

The FILTLIST definitions for SCDBFAST and SCDBCRT were coded to ensure that table spaces assigned with these attributes would not be subject to HSM migration. Any attempt to deviate from this naming pattern would result in a *null* Storage Class being assigned.

A.5.2 Management Class

The Management Classes from the previous exercise were used, but the ACS code was amended to reflect the appropriate naming patterns. Figure 106 on page 176 shows an extract of the FILTLIST definitions:

A.5.3 Storage Groups

No changes were made to the Storage Group ACS routine from test case three. The volumes allocated to each storage group are shown in Table 29 on page 175.

Table 29. Test Case 4 - Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDBTEST	RV1CU2 RV3CU2
SGDB20	RV1CU1 RV3CU1
SGDBFAST	RV1CU0 RV2CU0
SGDBCRT	RV2CU3 RV3CU3

```

/*****/
/* STORAGE CLASS                               */
/* FILTLIST DEFINITIONS                         */
/*****/

FILTLIST SCDBMED  INCLUDE (DB2P.DSNDB%.M*.**)

FILTLIST SCBCRIT  INCLUDE (DB2P.DSNDB%.CO*.**)

FILTLIST SCDBFAST INCLUDE (DB2P.DSNDB%.F0*.**)

FILTLIST SCDBTEST INCLUDE (DB2D.DSNDB%.T*.**,
                           DB2T.DSNDB%.T*.**)
/*****/
/* SELECTION ROUTINE FOR DB2 TABLE SPACES     */
/*****/

IF &DSN = &SCDBMED
  THEN DO
    SET &STORCLAS = 'SCDBMED'
    EXIT
  END

IF &DSN = &SCBCRIT
  THEN DO
    SET &STORCLAS = 'SCBCRIT'
    EXIT
  END

IF &DSN = &SCDBFAST
  THEN DO
    SET &STORCLAS = 'SCDBFAST'
    EXIT
  END

IF &DSN = &SCDBTEST
  THEN DO
    SET &STORCLAS = 'SCDBTEST'
  END
ELSE DO
  SET &STORCLAS = ''
  END
END

```

Figure 105. Test Case 4 - Storage Class Routine Extract

```

/*****/
/* MANAGEMENT CLASS                             */
/* FILTLIST DEFINITIONS                         */
/*****/

FILTLIST MCDB20  INCLUDE (DB2P.DSNDB%.%0*.**,
                           DB2T.DSNDB%.T0*.**,
                           DB2D.DSNDB%.T0*.**)

FILTLIST MCDB21  INCLUDE (DB2P.DSNDB%.M1*.**,
                           DB2D.DSNDB%.T1*.**,
                           DB2T.DSNDB%.T1*.**)

FILTLIST MCDB22  INCLUDE (DB2D.DSNDB%.T2*.**,
                           DB2T.DSNDB%.T2*.**)

```

Figure 106. Test Case 4 - Management Class Extract

A.5.4 DB2 Definitions

1. Three STOGROUP's were defined on system DB2P:
 - SGBTRANS with (VOLUMES(*))
 - SGCUSTMR with (VOLUMES(*))
 - SGBRANCH with (VOLUMES(*))
2. One STOGROUP was defined on system DB2D:
 - SGTEST with (VOLUMES(*))
3. Three databases were defined on system DB2P:
 - BTRANS, using STOGROUP SGBTRANS
 - CUSTOMER, using STOGROUP SGCUSTMR
 - BRANCH, using STOGROUP SGBRANCH
4. One database was defined on system DB2D:
 - TESTRANS, using STOGROUP SGTEST
5. Three table spaces were defined on system DB2P:
 - M1CHECK in BTRANS
 - F0CUST01 in CUSTOMER
 - C0BRCH01 in BRANCH
6. One table space was defined on system DB2D:
 - T2SUPPLY in TESTRANS

A.5.5 Data Set Allocation Results

Once test cases had been built using ISMF, and the ACDS updated, the table spaces were allocated. Figure 107 on page 178 shows an IDCAMS LISTCAT extract of the four table spaces, each displaying different SMS attributes, and volume allocation.

```

CLUSTER ----- DB2P.DSNDBC.BRANCH.C0BRCH01.I0001.A001
  IN-CAT --- UCAT.VSBOX09
  HISTORY
SMSDATA
  STORAGECLASS ---SCDBCRIT      MANAGEMENTCLASS---MCDB20
  DATACLASS -----(NULL)      LBACKUP ---0000.000.0000
VOLUME
  VOLSER-----RV3CU3

CLUSTER ----- DB2P.DSNDBC.BTRANS.M1CHECK.I0001.A001
  IN-CAT --- UCAT.VSBOX09
  HISTORY
SMSDATA
  STORAGECLASS ----SCDEMED      MANAGEMENTCLASS---MCDB21
  DATACLASS -----(NULL)      LBACKUP ---0000.000.0000
VOLUME
  VOLSER-----RV3CU1

CLUSTER ----- DB2P.DSNDBC.CUSTOMER.F0CUST01.I0001.A001
  IN-CAT --- UCAT.VSBOX09
  HISTORY
SMSDATA
  STORAGECLASS ---SCDBFAST      MANAGEMENTCLASS---MCDB20
  DATACLASS -----(NULL)      LBACKUP ---0000.000.0000
VOLUME
  VOLSER-----RV1CU0

CLUSTER ----- DB2D.DSNDBC.TESTTRANS.T2SUPPLY.I0001.A001
  IN-CAT --- UCAT.VSBOX09
  HISTORY
SMSDATA
  STORAGECLASS ---SCDBTEST      MANAGEMENTCLASS---MCDB22
  DATACLASS -----(NULL)      LBACKUP ---0000.000.0000
VOLUME
  VOLSER-----RV1CU2

```

Figure 107. Test Case 4 - IDCAMS LISTCAT Extract

A.6 Partitioned Table Space Using SMS Distribution

Test case 5 illustrates how it is possible to define a partitioned table space, allowing SMS to control the allocation of each separate partition.

The aim is to distribute the different data sets across multiple disk volumes and access paths, in order to obtain the benefits of DB2 parallelism. This example defines a partitioned table space with eight partitions, using an existing naming convention.

A.6.1 Define Volumes to SMS Storage Group

The SMS Storage Group SGDBTEST, was expanded to contain eight disk volumes. Figure 108 on page 179 shows the ISMF panel, option 6.1 using the LISTVOL line command, to display the assigned volumes.

Table 30. Test Case 5 - Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDBTEST	RV1CU1 RV1CU2 RV1CU3 RV2CU2 RV2CU3 RV3CU1 RV3CU2 RV3CU3

```

                                VOLUME LIST
Command ==>                                Scroll ==>HALF
                                           Entries 1-8 of 8
Enter Line Operators below:                Data Columns 3-8

  LINE      VOLUME FREE    %   ALLOC   FRAG   LARGEST   FREE
  OPERATOR  SERIAL SPACE  FREE  SPACE   INDEX  EXTENT   EXTENTS
  --- (1) --- - (2) -- -- (3) -- (4) - -- (5) -- - (6) - -- (7) -- -- (8) --
              RV1CU1 2764251   99   7249    0  2763200    1
              RV1CU2 2764251   99   7249    0  2763200    1
              RV1CU3 2764251   99   7249    0  2763200    1
              RV2CU2 2764251   99   7249    0  2763200    1
              RV2CU3 2764251   99   7249    0  2763200    1
              RV3CU1 2764251   99   7249    0  2763200    1
              RV3CU2 2764251   99   7249    0  2763200    1
              RV3CU3 2764251   99   7249    0  2763200    1
  -----
                                BOTTOM OF DATA
  -----

```

Figure 108. Test Case 5 - ISMF Volume List Display

A.6.2 ACS Routines

The Storage, Management Class, and Storage Group routines from test case 3 were used for this test case.

A.6.3 DB2 Definitions

The same CREATE STOGROUP SQL statement used in test case 3 was used here to allow SMS to control the table space allocation by specifying the parameter VOLUMES("*"), this is Figure 100 on page 172.

The CREATE DATABASE statement is shown in Figure 109 on page 179.

```

CREATE DATABASE BPAOLOR1
  STOGROUP SGRV1CU0
  BUFFERPOOL BP0
  CCSID EBCDIC

```

Figure 109. Test Case 5 - CREATE DATABASE

The CREATE TABLESPACE statement in Figure 110 on page 180, is an extract of the SQL code used.

```

CREATE TABLESPACE PART1
  IN BPAOLR1
  USING STOGROUP SGRV1CU0
    PRIQTY 20
    SECQTY 20
    ERASE NO

  NUMPARTS 8
    (PART 1 USING STOGROUP SGRV1CU0
      PRIQTY 1200000
      SECQTY 1200000,

      PART 2 USING STOGROUP SGRV1CU0
      PRIQTY 1200000
      SECQTY 1200000,

      .....

      PART 8 USING STOGROUP SGRV1CU0
      PRIQTY 1200000
      SECQTY 1200000) .....

```

Figure 110. Test Case 5 - CREATE TABLESPACE Extract

A.6.4 Data Set Allocation Results

The following results were displayed once the table space was allocated. Figure 111 on page 180 shows the ISPF data set list display. From this figure we can see that one table space partition was allocated per disk volume, ensuring an optimum spread for performance across all the RVA's. Figure 112 on page 181, shows the ISMF Storage Group volume display panel.

It can be seen that SMS has evenly distributed the data sets on each of the available volumes assigned to the storage group. This was achieved by using standard SMS volume selection criteria. There is no guarantee that this would occur for every allocation, given this scenario.

```

Menu Options View Utilities Compilers Help
-----
DSLIST - Data Sets Matching DB2D                               Row 1 of 8
Enter "/" to select action                                     Message      Volume
-----
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A005                        RV1CU1
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A003                        RV1CU2
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A001                        RV1CU3
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A008                        RV2CU2
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A004                        RV2CU3
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A007                        RV3CU1
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A002                        RV3CU2
DB2D.DSNDBD.BPAOLR1.PART1.I0001.A006                        RV3CU3

```

Figure 111. Test Case 5 - ISPF Data Set List of Table Space Partitions

```

                                VOLUME LIST
Command ===>                                Scroll ===> HALF
                                           Entries 1-8 of 8
Enter Line Operators below:                Data Columns 3-8 of 40

  LINE      VOLUME FREE   %   ALLOC   FRAG   LARGEST   FREE
OPERATOR   SERIAL SPACE  FREE SPACE  INDEX  EXTENT   EXTENTIS
--- (1) --- - (2) -- -- (3) -- (4) - -- (5) -- - (6) - -- (7) -- -- (8) --
          RV1CU1 1380300   50 1391200   0 1379525   2
          RV1CU2 1380300   50 1391200   0 1379525   2
          RV1CU3 1380300   50 1391200   0 1379525   2
          RV2CU2 1380300   50 1391200   0 1379525   2
          RV2CU3 1380300   50 1391200   0 1379525   2
          RV3CU1 1380300   50 1391200   0 1379525   2
          RV3CU2 1380300   50 1391200   0 1379525   2
          RV3CU3 1380300   50 1391200   0 1379525   2
-----
                                BOTTOM OF DATA
  
```

Figure 112. Test Case 5 - ISMF Storage Group Volume Display

Figure 113 on page 181 shows an extract from an IDCAMS LISTCAT display of one of the Table space partitions defined in this test, showing the SMS attributes, and the space allocation in cylinders:

```

CLUSTER ----- DB2D.DSNUDB.BPAOLOR1.PART1.I0001.A001
IN-CAT --- UCAT.VSEBOX09
HISTORY
  DATASET-OWNER-----HAIMO      CREATION-----1999.048
SMSDATA
  STORAGECLASS ---SCDBTEST      MANAGEMENTCLASS--MCDB21
  DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
ALLOCATION
  SPACE-TYPE-----CYLINDER      HI-A-RBA-----1229045760
  SPACE-PRI-----1667           HI-U-RBA-----1474560
  SPACE-SEC-----1667
VOLUME
  VOLSER-----RV1CU3
  
```

Figure 113. Test Case 5 - IDCAMS LISTCAT Display Extract

A.7 Partitioned Table Spaces Using SMS, User Distribution

Test case 6 illustrates how it is possible to define partitioned table spaces, ensuring SMS maps each separate partition to a separate Storage Group.

The aim, as in the previous test case, is to obtain the benefits of DB2 parallelism, but using the naming convention of the Table spaces to assign partitions to each Storage Group. This example defines two partitioned table spaces, each with four partitions using an existing naming conventions.

A.7.1 Create Storage Groups

Eight Storage Groups are allocated for the purpose of this test. One disk volume is assigned to each Storage Group (SGDBA001 - 4 and SGDBB001 - 4). Table 31

on page 182 shows the volume distribution. Figure 114 on page 182 shows the ISMF panel Storage Group list displaying the eight Storage Groups.

Table 31. Test Case 6 - Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDBA001	RV1CU0
SGDBA002	RV1CU2
SGDBA003	RV2CU0
SGDBA004	RV2CU1
SGDBB001	RV2CU3
SGDBB002	RV3CU0
SGDBB003	RV3CU2
SGDBB004	RV3CU3

```

                                STORAGE GROUP LIST
Command ==>>                                Scroll ==>> HALF
                                                Entries 1-8 of 8
                                                Data Columns 3-7 of 40

CDS Name : SMS.SCDS1.SCDS

Enter Line Operators below:

      LINE      STORGRP  SG          VIO      VIO  AUTO      MIGRATE SYSTEM
      OPERATOR  NAME      TYPE          MAXSIZE  UNIT  MIGRATE  OR SYS GROUP
      --- (1) --- -- (2) --- ----- (3) ----- -- (4) -- (5) - -- (6) --- ----- (7) -----
              SGDBA001 POOL          -----  ---- NO          -----
              SGDBA002 POOL          -----  ---- NO          -----
              SGDBA003 POOL          -----  ---- NO          -----
              SGDBA004 POOL          -----  ---- NO          -----
              SGDBB001 POOL          -----  ---- NO          -----
              SGDBB002 POOL          -----  ---- NO          -----
              SGDBB003 POOL          -----  ---- NO          -----
              SGDBB004 POOL          -----  ---- NO          -----
      -----  -----  -----  BOTTOM OF DATA -----  -----

```

Figure 114. Test Case 6 - ISMF Storage Group List

A.7.2 ACS Routines

The following amendments were applied to the ACS routines using the ISMF edit facility (option 7.1):

1. Data Class was not used.
2. A Storage Class, SCDBFAST, was defined for partitions. A *null* Storage Class was given to all other data set allocations.
3. A Management Class, MCDB20 was defined for partitions.
4. Code was added to the ACS source to allow the selection of the newly defined Storage Groups, based upon the naming convention of the table spaces; &DSN(4) is the variable used to identify the contents of the fourth level qualifier of the data set being allocated. Figure 115 on page 183 shows the ACS routine code extract for the Storage Group allocation:

```

/*****
/* STORAGE GROUP SELECTION ROUTINE      */
/*****
SELECT
  WHEN ((&DSN(4) = 'ALPHA')
    AND (&LLQ EQ 'A001'))
    SET &STORGRP = 'SGDBA001'
  WHEN ((&DSN(4) = 'ALPHA')
    AND (&LLQ EQ 'A002'))
    SET &STORGRP = 'SGDBA002'
  WHEN ((&DSN(4) = 'ALPHA')
    AND (&LLQ EQ 'A003'))
    SET &STORGRP = 'SGDBA003'
  WHEN ((&DSN(4) = 'ALPHA')
    AND (&LLQ EQ 'A004'))
    SET &STORGRP = 'SGDBA004'
  WHEN ((&DSN(4) = 'BETA')
    AND (&LLQ EQ 'A001'))
    SET &STORGRP = 'SGDBB001'
  WHEN ((&DSN(4) = 'BETA')
    AND (&LLQ EQ 'A002'))
    SET &STORGRP = 'SGDBB002'
  WHEN ((&DSN(4) = 'BETA')
    AND (&LLQ EQ 'A003'))
    SET &STORGRP = 'SGDBB003'
  WHEN ((&DSN(4) = 'BETA')
    AND (&LLQ EQ 'A004'))
    SET &STORGRP = 'SGDBB004'
END

```

Figure 115. Test Case 6 - Storage Group ACS Routine Extract

A.7.3 DB2 Definitions

Two STOGROUPs were created in DB2P, (SGDBA000 and SGDBB000). Both STOGROUPs have VOLUMES(""), as in example of Figure 100 on page 172.

DATABASE BPAOLOR1 is created using STOGROUP SGDBA000.

Two TABLESPACEs are created. Figure 116 on page 184 shows an extract of the SQL statements used for this purposes.

A.7.4 Data Set Allocation Results

A number of test cases were built using ISMF, option 7.4, prior to updating the active configuration to prove the validity of the ACS routines. All produced successful results.

The two partitioned data sets were then allocated. Figure 117 on page 184 is a view of all data sets with a pattern name of '**DB2P.DSNDBD.BPAOLOR1.****', using an extract of ISMF option 2. The output shows each partition on a separate disk volume. This is the desired objective.

```

CREATE TABLESPACE ALPHA
  IN BPAOLOR1
  USING STOGROUP SGDBA000
  ....
  Numparts 4

  (PART 1 USING STOGROUP SGDBA000
  ....

CREATE TABLESPACE BETA
  IN BPAOLOR1
  USING STOGROUP SGDBB000
  ....
  Numparts 4

  (PART 1 USING STOGROUP SGDBB000
  ....

```

Figure 116. Test Case 6 - CREATE TABLESPACE Extract

Command ==>

Scroll ==> CSR

Entries 1-7 of 8

Enter Line Operators below: Data Columns 3-5 (and 17)

LINE OPERATOR	DATA SET NAME	ALLOC SPACE	ALLOC USED	% NOT USED	VOLUME SERIAL
--- (1) ---	----- (2) -----	-- (3) --	-- (4) --	- (5) -	--- (17) --
	DB2P.DSNDBD.BPAOLOR1.ALPHA. I0001.A001	346126	1440	99	RV1CU0
	DB2P.DSNDBD.BPAOLOR1.ALPHA. I0001.A002	346126	1440	99	RV1CU2
	DB2P.DSNDBD.BPAOLOR1.ALPHA. I0001.A003	346126	1440	99	RV2CU0
	DB2P.DSNDBD.BPAOLOR1.ALPHA. I0001.A004	346126	1440	99	RV2CU1
	DB2P.DSNDBD.BPAOLOR1.BETA. I0001.A001	461502	1440	99	RV2CU3
	DB2P.DSNDBD.BPAOLOR1.BETA. I0001.A002	461502	1440	99	RV3CU0
	DB2P.DSNDBD.BPAOLOR1.BETA. I0001.A003	461502	1440	99	RV3CU2
	DB2P.DSNDBD.BPAOLOR1.BETA. I0001.A004	461502	1440	99	RV3CU3

Figure 117. Test Case 6 - ISMF Data Set List Extract

Appendix B. Test Cases for DB2 Recovery Data Sets

This appendix shows a selection of test cases generated during the writing of this publication for the definition of DB2 recovery data sets. The following tests are documented:

1. Allocation of BSDS and active logs using SMS.
2. Allocation of archive logs using SMS.
3. Allocation of image copies using SMS.

All tests were performed under laboratory conditions, and are presented as examples of the options and methodology used to achieve the end results.

B.1 BSDS and Active Logs

This test case illustrates the allocation of the BSDS and active logs, and is based on the following assumptions:

- There are three DB2 subsystems: DB2D, DB2T, and DB2P.
- The naming standard for the data sets is shown in 3.6.1, “Bootstrap Data Sets” on page 17 and in 3.6.2, “Active Logs” on page 18, using the subsystem identifier as high level qualifier.
- Storage Class, Management Class and Storage Group constructs are created with the criteria as defined in 7.1, “SMS Examples for DB2 Recovery Data Sets” on page 63.
- The logs and the BSDSs for the DB2P system are allocated on SMS Storage Group SGDB2PLG. For DB2D and DB2T they are allocated on SGDBACTL.
- The DB2P subsystem needs the allocation shown in Figure 39 on page 115.
- The DB2D and DB2T subsystems have little performance and availability requirements. All their data sets will be allocated on one volume.

B.1.1 SMS Storage Class

Storage Class SCDBACTL was defined for all BSDSs and active log data sets of the three subsystems.

Once allocated, these data sets are rarely redefined, and they require high performance and availability, so GUARANTEED SPACE is employed to position them on specific disk volumes. Figure 118 on page 186 shows panel two of ISMF option 5.3, to display this feature.

Figure 119 on page 186 shows an extract of the Storage Class ACS routine used to assign the BSDS and active log data sets.

```

                                STORAGE CLASS DEFINE                                Page 2 of 2
Command ==>

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Class Name  : SCDBACTL

To DEFINE Storage Class, Specify:

Guaranteed Space . . . . . Y                (Y or N)
Guaranteed Synchronous Write . . . N        (Y or N)
CF Cache Set Name . . . . .                (up to 8 chars or blank)
CF Direct Weight . . . . .                (1 to 11 or blank)
CF Sequential Weight . . . . .            (1 to 11 or blank)

Use ENTER to Perform Verification; Use UP Command to View previous Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 118. ISMF Storage Class Definition for BSDS and Active Logs

```

/*****/
/* STORAGE CLASS                               */
/* FILTLIST DEFINITIONS                         */
/*****/

FILTLIST DBSYS    INCLUDE (DB2*.BSDS*.**,
                          DB2*.LOGCOPY*.DS*)

/*****/
/* SELECTION ROUTINE FOR DB2 BSDS & ACTIVE LOGS */
/*****/
SELECT
  WHEN (&DSN = &DBSYS)
    SET &STORCLAS = 'SCDBACTL'
  OTHERWISE SET &STORCLAS = ''
END

```

Figure 119. Storage Class Routine Extract for BSDS and Active Logs

B.1.2 SMS Management Class

One Management Class, MCDBACTL, was defined for all BSDS and active log data sets of the three subsystems.

Figure 120 on page 187 shows an extract of a Management Class ACS routine to handle BSDS and active log data sets.


```

/*****/
/* MANAGEMENT CLASS */
/* FILTLIST DEFINITIONS */
/*****/

FILTLIST ACTLOG INCLUDE(DB2*.BSDS*.**,
                        DB2*.LOGCOPY*.DS*)

/*****/
/* SELECTION ROUTINE FOR BSDS & ACTIVE LOGS */
/*****/

      IF &DSN EQ &ACTLOG
      THEN DO
          SET &MGMTCLAS = 'MCDBACTL'
          EXIT
      END

```

Figure 120. Management Class Routine Extract for BSDS and Active Logs

B.1.3 Storage Group

Storage Group SGDB2PLG was defined for use by the BSDS and active log data sets of the DB2P subsystem. Five disk volumes were defined to this Storage Group: RV1CU1, RV2CU1 and RV3CU1 for the active logs; and RV2CU3 and RV3CU3 for the BSDS's.

Storage Group SGDBACTL was defined for use by the BSDS and active log data sets of the DB2D and DB2T subsystems. Two disk volumes were defined to this storage group: RV1CU3 and RV2CU2.

A disk volume summary is shown in Table 32 on page 187.

Table 32. BSDS and Active Logs - Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDB2PLG	RV1CU1 RV2CU1 RV3CU1 RV2CU3 RV3CU3
SGDBACTL	RV1CU3 RV2CU2

Figure 121 on page 188 shows an extract of the Storage Group ACS routine.

```

/*****
/* STORAGE GROUP                               */
/* SELECTION ROUTINE FOR DB2 BSDS & ACTIVE LOGS */
/*****

SELECT
  WHEN ((&STORCLAS = 'SCDBACTL')
        AND (&DSN(1) = 'DB2D'))
    SET &STORGRP = 'SGDBACTL'
  WHEN ((&STORCLAS = 'SCDBACTL')
        AND (&DSN(1) = 'DB2T'))
    SET &STORGRP = 'SGDBACTL'
  WHEN ((&STORCLAS = 'SCDBACTL')
        AND (&DSN(1) = 'DB2P'))
    SET &STORGRP = 'SGDB2PLG'

END

```

Figure 121. Storage Group Routine Extract for BSDS and Active Logs

B.1.4 ISMF Test Cases

Data set name pattern **DB2P.BSDS01.DATA** was tested against the active SMS configuration. Figure 122 on page 188 shows the output from the ISMF test case results:

```

***** Top of Data *****
                        ACS TESTING RESULTS

CDS NAME           : ACTIVE
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY  : PAOLOR3.JCL.CNTL

ACS TEST
MEMBER           EXIT CODE  RESULTS
-----
DESCRIPTION: DB2P.BSDS01.DATA
SMSTEST1                0 DC = NULL VALUE ASSIGNED
                        0 SC = NULL VALUE ASSIGNED
NOTE: MC AND SG NOT EXECUTED WHEN ACS READ/WRITE VARIABLE STORCLAS = '

ACS TESTING RC:  00

```

Figure 122. ISMF Test Result for BSDS (1)

The ACS routines assigned a *null* Storage Class as expected, thus making the data set non-SMS managed.

Figure 123 on page 189 shows the same data set name pattern tested against the **updated** SCDS. As can be seen on this occasion, the data set acquires a valid Storage Class and is therefore eligible for SMS management. Valid Management Class and Storage Group attributes are also assigned:

```

                                ACS TESTING RESULTS

CDS NAME           : SMS.SCDS1.SCDS
ACS ROUTINE TYPES: DC SC MC SG
ACS TEST LIBRARY  : PAOLOR3.JCL.CNTL

ACS TEST
MEMBER           EXIT CODE  RESULTS
-----
DESCRIPTION: DB2P.BSDS01.DATA
SMSTEST1         0 DC = NULL VALUE ASSIGNED
                  0 SC = SCDBACTL
                  0 MC = MCDBACTL
                  0 SG = SGDB2PLG

ACS TESTING RC: 00

```

Figure 123. ISMF Test Result for BSDS (2)

B.1.5 Data Set Allocation Results

Following the activation of the new SMS configuration, a number of data sets were defined. Figure 124 on page 189 shows an extract of the installation supplied IDCAMS parameters used to define two BSDSs.

```

DEFINE CLUSTER           -
( NAME (DB2P.BSDS01)    -
  VOLUMES (RV2CU3)     -
  REUSE                 -
  SHAREOPTIONS (2 3)   -
DATA                   -
( NAME (DB2P.BSDS01.DATA) -
  RECORDS (180 20)     -
  RECORDSIZE (4089 4089) -
  CONTROLINTERVALSIZE (4096) -
  FREESPACE (0 20)    -
  KEYS (4 0) )        -
INDEX                  -
( NAME (DB2P.BSDS01.INDEX) -
  RECORDS (5 5)       -
  CONTROLINTERVALSIZE (1024) )

```

Figure 124. IDCAMS Definition Extract for BSDS

Figure 125 on page 190 shows the ISPF data set list, displaying the BSDSs created in this test, and the volumes on which they were allocated, using the attributes assigned by SMS. By using the GUARANTEED SPACE parameter in the Storage Class definition, the data sets were positioned specifically on different volumes on different RVAs to enhance integrity.

```

Menu Options View Utilities Compilers Help
-----
DSLIST - Data Sets Matching DB2P.BSDS*                               Row 1 of 6
Command ==>>>                                                    Scroll ==>> CSR

Command - Enter "/" to select action                                Message                                Volume
-----
      DB2P.BSDS01                                                *VSAM*
      DB2P.BSDS01.DATA                                           RV2CU3
      DB2P.BSDS01.INDEX                                          RV2CU3
      DB2P.BSDS02                                                *VSAM*
      DB2P.BSDS02.DATA                                           RV3CU3
      DB2P.BSDS02.INDEX                                          RV3CU3
***** End of Data Set list *****

```

Figure 125. ISPF Data Set List of BSDS's

Six active log data sets were also defined. Figure 126 on page 190 shows an extract of SYSPRINT messages from the output of the IDCAMS definition batch job used for the allocation.

Figure 127 on page 191 shows an ISPF data set list to display all data sets with a pattern name of **DB2P.LOG***.

```

IDCAMS  SYSTEM SERVICES                                           TIME: 13:38:14

DEFINE CLUSTER -
( NAME (DB2P.LOGCOPY1.DS01) -
  VOLUMES (RV1CU1)          -
  REUSE                      -
  RECORDS (250000)          -
  LLINEAR )
DATA
( NAME (DB2P.LOGCOPY1.DS01.DATA) -
)
IDC0508I DATA ALLOCATION STATUS FOR VOLUME RV1CU1 IS 0
IDC0181I STORAGECLASS USED IS SCDBACTL
IDC0181I MANAGEMENTCLASS USED IS MCDBACTL
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

```

Figure 126. SYSPRINT Messages Extract for Active Log IDCAMS Definition

```

Menu Options View Utilities Compilers Help
-----
DSLIST - Data Sets Matching DB2P.LOG*                               Row 1 of 12
Command ==>>>                                                    Scroll ==>> CSR

Command - Enter "/" to select action                                Message                                Volume
-----
      DB2P.LOGCOPY1.DS01                                           *VSAM*
      DB2P.LOGCOPY1.DS01.DATA                                       RV1CU1
      DB2P.LOGCOPY1.DS02                                           *VSAM*
      DB2P.LOGCOPY1.DS02.DATA                                       RV2CU1
      DB2P.LOGCOPY1.DS03                                           *VSAM*
      DB2P.LOGCOPY1.DS03.DATA                                       RV3CU1
      DB2P.LOGCOPY2.DS01                                           *VSAM*
      DB2P.LOGCOPY2.DS01.DATA                                       RV2CU1
      DB2P.LOGCOPY2.DS02                                           *VSAM*
      DB2P.LOGCOPY2.DS02.DATA                                       RV3CU1
      DB2P.LOGCOPY2.DS03                                           *VSAM*
      DB2P.LOGCOPY2.DS03.DATA                                       RV1CU1
***** End of Data Set list *****

```

Figure 127. ISPF Data Set List of Active Logs

It can be seen from the above the list, that the use of the GUARANTEED SPACE parameter has allowed the even distribution of the six files across the three designated volumes, ensuring that optimum performance and availability are provided.

B.2 Archive Logs

This second test case illustrates the allocation of archive logs, and is based on the following assumptions:

- There are three DB2 subsystems: DB2D, DB2T, and DB2P.
- The naming standard for the data sets is shown in 3.6.3, “Archive Logs” on page 19, using as high level qualifier the subsystem identifier.
- Storage Class, Management Class and Storage Group constructs are created with the criteria as defined in 7.1, “SMS Examples for DB2 Recovery Data Sets” on page 63.
- The ACS routines from the example in B.1, “BSDS and Active Logs” on page 185 will be extended to support the archive log data sets.

B.2.1 Storage Class

One Storage Class, SCDBARCH was added to the SMS configuration for all archive log data sets of the three subsystems.

Figure 128 on page 192 shows an extract of the extended Storage Class ACS routine used in the previous test case, now incorporating the archive logs:

```

/*****/
/* STORAGE CLASS                               */
/* FILTLIST DEFINITIONS                         */
/*****/

FILTLIST DBSYS    INCLUDE (DB2*.BSDS*.**,
                          DB2*.LOGCOPY*.DS*)

FILTLIST DBARCH  INCLUDE (DB2*.ARCHLOG*.**)

/*****/
/* SELECTION ROUTINE FOR DB2 RECOVERY DATA SETS */
/*****/
SELECT
  WHEN (&DSN = &DBSYS)
    SET &STORCLAS = 'SCDBACTL'
  WHEN (&DSN = &DBARCH)
    SET &STORCLAS = 'SCDBARCH'
  OTHERWISE SET &STORCLAS = ''
END

```

Figure 128. Storage Class Routine Incorporating Archive Logs

B.2.2 Management Class

Two Management Classes, MCDVICM and MCDVLV2, were added to the SMS configuration to separate primary and secondary archive log data sets of the three subsystems with different criteria.

Figure 129 on page 192 shows an extract of the extended Management Class ACS routine used in the previous test case, now incorporating the archive logs:

```

/*****/
/* MANAGEMENT CLASS                             */
/* FILTLIST DEFINITIONS                         */
/*****/

FILTLIST ACTLOG  INCLUDE (DB2*.BSDS*.**,
                          DB2*.LOGCOPY*.DS*)

/*****/
/* SELECTION ROUTINE FOR DB2 RECOVERY DATA SETS */
/*****/

  IF &DSN EQ &ACTLOG
    THEN DO
      SET &MGMTCLAS = 'MCDVACTL'
      EXIT
    END

  IF (&DSN(2) EQ 'ARCHLOG1')
    THEN DO
      SET &MGMTCLAS = 'MCDVICM'
    END
  ELSE DO
    SET &MGMTCLAS = 'MCDVLV2'
  END
END

```

Figure 129. Management Class Routine Incorporating Archive Logs

B.2.3 Storage Group

One Storage Group, SGDBARCH, was added for all archive log data sets of the three DB2 subsystems. Three disk volumes, RV1CU0, RV2CU0, and RV3CU0, were defined to the Storage Group as shown in Table 33 on page 193.

Figure 130 on page 193 shows an extract of the extended Storage Group ACS routine used in the previous test case, now incorporating the archive logs.

Table 33. Archive Logs—Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDBARCH	RV1CU0 RV2CU0 RV3CU0

```

/*****
/* STORAGE GROUP                               */
/* SELECTION ROUTINE FOR DB2 RECOVERY DATA SETS */
/*****
SELECT
      WHEN ((&STORCLAS = 'SCDBACTL')
            AND (&DSN(1) = 'DB2D'))
        SET &STORGRP = 'SGDBACTL'
      WHEN ((&STORCLAS = 'SCDBACTL')
            AND (&DSN(1) = 'DB2T'))
        SET &STORGRP = 'SGDBACTL'
      WHEN ((&STORCLAS = 'SCDBACTL')
            AND (&DSN(1) = 'DB2P'))
        SET &STORGRP = 'SGDB2PLG'
      WHEN (&STORCLAS = 'SCDBARCH')
        SET &STORGRP = 'SGDBARCH'
END

```

Figure 130. Storage Group Routine Incorporating Archive Logs

B.2.4 Data Set Allocation Results

ISMF Test cases were built and executed against both the updated SCDS and ACDS, producing the expected results.

After the new SMS configuration was activated, the following MVS command was issued using SDSF to generate new archive logs:

```
=DB2P ARCHIVE LOG MODE(QUIESCE)
```

Figure 131 on page 193 displays an extract of the subsequent MVS SYSLOG messages issued.

```

DSNJ003I =DB2P DSNJOFF3 FULL ARCHIVE LOG VOLUME 474
DSNAME=DB2P.ARCHLOG2.D99057.T1202900.A0000001, STARTRBA=000000000000,
ENDRBA=00000068AFFF, STARTTIME=B1B68D1C709B, ENDTIME=B1DDE7C34366,
UNIT=SYSALLDA, COPY2VOL=RV3CU0, VOLSPAN=00, CATLG=YES
DSNJ139I =DB2P LOG OFFLOAD TASK ENDED

```

Figure 131. SYSLOG Message Output Extract for Archive Logs

The four data sets created were allocated successfully on disk volumes assigned to Storage Group SGDBARCH. Figure 132 on page 194 shows an ISPF data set list to display all data sets with a pattern name of **DB2P.ARCH***:

```

Menu Options View Utilities Compilers Help
-----
DSLIST - Data Sets Matching DB2P.ARCH*                               Row 1 of 4
Command ==>                                                         Scroll ==> CSR

Command - Enter "/" to select action                                Message                                Volume
-----
DB2P.ARCHLOG1.D99057.T1202900.A0000001                            RV1CU0
DB2P.ARCHLOG1.D99057.T1202900.B0000001                            RV2CU0
DB2P.ARCHLOG2.D99057.T1202900.A0000001                            RV3CU0
DB2P.ARCHLOG2.D99057.T1202900.B0000001                            RV1CU0
***** End of Data Set list *****

```

Figure 132. ISPF Data Set List of Archive Logs

Although all data sets have been allocated to the same Storage Group, their Management Classes vary depending upon the naming pattern. Figure 133 on page 194 displays an IDCAMS LISTCAT extract of two of the data sets for comparison with the fields highlighted:

```

NONVSAM ----- DB2P.ARCHLOG1.D99057.T1202900.A0000001
IN-CAT --- UCAT.VSBOX09
HISTORY
DATASET-OWNER----- (NULL)      CREATION-----1999.057
RELEASE-----2          EXPIRATION-----2026.194
ACCOUNT- INFO----- (NULL)
SMSDATA
STORAGECLASS ---SCDBARCH      MANAGEMENTCLASS--MCDBICM
DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
VOLUMES
VOLSER-----RV1CU0          DEVTYPE-----X'3010200F'

NONVSAM ----- DB2P.ARCHLOG2.D99057.T1202900.A0000001
IN-CAT --- UCAT.VSBOX09
HISTORY
DATASET-OWNER----- (NULL)      CREATION-----1999.057
RELEASE-----2          EXPIRATION-----2026.194
ACCOUNT- INFO----- (NULL)
SMSDATA
STORAGECLASS ---SCDBARCH      MANAGEMENTCLASS--MCDBLV2
DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
VOLUMES
VOLSER-----RV3CU0          DEVTYPE-----X'3010200F'

```

Figure 133. IDCAMS LISTCAT of Management Class Comparison for Archive Logs

B.3 Image Copies

This third test case illustrates the allocation of a selection of image copies with varying criteria, and is based on the following assumptions:

- There are three DB2 subsystems: DB2D, DB2T, and DB2P.

- The naming standard for the data sets is shown in 3.8.5, “Image Copy Names” on page 24, using as a high level qualifier the subsystem identifier, followed by IC. This name includes three codes in the second qualifier:
 - Col 1 **P** or **S** to indicate primary or secondary copy
 - Col 2 **S** or **H** to indicate standard or critical copy
 - Col 3 **D, W** or **M** to indicate daily, weekly or monthly copy
- Storage Class, Management Class and Storage Group constructs are defined for all image copies on all subsystems as specified in 7.5, “Image Copies” on page 71.
 - Primary image copies are placed on SGDBIC or SGDBICH
 - Secondary image copies are placed on SGDBARCH
- The ACS routines from the example in B.2, “Archive Logs” on page 191 will be extended to support the image copy data sets.

B.3.1 Storage Class

Two additional Storage Classes were defined:

- SCDBIC for data sets with standard performance requirements
- SCDBICH for data sets with high performance requirements

Figure 128 on page 192 shows an extract of the Storage Class ACS routine used in the previous tests, now including the separation of image copies to provide different levels of performance, depending upon the naming convention:

```

/*****/
/* STORAGE CLASS                               */
/* FILTLIST DEFINITIONS                         */
/*****/

FILTLIST DBSYS    INCLUDE (DB2*.BSDS*.**,
                          DB2*.LOGCOPY*.DS*)

FILTLIST DBARCH   INCLUDE (DB2*.ARCHLOG*.**)

FILTLIST ICHIGH   INCLUDE (DB2%IC.PH*.T*.*.A*)

FILTLIST ICSTD    INCLUDE (DB2%IC.S*.T*.*.A*,
                          DB2%IC.%S*.T*.*.A*)

/*****/
/* SELECTION ROUTINE FOR DB2 RECOVERY DATA SETS */
/*****/
SELECT
  WHEN (&DSN = &DBSYS)
    SET &STORCLAS = 'SCDBACTL'
  WHEN (&DSN = &DBARCH)
    SET &STORCLAS = 'SCDBARCH'
  WHEN (&DSN = &ICSTD)
    SET &STORCLAS = 'SCDBIC'
  WHEN (&DSN = &ICHIGH)
    SET &STORCLAS = 'SCDBICH'
  OTHERWISE SET &STORCLAS = ''
END

```

Figure 134. Storage Class Routine Extract Incorporating Image Copies

B.3.2 Management Class

Two additional Management Classes were defined;

- MCDBICD for daily primary image copy data sets
- MCDBICW for weekly primary image copy data sets

Figure 120 on page 187 shows an extract of the Management Class ACS routine used in the previous tests, now including the two additional classes for availability, again depending upon the naming convention:

```
/* ***** */
/* MANAGEMENT CLASS */
/* FILTLIST DEFINITIONS */
/* ***** */

FILTLIST ACTLOG INCLUDE (DB2*.BSDS*.**,
                        DB2*.LOGCOPY*.DS*)

FILTLIST ICD INCLUDE (DB2%IC.P%D*.T*.*.A*)

FILTLIST ICW INCLUDE (DB2%IC.P%W*.T*.*.A*)

FILTLIST ICM INCLUDE (DB2%IC.P%M*.T*.*.A*)

/* ***** */
/* SELECTION ROUTINE FOR DB2 RECOVERY DATA SETS */
/* ***** */
IF &DSN EQ &ACTLOG
  THEN DO
    SET &MGMTCLAS = 'MCDBACTL'
  EXIT
END

IF &DSN EQ &ICD
  THEN DO
    SET &MGMTCLAS = 'MCDBICD'
  EXIT
END

IF &DSN EQ &ICW
  THEN DO
    SET &MGMTCLAS = 'MCDBICW'
  EXIT
END

IF (&DSN(2) EQ 'ARCHLOG1' OR &DSN EQ &ICM)
  THEN DO
    SET &MGMTCLAS = 'MCDBICM'
  END
ELSE DO
  SET &MGMTCLAS = 'MCDBLV2'
  END
END
```

Figure 135. Management Class Routine Extract Incorporating Image Copies

B.3.3 Storage Group

Two additional storage groups are defined to cater for image copy data sets, SGDBIC and SGDICH. Table 34 on page 197 shows the distribution of volumes across all the storage groups used in all three examples of this appendix.

Table 34. Image Copies—Storage Group Volumes

SMS STORAGE GROUP	VOLUMES
SGDB2PLG	RV1CU1 RV2CU1 RV3CU1 RV2CU3 RV3CU3
SGDBACTL	RV1CU3 RV2CU2
SGDBARCH	RV1CU0 RV2CU0 RV3CU0
SGDBIC	RV1CU2
SGDBICH	RV3CU2

Figure 121 on page 188 shows an extract of the Storage Group ACS routine to divide the data sets by use of the &STORCLAS and &MGMTCLAS variables:

```

/*****
/* STORAGE GROUP
/* SELECTION ROUTINE FOR DB2 RECOVERY DATA SETS
*****/
SELECT
    WHEN ((&STORCLAS = 'SCDBACTL')
        AND (&DSN(1) = 'DB2D'))
        SET &STORGRP = 'SGDBACTL'
    WHEN ((&STORCLAS = 'SCDBACTL')
        AND (&DSN(1) = 'DB2T'))
        SET &STORGRP = 'SGDBACTL'
    WHEN ((&STORCLAS = 'SCDBACTL')
        AND (&DSN(1) = 'DB2P'))
        SET &STORGRP = 'SGDB2PLG'
    WHEN ((&STORCLAS = 'SCDBARCH')
        OR (&MGMTCLAS = 'MCDBLV2'))
        SET &STORGRP = 'SGDBARCH'
    WHEN (&STORCLAS = 'SCDBIC')
        SET &STORGRP = 'SGDBIC'
    WHEN (&STORCLAS = 'SCDBICH')
        SET &STORGRP = 'SGDBICH'

END

```

Figure 136. Storage Group Routine Extract Incorporating Image Copies

B.3.4 Data Set Allocation Results

Again, ISMF test cases were built and executed against the updated SCDS and the ACDS, producing the expected results.

Following the activation of the new SMS configuration, a number of image copy data sets were allocated. Figure 137 on page 198 shows the JCL used in this exercise.

```

//IMAGCOPY EXEC DSNUPROC, PARM='DB2P, DSN8', COND=(4, LT)
//DSNTRACE DD   SYSOUT=*
//SYSCOPY1 DD   DSN=DB2PIC.PHD99060.T130000.DSN8S61P.A001,
//              UNIT=3390, DISP=(, CATLG, DELETE), SPACE=(4000, (20, 20))
//SYSCOPY2 DD   DSN=DB2PIC.SSD99060.T130000.DSN8S61P.A001,
//              UNIT=3390, DISP=(, CATLG, DELETE), SPACE=(4000, (20, 20))
//SYSCOPY3 DD   DSN=DB2PIC.PSW99060.T130000.DSN8S61R.A001,
//              UNIT=3390, DISP=(, CATLG, DELETE), SPACE=(4000, (20, 20))
//SYSCOPY4 DD   DSN=DB2PIC.SSW99060.T130000.DSN8S61R.A001,
//              UNIT=3390, DISP=(, CATLG, DELETE), SPACE=(4000, (20, 20))
//SYSCOPY5 DD   DSN=DB2PIC.PSM99060.T130000.DSN8S61S.A001,
//              UNIT=3390, DISP=(, CATLG, DELETE), SPACE=(4000, (20, 20))
//SYSCOPY6 DD   DSN=DB2PIC.SSM99060.T130000.DSN8S61S.A001,
//              UNIT=3390, DISP=(, CATLG, DELETE), SPACE=(4000, (20, 20))
//SYSIN      DD   *
                COPY TABLESPACE DSN8D61A.DSN8S61P COPYDDN(SYSCOPY1, SYSCOPY2)

                COPY TABLESPACE DSN8D61A.DSN8S61R COPYDDN(SYSCOPY3, SYSCOPY4)

                COPY TABLESPACE DSN8D61A.DSN8S61S COPYDDN(SYSCOPY5, SYSCOPY6)
/*

```

Figure 137. JCL for Image Copy Allocation

Figure 138 on page 198 shows an extract of the JES output messages generated from the above JCL:

```

IGD101I SMS ALLOCATED TO DDNAME (SYSCOPY )
        DSN (DB2PIC.PHD99060.T130000.DSN8S61P.A001      )
        STORCLAS (SCDBICH) MGMTICLAS (MCDBICD) DATACLAS (      )
        VOL SER NOS= RV3CU2
IGD101I SMS ALLOCATED TO DDNAME (SYSCOPY2)
        DSN (DB2PIC.SSD99060.T130000.DSN8S61P.A001      )
        STORCLAS (SCDBIC) MGMTICLAS (MCDBLV2) DATACLAS (      )
        VOL SER NOS= RV3CU0
IGD101I SMS ALLOCATED TO DDNAME (SYSCOPY3)
        DSN (DB2PIC.PSW99060.T130000.DSN8S61R.A001      )
        STORCLAS (SCDBIC) MGMTICLAS (MCDBICW) DATACLAS (      )
        VOL SER NOS= RV1CU2
IGD101I SMS ALLOCATED TO DDNAME (SYSCOPY4)
        DSN (DB2PIC.SSW99060.T130000.DSN8S61R.A001      )
        STORCLAS (SCDBIC) MGMTICLAS (MCDBLV2) DATACLAS (      )
        VOL SER NOS= RV1CU0
IGD101I SMS ALLOCATED TO DDNAME (SYSCOPY5)
        DSN (DB2PIC.PSM99060.T130000.DSN8S61S.A001      )
        STORCLAS (SCDBIC) MGMTICLAS (MCDBICM) DATACLAS (      )
        VOL SER NOS= RV1CU2
IGD101I SMS ALLOCATED TO DDNAME (SYSCOPY6)
        DSN (DB2PIC.SSM99060.T130000.DSN8S61S.A001      )
        STORCLAS (SCDBIC) MGMTICLAS (MCDBLV2) DATACLAS (      )
        VOL SER NOS= RV2CU0

```

Figure 138. Image Copy Allocation JES Output Messages

Figure 139 on page 199 shows an ISPF data set list, displaying the six image copies created in this test, and the volumes on which they were allocated, according to the SMS attributes assigned to them.

```

DSLIST - Data Sets Matching DB2PIC.*                               Row 1 of 6
Command ===>                                                    Scroll ===> CSR

Command - Enter "/" to select action                               Message                               Volume
-----
DB2PIC.PSW99060.T130000.DSN8S61R.A001                          RV1CU2
DB2PIC.SSW99060.T130000.DSN8S61R.A001                          RV1CU0
DB2PIC.PHD99060.T130000.DSN8S61P.A001                          RV3CU2
DB2PIC.SSD99060.T130000.DSN8S61P.A001                          RV3CU0
DB2PIC.PSM99060.T130000.DSN8S61S.A001                          RV1CU2
DB2PIC.SSM99060.T130000.DSN8S61S.A001                          RV2CU0
***** End of Data Set list *****

```

Figure 139. ISPF Data Set List of Image Copies

Figure 140 on page 199 shows an IDCAMS LISTCAT display of the primary and the secondary image copy created in this test to backup the table space DSN8S61P. The display shows their SMS attributes and volume allocation highlighted.

```

NONVSAM ----- DB2PIC.PHD99060.T130000.DSN8S61P.A001
IN-CAT --- UCAT.VSBOX09
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1999.055
  RELEASE-----2          EXPIRATION-----0000.000
  ACCOUNT-INFO----- (NULL)
SMSDATA
  STORAGECLASS ----SCDBICH  MANAGEMENTCLASS--MCDBICD
  DATACLASS ----- (NULL)    LBACKUP ---0000.000.0000
VOLUMES
  VOLSER-----RV3CU2    DEVTYPE-----X'3010200F'

NONVSAM ----- DB2PIC.SSD99060.T130000.DSN8S61P.A001
IN-CAT --- UCAT.VSBOX09
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1999.055
  RELEASE-----2          EXPIRATION-----0000.000
  ACCOUNT-INFO----- (NULL)
SMSDATA
  STORAGECLASS ----SCDBIC  MANAGEMENTCLASS--MCDBLV2
  DATACLASS ----- (NULL)    LBACKUP ---0000.000.0000
VOLUMES
  VOLSER-----RV3CU0    DEVTYPE-----X'3010200F'

```

Figure 140. IDCAMS LISTCAT Extract of Image Copy Data Sets

Appendix C. DB2 PM Accounting Trace Report

1 LOCATION: USIBMT6BOAPLX DB2 PERFORMANCE MONITOR (V5) PAGE: 1-1
 GROUP: BOAG ACCOUNTING TRACE - LONG REQUESTED FROM: 02/12/99 15:06:03.00
 MEMBER: NB22 TO: 02/12/99 15:43:47.00
 SUBSYSTEM: NB22 ACTUAL FROM: 02/12/99 15:43:45.28
 DB2 VERSION: V5

----- IDENTIFICATION -----
 ACCT TSTAMP: 02/12/99 15:43:45.28 PLANNAME: POCDRIVE WLM SCL: 'BLANK' CICS NET: N/A
 BEGIN TIME : 02/12/99 15:06:04.28 PROD ID : N/P CICS LUN: N/A
 END TIME : 02/12/99 15:43:45.28 PROD VER: N/P LUW NET: BOAG CICS INS: N/A
 REQUESTER : USIBMT6BOAPLX CORRNAME: RBQ05A LUW LUN: T6E3BOA
 MAINPACK : POCDRIVE CORRNMBR: 'BLANK' LUW INS: B1CBC859371E END USER: 'BLANK'
 PRMAUTH : HPQUSER CONNTYPE: TSO LUW SEQ: 1 TRANSACT: 'BLANK'
 ORIGAUTH : HPQUSER CONNECT : BATCH WS NAME : 'BLANK'

MVS ACCOUNTING DATA : RBQ05A
 RRSF ACCOUNTING TOKEN: N/A

ELAPSED TIME DISTRIBUTION		CLASS 2 TIME DISTRIBUTION	
APPL	>	CPU	> 37%
DB2	> 90%	NOTACC	> 53%
SUSP	> 10%	SUSP	> 10%

TIMES/EVENTS	APPL (CLASS 1)	DB2 (CLASS 2)	IFI (CLASS 5)	CLASS 3 SUSP.	ELAPSED TIME	EVENTS	HIGHLIGHTS
ELAPSED TIME	37:41.001054	37:40.386069	N/P	LOCK/LATCH	0.060799	2760	THREAD TYPE : ALLIED
CPU TIME	1:06:21.580844	1:06:21.549125	N/P	SYNCHRON. I/O	56.770913	19559	TERM.CONDITION: NORMAL
TCB	14:02.087183	14:02.055513	N/P	OTHER READ I/O	29:59.155952	143669	INVOKE REASON : DEALLOC
TCB-STPROC	0.000000	0.000000	N/A	OTHER WRTE I/O	0.000000	0	COMMITTS : 2
PAR.TASKS	52:19.493661	52:19.493612	N/A	SER.TASK SWTCH	0.002627	2	ROLLBACK : 0
SUSPEND TIME	N/A	30:55.990291	N/A	ARC.LOG(QUIES)	0.000000	0	INCREM.BINDS : 0
TCB	N/A	3:48.273531	N/A	ARC.LOG READ	0.000000	0	UPDATE/COMMIT : 0.00
PAR.TASKS	N/A	27:07.716760	N/A	DRAIN LOCK	0.000000	0	SYNCH I/O AVG. : 0.002903
NOT ACCOUNT.	N/A	19:50.057025	N/P	CLAIM RELEASE	0.000000	0	PROGRAMS : 0
DB2 ENT/EXIT	N/A	217	N/A	PAGE LATCH	0.000000	0	PARALLELISM : CP
EN/EX-STPROC	N/A	0	N/A	STORED PROC.	0.000000	0	
DCAPT.DESCR.	N/A	N/A	N/P	NOTIFY MSGS	0.000000	0	
LOG EXTRACT.	N/A	N/A	N/P	GLOBAL CONT.	0.000000	0	
				TOTAL CLASS 3	30:55.990291	165990	

SQL DML	TOTAL	SQL DCL	TOTAL	SQL DDL	CREATE	DROP	ALTER	LOCKING	TOTAL	DATA SHARING	TOTAL
SELECT	0	LOCK TABLE	0	TABLE	0	0	0	TIMEOUTS	0	GLB CONT (%)	0
INSERT	0	GRANT	0	TEMP TABLE	0	N/A	N/A	DEADLOCKS	0	FLS CONT (%)	0
UPDATE	0	REVOKE	0	INDEX	0	0	0	ESCAL. (SHAR)	0	LOCK REQUEST	0
DELETE	0	SET SQLID	0	TABLESPACE	0	0	0	ESCAL. (EXCL)	0	UNLOCK REQST	0
		SET H.VAR.	0	DATABASE	0	0	0	MAX.LCK HELD	25	CHANGE REQST	0
DESCRIBE	0	SET DEGREE	1	STOGROUP	0	0	0	LOCK REQUEST	103	LOCK - XES	18
DESC.TBL	0	SET RULES	0	SYNONYM	0	0	N/A	UNLOCK REQST	18	UNLOCK-XES	1
PREPARE	1	CONNECT 1	0	VIEW	0	0	N/A	QUERY REQST	0	CHANGE-XES	0
OPEN	1	CONNECT 2	1	ALIAS	0	0	N/A	CHANGE REQST	1	SUSP - IRLM	0
FETCH	101	SET CONNEC	0	PACKAGE	N/A	0	N/A	OTHER REQST	0	SUSP - YES	0
CLOSE	1	RELEASE	0					LOCK SUSP.	0	SUSP - FALSE	0
		CALL	0	TOTAL	0	0	0	LATCH SUSP.	0	INCOMP.LOCK	0
		ASSOC LOC.	0	RENAME TBL	0			OTHER SUSP.	1	NOTIFY SENT	1
DML-ALL	104	ALLOC CUR.	0	COMMENT ON	0			TOTAL SUSP.	1		
		DCL-ALL	2	LABEL ON	0						

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 ACCOUNTING TRACE - LONG

PAGE: 1-2
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:43:47.00
 ACTUAL FROM: 02/12/99 15:43:45.28

----- IDENTIFICATION -----

ACCT TSTAMP: 02/12/99 15:43:45.28 PLANNAME: POCDRIVE WLM SCL: 'BLANK' CICS NET: N/A
 BEGIN TIME : 02/12/99 15:06:04.28 PROD ID : N/P CICS LUN: N/A
 END TIME : 02/12/99 15:43:45.28 PROD VER: N/P LUW NET: BOAG CICS INS: N/A
 REQUESTER : USIBMT6BOAPLX CORRNAME: RBQ05A LUW LUN: T6E3BOA
 MAINPACK : POCDRIVE CORRNMBR: 'BLANK' LUW INS: B1CBC859371E END USER: 'BLANK'
 PRIMAUTH : HPQUSER CONNTYPE: TSO LUW SEQ: 1 TRANSACT: 'BLANK'
 ORIGAUTH : HPQUSER CONNECT : BATCH WS NAME : 'BLANK'

RID LIST	TOTAL	QUERY PARALLEL.	TOTAL	STORED PROC.	TOTAL	DRAIN/CLAIM	TOTAL	DATA CAPTURE	TOTAL
USED	0	MAXIMUM MEMBERS	N/P	CALL STMTS	0	DRAIN REQST	0	IFI CALLS	N/P
FAIL-NO STORAGE	0	MAXIMUM DEGREE	5	PROC. ABENDS	0	DRAIN FAILED	0	REC.CAPTURED	N/P
FAIL-LIMIT EXC.	0	GROUPS EXECUTED	1	CALL TIMEOUT	0	CLAIM REQST	88	LOG REC.READ	N/P
		RAN AS PLANNED	1	CALL REJECT	0	CLAIM FAILED	2	ROWS RETURN	N/P
		RAN REDUCED	0					RECORDS RET.	N/P
		ONE DB2 COOR=N	0					DATA DES.RET	N/P
		ONE DB2 ISOLAT	0					TABLES RET.	N/P
		SEQ - CURSOR	0					DESCRIBES	N/P
		SEQ - NO ESA	0						
		SEQ - NO BUF	0						
		SEQ - ENCL.SER	0						
		MEMB SKIPPED(%)	0						
		DISABLED BY RLF	NO						

SERVICE UNITS	CLASS 1	CLASS 2	OPTIMIZATION	TOTAL
CPU	7600249	7600188	REOPTIMIZATION	0
TCB	1607420	1607359	PREP_STMT_MATCH	0
TCB-STPROC	0	0	PREP_STMT_NO_MATCH	0
PAR.TASKS	5992829	5992829	IMPLICIT_PREPARES	0
			PREP_FROM_CACHE	0
			CACHE_LIMIT_EXCEED	0
			PREP_STMT_PURGED	0

----- RESOURCE LIMIT FACILITY -----

TYPE: N/P TABLE ID: N/P SERV.UNITS: N/P CPU SECONDS: 0.000000 MAX CPU SEC: N/P

BP0	TOTAL	BP2	TOTAL	BP4	TOTAL
BPOOL HIT RATIO (%)	100	BPOOL HIT RATIO (%)	0	BPOOL HIT RATIO (%)	50
GETPAGES	107	GETPAGES	5835348	GETPAGES	300350
BUFFER UPDATES	0	BUFFER UPDATES	0	BUFFER UPDATES	0
SYNCHRONOUS WRITE	0	SYNCHRONOUS WRITE	0	SYNCHRONOUS WRITE	0
SYNCHRONOUS READ	0	SYNCHRONOUS READ	18796	SYNCHRONOUS READ	754
SEQ. PREFETCH REQS	0	SEQ. PREFETCH REQS	163939	SEQ. PREFETCH REQS	702
LIST PREFETCH REQS	0	LIST PREFETCH REQS	0	LIST PREFETCH REQS	0
DYN. PREFETCH REQS	0	DYN. PREFETCH REQS	22121	DYN. PREFETCH REQS	3944
PAGES READ ASYNCHR.	0	PAGES READ ASYNCHR.	5795249	PAGES READ ASYNCHR.	148634
HPOOL WRITES	0	HPOOL WRITES	0	HPOOL WRITES	0
HPOOL WRITES-FAILED	0	HPOOL WRITES-FAILED	0	HPOOL WRITES-FAILED	0
PAGES READ ASYN-HPOOL	0	PAGES READ ASYN-HPOOL	0	PAGES READ ASYN-HPOOL	0
HPOOL READS	0	HPOOL READS	0	HPOOL READS	0
HPOOL READS-FAILED	0	HPOOL READS-FAILED	0	HPOOL READS-FAILED	0

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 ACCOUNTING TRACE - LONG

PAGE: 1-3
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:43:47.00
 ACTUAL FROM: 02/12/99 15:43:45.28

---- IDENTIFICATION -----

ACCT TSTAMP: 02/12/99 15:43:45.28	PLANNAME: POCDRIVE	WLM SCL: 'BLANK'	CICS NET: N/A
BEGIN TIME : 02/12/99 15:06:04.28	PROD ID : N/P		CICS LUN: N/A
END TIME : 02/12/99 15:43:45.28	PROD VER: N/P	LUW NET: BOAG	CICS INS: N/A
REQUESTER : USIBMT6BOAPLX	CORRNAME: RBQ05A	LUW LUN: T6E3BOA	
MAINPACK : POCDRIVE	CORRNMBR: 'BLANK'	LUW INS: B1CBC859371E	END USER: 'BLANK'
PRMAUTH : HPQUSER	CONNTYPE: TSO	LUW SEQ: 1	TRANSACT: 'BLANK'
ORIGAUTH : HPQUSER	CONNECT : BATCH		WS NAME : 'BLANK'

BP5	TOTAL	TOT4K	TOTAL
-----	-----	-----	-----
BPOOL HIT RATIO (%)	0	BPOOL HIT RATIO (%)	2
GETPAGES	70	GETPAGES	6135875
BUFFER UPDATES	48	BUFFER UPDATES	48
SYNCHRONOUS WRITE	0	SYNCHRONOUS WRITE	0
SYNCHRONOUS READ	9	SYNCHRONOUS READ	19559
SEQ. PREFETCH REQS	8	SEQ. PREFETCH REQS	164649
LIST PREFETCH REQS	0	LIST PREFETCH REQS	0
DYN. PREFETCH REQS	0	DYN. PREFETCH REQS	26065
PAGES READ ASYNCHR.	64	PAGES READ ASYNCHR.	5943947
HPOOL WRITES	0	HPOOL WRITES	0
HPOOL WRITES-FAILED	0	HPOOL WRITES-FAILED	0
PAGES READ ASYN-HPOOL	0	PAGES READ ASYN-HPOOL	0
HPOOL READS	0	HPOOL READS	0
HPOOL READS-FAILED	0	HPOOL READS-FAILED	0

Appendix D. DB2 PM Statistics Report

```

1  LOCATION: USIBMT6BOAPLX          DB2 PERFORMANCE MONITOR (V5)          PAGE: 2-1
    GROUP: BOAG                      STATISTICS REPORT - LONG          REQUESTED FROM: 02/12/99 15:06:03.00
    MEMBER: NB22                      TO: 02/12/99 15:44:05.00
    SUBSYSTEM: NB22                   INTERVAL FROM: 02/12/99 15:09:49.64
    DB2 VERSION: V5                   SCOPE: MEMBER                      TO: 02/12/99 15:43:45.80
  
```

----- HIGHLIGHTS -----

```

INTERVAL START : 02/12/99 15:09:49.64  SAMPLING START: 02/12/99 15:09:49.64  TOTAL THREADS      :    0.00
INTERVAL END   : 02/12/99 15:43:45.80  SAMPLING END   : 02/12/99 15:43:45.80  TOTAL COMMITS      :    2.00
INTERVAL ELAPSED:      33:56.157107    OUTAGE ELAPSED:      0.000000    DATA SHARING MEMBER:  N/A
  
```

SQL DML	QUANTITY	/MINUTE	/THREAD	/COMMIT	SQL DCL	QUANTITY	/MINUTE	/THREAD	/COMMIT
SELECT	0.00	0.00	N/C	0.00	LOCK TABLE	0.00	0.00	N/C	0.00
INSERT	0.00	0.00	N/C	0.00	GRANT	0.00	0.00	N/C	0.00
UPDATE	0.00	0.00	N/C	0.00	REVOKE	0.00	0.00	N/C	0.00
DELETE	0.00	0.00	N/C	0.00	SET HOST VARIABLE	0.00	0.00	N/C	0.00
					SET CURRENT SQLID	0.00	0.00	N/C	0.00
PREPARE	0.00	0.00	N/C	0.00	SET CURRENT DEGREE	0.00	0.00	N/C	0.00
DESCRIBE	0.00	0.00	N/C	0.00	SET CURRENT RULES	0.00	0.00	N/C	0.00
DESCRIBE TABLE	0.00	0.00	N/C	0.00					
OPEN	0.00	0.00	N/C	0.00	CONNECT TYPE 1	0.00	0.00	N/C	0.00
CLOSE	1.00	0.03	N/C	0.50	CONNECT TYPE 2	1.00	0.03	N/C	0.50
FETCH	101.00	2.98	N/C	50.50	RELEASE	0.00	0.00	N/C	0.00
					SET CONNECTION	0.00	0.00	N/C	0.00
TOTAL	102.00	3.01	N/C	51.00	ASSOCIATE LOCATORS	0.00	0.00	N/C	0.00
					ALLOCATE CURSOR	0.00	0.00	N/C	0.00
					TOTAL	1.00	0.03	N/C	0.50

1 LOCATION: USIBMT6BOAPLX

DB2 PERFORMANCE MONITOR (V5)

PAGE: 2-2

GROUP: BOAG

STATISTICS REPORT - LONG

REQUESTED FROM: 02/12/99 15:06:03.00

MEMBER: NB22

TO: 02/12/99 15:44:05.00

SUBSYSTEM: NB22

INTERVAL FROM: 02/12/99 15:09:49.64

DB2 VERSION: V5

SCOPE: MEMBER

TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START	: 02/12/99 15:09:49.64	SAMPLING START:	02/12/99 15:09:49.64	TOTAL THREADS	:	0.00
INTERVAL END	: 02/12/99 15:43:45.80	SAMPLING END	: 02/12/99 15:43:45.80	TOTAL COMMITS	:	2.00
INTERVAL ELAPSED:	33:56.157107	OUTAGE ELAPSED:	0.000000	DATA SHARING MEMBER:	:	N/A

SQL DDL	QUANTITY	/MINUTE	/THREAD	/COMMIT	STORED PROCEDURES	QUANTITY	/MINUTE	/THREAD	/COMMIT
CREATE TABLE	0.00	0.00	N/C	0.00	CALL STATEMENTS EXECUTED	0.00	0.00	N/C	0.00
CREATE TEMP TABLE	0.00	0.00	N/C	0.00	PROCEDURE ABENDS	0.00	0.00	N/C	0.00
CREATE INDEX	0.00	0.00	N/C	0.00	CALL STATEMENT TIMEOUTS	0.00	0.00	N/C	0.00
CREATE VIEW	0.00	0.00	N/C	0.00	CALL STATEMENT REJECTED	0.00	0.00	N/C	0.00
CREATE SYNONYM	0.00	0.00	N/C	0.00					
CREATE TABLESPACE	0.00	0.00	N/C	0.00					
CREATE DATABASE	0.00	0.00	N/C	0.00					
CREATE STOGROUP	0.00	0.00	N/C	0.00					
CREATE ALIAS	0.00	0.00	N/C	0.00					
ALTER TABLE	0.00	0.00	N/C	0.00					
ALTER INDEX	0.00	0.00	N/C	0.00					
ALTER TABLESPACE	0.00	0.00	N/C	0.00					
ALTER DATABASE	0.00	0.00	N/C	0.00					
ALTER STOGROUP	0.00	0.00	N/C	0.00					
DROP TABLE	0.00	0.00	N/C	0.00					
DROP INDEX	0.00	0.00	N/C	0.00					
DROP VIEW	0.00	0.00	N/C	0.00					
DROP SYNONYM	0.00	0.00	N/C	0.00					
DROP TABLESPACE	0.00	0.00	N/C	0.00					
DROP DATABASE	0.00	0.00	N/C	0.00					
DROP STOGROUP	0.00	0.00	N/C	0.00					
DROP ALIAS	0.00	0.00	N/C	0.00					
DROP PACKAGE	0.00	0.00	N/C	0.00					
RENAME TABLE	0.00	0.00	N/C	0.00					
COMMENT ON	0.00	0.00	N/C	0.00					
LABEL ON	0.00	0.00	N/C	0.00					
TOTAL	0.00	0.00	N/C	0.00					

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-3
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

EDM POOL	QUANTITY	/MINUTE	/THREAD	/COMMIT	SUBSYSTEM SERVICES	QUANTITY	/MINUTE	/THREAD	/COMMIT
PAGES IN EDM POOL	12500.00	N/A	N/A	N/A	IDENTIFY	0.00	0.00	N/C	0.00
% PAGES IN USE	0.22				CREATE THREAD	0.00	0.00	N/C	0.00
FREE PAGES IN FREE CHAIN	12473.12	N/A	N/A	N/A	SIGNON	0.00	0.00	N/C	0.00
PAGES USED FOR CT	0.88	N/A	N/A	N/A	TERMINATE	2.00	0.06	N/C	1.00
PAGES USED FOR DBD	23.00	N/A	N/A	N/A	ROLLBACK	0.00	0.00	N/C	0.00
PAGES USED FOR SKCT	3.00	N/A	N/A	N/A					
PAGES USED FOR PT	0.00	N/A	N/A	N/A	COMMIT PHASE 1	0.00	0.00	N/C	0.00
PAGES USED FOR SKPT	0.00	N/A	N/A	N/A	COMMIT PHASE 2	0.00	0.00	N/C	0.00
					READ ONLY COMMIT	0.00	0.00	N/C	0.00
FAILS DUE TO POOL FULL	0.00	0.00	N/C	0.00					
					UNITS OF RECOVERY INDOUBT	0.00	0.00	N/C	0.00
REQUESTS FOR CT SECTIONS	0.00	0.00	N/C	0.00	UNITS OF REC.INDBT RESOLVED	0.00	0.00	N/C	0.00
CT NOT IN EDM POOL	0.00	0.00	N/C	0.00					
CT REQUESTS/CT NOT IN EDM	N/C				SYNCHS(SINGLE PHASE COMMIT)	2.00	0.06	N/C	1.00
REQUESTS FOR PT SECTIONS	0.00	0.00	N/C	0.00	QUEUED AT CREATE THREAD	0.00	0.00	N/C	0.00
PT NOT IN EDM POOL	0.00	0.00	N/C	0.00	SUBSYSTEM ALLIED MEMORY EOT	0.00	0.00	N/C	0.00
PT REQUESTS/PT NOT IN EDM	N/C				SUBSYSTEM ALLIED MEMORY EOM	0.00	0.00	N/C	0.00
REQUESTS FOR DBD SECTIONS	6.00	0.18	N/C	3.00	SYSTEM EVENT CHECKPOINT	0.00	0.00	N/C	0.00
DBD NOT IN EDM POOL	0.00	0.00	N/C	0.00					
DBD REQUESTS/DBD NOT IN EDM	N/C								
PREP_STMT_CACHE_INSERTS	0.00	0.00	N/C	0.00					
PREP_STMT_CACHE_REQUESTS	0.00	0.00	N/C	0.00					
PREP_STMT_CACHE_PAGES_USED	0.00	N/A	N/A	N/A					
PREP_STMT_HIT_RATIO	N/C								

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-4
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

OPEN/CLOSE ACTIVITY	QUANTITY	/MINUTE	/THREAD	/COMMIT	LOG ACTIVITY	QUANTITY	/MINUTE	/THREAD	/COMMIT
OPEN DATASETS - HWM	102.00	N/A	N/A	N/A	READS SATISFIED-OUTPUT BUF	0.00	0.00	N/C	0.00
OPEN DATASETS	102.00	N/A	N/A	N/A	READS SATISFIED-OUTP.BUF(%)	N/C			
DS NOT IN USE,NOT CLOSE-HWM	80.00	N/A	N/A	N/A	READS SATISFIED-ACTIVE LOG	0.00	0.00	N/C	0.00
DS NOT IN USE,NOT CLOSED	38.47	N/A	N/A	N/A	READS SATISFIED-ACTV.LOG(%)	N/C			
IN USE DATA SETS	63.53	N/A	N/A	N/A	READS SATISFIED-ARCHIVE LOG	0.00	0.00	N/C	0.00
					READS SATISFIED-ARCH.LOG(%)	N/C			
DSETS CLOSED-THRESH.REACHED	0.00	0.00	N/C	0.00	TAPE VOLUME CONTENTION WAIT	0.00	0.00	N/C	0.00
DSETS CONVERTED R/W -> R/O	0.00	0.00	N/C	0.00	WRITE-NOWAIT	0.00	0.00	N/C	0.00
					WRITE OUTPUT LOG BUFFERS	509.00	15.00	N/C	254.50
					BSDS ACCESS REQUESTS	0.00	0.00	N/C	0.00
					UNAVAILABLE OUTPUT LOG BUFF	0.00	0.00	N/C	0.00
					CONTR.INTERV.CREATED-ACTIVE	5.00	0.15	N/C	2.50
					ARCHIVE LOG READ ALLOCATION	0.00	0.00	N/C	0.00
					ARCHIVE LOG WRITE ALLOCAT.	0.00	0.00	N/C	0.00
					CONTR.INTERV.OFFLOADED-ARCH	0.00	0.00	N/C	0.00
					READ DELAYED-UNAVAIL.RESOUR	0.00	0.00	N/C	0.00
					LOOK-AHEAD MOUNT ATTEMPTED	0.00	0.00	N/C	0.00
					LOOK-AHEAD MOUNT SUCCESSFUL	0.00	0.00	N/C	0.00

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-5
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

PLAN/PACKAGE PROCESSING	QUANTITY	/MINUTE	/THREAD	/COMMIT	DB2 COMMANDS	QUANTITY	/MINUTE
INCREMENTAL BINDS	0.00	0.00	N/C	0.00	DISPLAY DATABASE	0.00	0.00
					DISPLAY THREAD	0.00	0.00
PLAN ALLOCATION ATTEMPTS	0.00	0.00	N/C	0.00	DISPLAY UTILITY	0.00	0.00
PLAN ALLOCATION SUCCESSFUL	0.00	0.00	N/C	0.00	DISPLAY TRACE	0.00	0.00
PACKAGE ALLOCATION ATTEMPT	0.00	0.00	N/C	0.00	DISPLAY RLIMIT	0.00	0.00
PACKAGE ALLOCATION SUCCESS	0.00	0.00	N/C	0.00	DISPLAY LOCATION	0.00	0.00
					DISPLAY ARCHIVE	0.00	0.00
PLANS BOUND	0.00	0.00	N/C	0.00	DISPLAY BUFFERPOOL	0.00	0.00
BIND ADD SUBCOMMANDS	0.00	0.00	N/C	0.00	DISPLAY GROUDBUFFERPOOL	0.00	0.00
BIND REPLACE SUBCOMMANDS	0.00	0.00	N/C	0.00	DISPLAY GROUP	0.00	0.00
TEST BINDS NO PLAN-ID	0.00	0.00	N/C	0.00	DISPLAY PROCEDURE	0.00	0.00
PACKAGES BOUND	0.00	0.00	N/C	0.00	ALTER BUFFERPOOL	0.00	0.00
BIND ADD PACKAGE SUBCOMMAND	0.00	0.00	N/C	0.00	ALTER GROUDBUFFERPOOL	0.00	0.00
BIND REPLACE PACKAGE SUBCOM	0.00	0.00	N/C	0.00	START DATABASE	0.00	0.00
					START TRACE	0.00	0.00
AUTOMATIC BIND ATTEMPTS	0.00	0.00	N/C	0.00	START DB2	0.00	0.00
AUTOMATIC BINDS SUCCESSFUL	0.00	0.00	N/C	0.00	START RLIMIT	0.00	0.00
AUTO.BIND INVALID RES. IDS	0.00	0.00	N/C	0.00	START DDF	0.00	0.00
AUTO.BIND PACKAGE ATTEMPTS	0.00	0.00	N/C	0.00	START PROCEDURE	0.00	0.00
AUTO.BIND PACKAGES SUCCESS	0.00	0.00	N/C	0.00	STOP DATABASE	0.00	0.00
					STOP TRACE	0.00	0.00
REBIND SUBCOMMANDS	0.00	0.00	N/C	0.00	STOP DB2	0.00	0.00
ATTEMPTS TO REBIND A PLAN	0.00	0.00	N/C	0.00	STOP RLIMIT	0.00	0.00
PLANS REBOUND	0.00	0.00	N/C	0.00	STOP DDF	0.00	0.00
REBIND PACKAGE SUBCOMMANDS	0.00	0.00	N/C	0.00	STOP PROCEDURE	0.00	0.00
ATTEMPTS TO REBIND PACKAGE	0.00	0.00	N/C	0.00	MODIFY TRACE	1.00	0.03
PACKAGES REBOUND	0.00	0.00	N/C	0.00	CANCEL THREAD	0.00	0.00
					TERM UTILITY	0.00	0.00
FREE PLAN SUBCOMMANDS	0.00	0.00	N/C	0.00	RECOVER BSDS	0.00	0.00
ATTEMPTS TO FREE A PLAN	0.00	0.00	N/C	0.00	RECOVER INDOUBT	0.00	0.00
PLANS FREED	0.00	0.00	N/C	0.00	RESET INDOUBT	0.00	0.00
FREE PACKAGE SUBCOMMANDS	0.00	0.00	N/C	0.00	RESET GENERICCLU	0.00	0.00
ATTEMPTS TO FREE A PACKAGE	0.00	0.00	N/C	0.00	ARCHIVE LOG	0.00	0.00
PACKAGES FREED	0.00	0.00	N/C	0.00	SET ARCHIVE	0.00	0.00
					UNRECOGNIZED COMMANDS	0.00	0.00
					TOTAL	1.00	0.03

1 LOCATION: USIBMT6BOAPLX DB2 PERFORMANCE MONITOR (V5) PAGE: 2-6
 GROUP: BOAG STATISTICS REPORT - LONG REQUESTED FROM: 02/12/99 15:06:03.00
 MEMBER: NB22 TO: 02/12/99 15:44:05.00
 SUBSYSTEM: NB22 INTERVAL FROM: 02/12/99 15:09:49.64
 DB2 VERSION: V5 SCOPE: MEMBER TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

RID LIST PROCESSING	QUANTITY	/MINUTE	/THREAD	/COMMIT	AUTHORIZATION MANAGEMENT	QUANTITY	/MINUTE	/THREAD	/COMMIT
MAX RID BLOCKS ALLOCATED	0.00	N/A	N/A	N/A	AUTH ATTEMPTS-PLAN	1.00	0.03	N/C	0.50
CURRENT RID BLOCKS ALLOCAT.	0.00	N/A	N/A	N/A	AUTH SUCC-PLAN	1.00	0.03	N/C	0.50
TERMINATED-NO STORAGE	0.00	0.00	N/C	0.00	AUTH SUCC-PLAN-W/O CATALOG	0.00	0.00	N/C	0.00
TERMINATED-EXCEED RDS LIMIT	0.00	0.00	N/C	0.00	AUTH SUCC-PLAN-PUB-W/O CAT	0.00	0.00	N/C	0.00
TERMINATED-EXCEED DM LIMIT	0.00	0.00	N/C	0.00					
TERMINATED-EXCEED PROC.LIM.	0.00	0.00	N/C	0.00	AUTH SUCC-PKG-W/O CATALOG	0.00	0.00	N/C	0.00
					AUTH SUCC-PKG-PUB-W/O CAT	0.00	0.00	N/C	0.00
					AUTH UNSUCC-PKG-CACHE	0.00	0.00	N/C	0.00
					PKG CACHE OVERFLW - AUTH ID	0.00	0.00	N/C	0.00
					PKG CACHE OVERFLW - COLL ID	0.00	0.00	N/C	0.00

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-7
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

LOCKING ACTIVITY	QUANTITY	/MINUTE	/THREAD	/COMMIT	DATA SHARING LOCKING	QUANTITY	/MINUTE	/THREAD	/COMMIT
SUSPENSIONS (ALL)	1.00	0.03	N/C	0.50	GLOBAL CONTENTION RATE (%)	0.00			
SUSPENSIONS (LOCK ONLY)	0.00	0.00	N/C	0.00	FALSE CONTENTION RATE (%)	0.00			
SUSPENSIONS (LATCH ONLY)	0.00	0.00	N/C	0.00					
SUSPENSIONS (OTHER)	1.00	0.03	N/C	0.50	LOCK REQUESTS (P-LOCKS)	0.00	0.00	N/C	0.00
					UNLOCK REQUESTS (P-LOCKS)	0.00	0.00	N/C	0.00
TIMEOUTS	0.00	0.00	N/C	0.00	CHANGE REQUESTS (P-LOCKS)	0.00	0.00	N/C	0.00
DEADLOCKS	0.00	0.00	N/C	0.00					
					SYNCH.XES - LOCK REQUESTS	2044.00	60.23	N/C	1022.00
LOCK REQUESTS	2062.00	60.76	N/C	1031.00	SYNCH.XES - CHANGE REQUESTS	0.00	0.00	N/C	0.00
UNLOCK REQUESTS	2073.00	61.09	N/C	1036.50	SYNCH.XES - UNLOCK REQUESTS	2055.00	60.56	N/C	1027.50
QUERY REQUESTS	0.00	0.00	N/C	0.00	ASYNCH.XES - RESOURCES	0.00	0.00	N/C	0.00
CHANGE REQUESTS	5.00	0.15	N/C	2.50					
OTHER REQUESTS	0.00	0.00	N/C	0.00	SUSPENDS - IRLM GLOBAL CONT	0.00	0.00	N/C	0.00
					SUSPENDS - XES GLOBAL CONT.	0.00	0.00	N/C	0.00
LOCK ESCALATION (SHARED)	0.00	0.00	N/C	0.00	SUSPENDS - FALSE CONTENTION	0.00	0.00	N/C	0.00
LOCK ESCALATION (EXCLUSIVE)	0.00	0.00	N/C	0.00	INCOMPATIBLE RETAINED LOCK	0.00	0.00	N/C	0.00
DRAIN REQUESTS	0.00	0.00	N/C	0.00	NOTIFY MESSAGES SENT	1.00	0.03	N/C	0.50
DRAIN REQUESTS FAILED	0.00	0.00	N/C	0.00	NOTIFY MESSAGES RECEIVED	0.00	0.00	N/C	0.00
CLAIM REQUESTS	56.00	1.65	N/C	28.00	P-LOCK/NOTIFY EXITS ENGINES	500.00	N/A	N/A	N/A
CLAIM REQUESTS FAILED	0.00	0.00	N/C	0.00	P-LCK/NFY EX.ENGINE UNAVAIL	0.00	0.00	N/C	0.00
					PSET/PART P-LCK NEGOTIATION	0.00	0.00	N/C	0.00
					PAGE P-LOCK NEGOTIATION	0.00	0.00	N/C	0.00
					OTHER P-LOCK NEGOTIATION	0.00	0.00	N/C	0.00
					P-LOCK CHANGE DURING NEG.	0.00	0.00	N/C	0.00

GLOBAL DDF ACTIVITY	QUANTITY	/MINUTE	/THREAD	/COMMIT	QUERY PARALLELISM	QUANTITY	/MINUTE	/THREAD	/COMMIT
DBAT QUEUED-MAXIMUM ACTIVE	N/P	N/P	N/P	N/A	MAX.DEGREE OF PARALLELISM	5.00	N/A	N/A	N/A
CONV.DEALLOC-MAX.CONNECTED	N/P	N/P	N/P	N/A	PARALLEL GROUPS EXECUTED	1.00	0.03	N/C	0.50
INACTIVE DBATS - CURRENTLY	N/P	N/A	N/A	N/A	RAN AS PLANNED	1.00	0.03	N/C	0.50
INACTIVE DBATS - HWM	N/P	N/A	N/A	N/A	RAN REDUCED	0.00	0.00	N/C	0.00
ACTIVE DBATS - CURRENTLY	N/P	N/A	N/A	N/A	SEQUENTIAL-CURSOR	0.00	0.00	N/C	0.00
ACTIVE DBATS - HWM	N/P	N/A	N/A	N/A	SEQUENTIAL-NO ESA	0.00	0.00	N/C	0.00
TOTAL DBATS - HWM	N/P	N/A	N/A	N/A	SEQUENTIAL-NO BUFFER	0.00	0.00	N/C	0.00
COLD START CONNECTIONS	N/P	N/P	N/P	N/P	SEQUENTIAL-ENCLAVE SER.	0.00	0.00	N/C	0.00
WARM START CONNECTIONS	N/P	N/P	N/P	N/P	ONE DB2 - COORDINATOR = NO	0.00	0.00	N/C	0.00
RESYNCHRONIZATION ATTEMPTED	N/P	N/P	N/P	N/P	ONE DB2 - ISOLATION LEVEL	0.00	0.00	N/C	0.00
RESYNCHRONIZATION SUCCEEDED	N/P	N/P	N/P	N/P	MEMBER SKIPPED (%)	0.00			

GROUP: BOAG

STATISTICS REPORT - LONG

REQUESTED FROM: 02/12/99 15:06:03.00

MEMBER: NB22

TO: 02/12/99 15:44:05.00

SUBSYSTEM: NB22

INTERVAL FROM: 02/12/99 15:09:49.64

DB2 VERSION: V5

SCOPE: MEMBER

TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START	: 02/12/99 15:09:49.64	SAMPLING START	: 02/12/99 15:09:49.64	TOTAL THREADS	:	0.00
INTERVAL END	: 02/12/99 15:43:45.80	SAMPLING END	: 02/12/99 15:43:45.80	TOTAL COMMITS	:	2.00
INTERVAL ELAPSED:	33:56.157107	OUTAGE ELAPSED:	0.000000	DATA SHARING MEMBER:		N/A
CPU TIMES	TCB TIME	SRB TIME	TOTAL TIME	/THREAD		/COMMIT
SYSTEM SERVICES ADDRESS SPACE	2.416151	0.344193	2.760344	N/C		1.380172
DATABASE SERVICES ADDRESS SPACE	0.018771	1:10:30.043799	1:10:30.062569	N/C		35:15.031285
IRLM	0.000490	0.958605	0.959095	N/C		0.479547
DDF ADDRESS SPACE	0.000000	0.000117	0.000117	N/C		0.000059
TOTAL	2.435412	1:10:31.346714	1:10:33.782125	N/C		35:16.891063
NON-CPU TIME	N/A	N/A	0.000000	N/C		0.000000

DB2 APPL.PROGR.INTERFACE	QUANTITY	/MINUTE	/THREAD	/COMMIT	DATA CAPTURE	QUANTITY	/MINUTE	/THREAD	/COMMIT
ABENDS	0.00	0.00	N/C	0.00	LOG RECORDS CAPTURED	0.00	0.00	N/C	0.00
UNRECOGNIZED	0.00	0.00	N/C	0.00	LOG READS PERFORMED	0.00	0.00	N/C	0.00
COMMAND REQUESTS	0.00	0.00	N/C	0.00	LOG RECORDS RETURNED	0.00	0.00	N/C	0.00
READA REQUESTS	0.00	0.00	N/C	0.00	DATA ROWS RETURNED	0.00	0.00	N/C	0.00
READS REQUESTS	0.00	0.00	N/C	0.00	DESCRIBES PERFORMED	0.00	0.00	N/C	0.00
WRITE REQUESTS	0.00	0.00	N/C	0.00	DATA DESCRIPTIONS RETURNED	0.00	0.00	N/C	0.00
					TABLES RETURNED	0.00	0.00	N/C	0.00
TOTAL	0.00	0.00	N/C	0.00					

OPTIMIZATION	QUANTITY	/MINUTE	/THREAD	/COMMIT	IFC DEST.	WRITTEN	NOT WRTN	BUF.OVER	NOT ACCP	WRT.FAIL
PREP_STMT_MATCH	0.00	0.00	N/C	0.00	SMF	36.00	0.00	0.00	0.00	0.00
PREP_STMT_NO_MATCH	0.00	0.00	N/C	0.00	GTF	0.00	0.00	N/A	0.00	0.00
IMPLICIT_PREPARES	0.00	0.00	N/C	0.00	OP1	0.00	0.00	N/A	0.00	N/A
PREP_FROM_CACHE	0.00	0.00	N/C	0.00	OP2	0.00	0.00	N/A	0.00	N/A
CACHE_LIMIT_EXCEED	0.00	0.00	N/C	0.00	OP3	0.00	0.00	N/A	0.00	N/A
PREP_STMT_PURGED	0.00	0.00	N/C	0.00	OP4	0.00	0.00	N/A	0.00	N/A
					OP5	0.00	0.00	N/A	0.00	N/A
					OP6	0.00	0.00	N/A	0.00	N/A
					OP7	0.00	0.00	N/A	0.00	N/A
					OP8	0.00	0.00	N/A	0.00	N/A
					RES	0.00	N/A	N/A	N/A	N/A
					TOTAL	36.00	0.00		0.00	0.00

IFC RECORD COUNTS	WRITTEN	NOT WRTN	LATCH CNT	QUANTITY	QUANTITY	QUANTITY	QUANTITY	MISCELLANEOUS
SYSTEM RELATED	7.00	7.00	LC01-LC04	0.00	0.00	0.00	0.00	BYPASS COL: 0.00
DATABASE RELATED	7.00	7.00	LC05-LC08	0.00	0.00	0.00	0.00	
ACCOUNTING	6.00	0.00	LC09-LC12	0.00	0.00	0.00	0.00	
START TRACE	1.00	0.00	LC13-LC16	0.00	5.00	0.00	0.00	
STOP TRACE	1.00	0.00	LC17-LC20	0.00	0.00	0.00	0.00	
SYSTEM PARAMETERS	0.00	0.00	LC21-LC24	0.00	0.00	1.00	3110.00	
SYS.PARMS-BPOOLS	7.00	7.00	LC25-LC28	0.00	0.00	0.00	0.00	
AUDIT	0.00	0.00	LC29-LC32	0.00	0.00	0.00	0.00	
TOTAL	29.00	21.00						

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-9
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BPO	GENERAL	QUANTITY	/MINUTE	/THREAD	/COMMIT	BPO	READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
	CURRENT ACTIVE BUFFERS	0.00	N/A	N/A	N/A		BPOOL HIT RATIO (%)	100.00			
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	N/C	0.00		GETPAGE REQUEST	3.00	0.09	N/C	1.50
	NUMBER OF DATASET OPENS	0.00	0.00	N/C	0.00		GETPAGE REQUEST-SEQUENTIAL	0.00	0.00	N/C	0.00
	BUFFERS ALLOCATED - VPOOL	1000.00	29.47	N/C	500.00		GETPAGE REQUEST-RANDOM	3.00	0.09	N/C	1.50
	BUFFERS ALLOCATED - HPOOL	0.00	0.00	N/C	0.00		SYNCHRONOUS READS	0.00	0.00	N/C	0.00
	HPOOL BUFFERS BACKED	0.00	0.00	N/C	0.00		SYNCHRON. READS-SEQUENTIAL	0.00	0.00	N/C	0.00
	DFHSM MIGRATED DATASET	0.00	0.00	N/C	0.00		SYNCHRON. READS-RANDOM	0.00	0.00	N/C	0.00
	DFHSM RECALL TIMEOUTS	0.00	0.00	N/C	0.00		GETPAGE PER SYN.READ-RANDOM	N/C			
	HPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH REQUEST	0.00	0.00	N/C	0.00
	VPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH READS	0.00	0.00	N/C	0.00
	VPOOL OR HPOOL EXP.FAILURE	0.00	0.00	N/C	0.00		PAGES READ VIA SEQ.PREFETCH	0.00	0.00	N/C	0.00
	CONCUR.PREF.I/O STREAMS-HWM	0.00	N/A	N/A	N/A		S.PRF.PAGES READ/S.PRF.READ	N/C			
	PREF.I/O STREAMS REDUCTION	0.00	0.00	N/C	0.00		LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
	PARALLEL QUERY REQUESTS	0.00	0.00	N/C	0.00		LIST PREFETCH READS	0.00	0.00	N/C	0.00
	PARALL.QUERY REQ.REDUCTION	0.00	0.00	N/C	0.00		PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
	PREF.QUANT.REDUCED TO 1/2	0.00	0.00	N/C	0.00		L.PRF.PAGES READ/L.PRF.READ	N/C			
	PREF.QUANT.REDUCED TO 1/4	0.00	0.00	N/C	0.00		DYNAMIC PREFETCH REQUESTED	0.00	0.00	N/C	0.00
							DYNAMIC PREFETCH READS	0.00	0.00	N/C	0.00
							PAGES READ VIA DYN.PREFETCH	0.00	0.00	N/C	0.00
							D.PRF.PAGES READ/D.PRF.READ	N/C			
							PREF.DISABLED-NO BUFFER	0.00	0.00	N/C	0.00
							PREF.DISABLED-NO READ ENG	0.00	0.00	N/C	0.00
							SYNC.HPOOL READ	0.00	0.00	N/C	0.00
							ASYN.HPOOL READ	0.00	0.00	N/C	0.00
							HPOOL READ FAILED	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-S	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-F	0.00	0.00	N/C	0.00
							PAGE-INS REQUIRED FOR READ	0.00	0.00	N/C	0.00

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-10
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BPO	WRITE OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT	BPO	SORT/MERGE	QUANTITY	/MINUTE	/THREAD	/COMMIT
	BUFFER UPDATES	0.00	0.00	N/C	0.00		MAX WORKFILES CONCURR. USED	0.00	N/A	N/A	N/A
	PAGES WRITTEN	0.00	0.00	N/C	0.00		MERGE PASSES REQUESTED	0.00	0.00	N/C	0.00
	BUFF.UPDATES/PAGES WRITTEN	N/C					MERGE PASS DEGRADED-LOW BUF	0.00	0.00	N/C	0.00
							WORKFILE REQ.REJCTD-LOW BUF	0.00	0.00	N/C	0.00
	SYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE REQ-ALL MERGE PASS	0.00	0.00	N/C	0.00
	ASYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE NOT CREATED-NO BUF	0.00	0.00	N/C	0.00
							WORKFILE PRF NOT SCHEDULED	0.00	0.00	N/C	0.00
	PAGES WRITTEN PER WRITE I/O	N/C					WORKFILE PAGES TO DESTRUCT	0.00	0.00	N/C	0.00
							WORKFILE PAGES NOT WRITTEN	0.00	0.00	N/C	0.00
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	VERTI.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	DM CRITICAL THRESHOLD	0.00	0.00	N/C	0.00						
	WRITE ENGINE NOT AVAILABLE	0.00	0.00	N/C	0.00						
	SYNC.HPOOL WRITE	0.00	0.00	N/C	0.00						
	ASYN.HPOOL WRITE	0.00	0.00	N/C	0.00						
	HPOOL WRITE FAILED	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-S	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-F	0.00	0.00	N/C	0.00						
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00	N/C	0.00						

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-11
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BP2	GENERAL	QUANTITY	/MINUTE	/THREAD	/COMMIT	BP2	READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
	CURRENT ACTIVE BUFFERS	97.50	N/A	N/A	N/A		BPOOL HIT RATIO (%)	0.03			
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	N/C	0.00		GETPAGE REQUEST	5674.7K	167.2K	N/C	2837.3K
	NUMBER OF DATASET OPENS	0.00	0.00	N/C	0.00		GETPAGE REQUEST-SEQUENTIAL	5233.2K	154.2K	N/C	2616.6K
	BUFFERS ALLOCATED - VPOOL	50000.00	1473.36	N/C	25.0K		GETPAGE REQUEST-RANDOM	441.4K	13.0K	N/C	220.7K
	BUFFERS ALLOCATED - HPOOL	0.00	0.00	N/C	0.00		SYNCHRONOUS READS	9154.00	269.74	N/C	4577.00
	HPOOL BUFFERS BACKED	0.00	0.00	N/C	0.00		SYNCHRON. READS-SEQUENTIAL	1435.00	42.29	N/C	717.50
	DFHSM MIGRATED DATASET	0.00	0.00	N/C	0.00		SYNCHRON. READS-RANDOM	7719.00	227.46	N/C	3859.50
	DFHSM RECALL TIMEOUTS	0.00	0.00	N/C	0.00		GETPAGE PER SYN.READ-RANDOM	57.19			
	HPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH REQUEST	163.1K	4804.82	N/C	81.5K
	VPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH READS	161.2K	4749.12	N/C	80.6K
	VPOOL OR HPOOL EXP.FAILURE	0.00	0.00	N/C	0.00		PAGES READ VIA SEQ.PREFETCH	5155.8K	151.9K	N/C	2577.9K
	CNCR.PREF.I/O STREAMS-HWM	5.00	N/A	N/A	N/A		S.PRF.PAGES READ/S.PRF.READ	31.99			
	PREF.I/O STREAMS REDUCTION	0.00	0.00	N/C	0.00		LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
	PARALLEL QUERY REQUESTS	1.00	0.03	N/C	0.50		LIST PREFETCH READS	0.00	0.00	N/C	0.00
	PARALL.QUERY REQ.REDUCTION	0.00	0.00	N/C	0.00		PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
	PREF.QUANT.REDUCED TO 1/2	0.00	0.00	N/C	0.00		L.PRF.PAGES READ/L.PRF.READ	N/C			
	PREF.QUANT.REDUCED TO 1/4	0.00	0.00	N/C	0.00		DYNAMIC PREFETCH REQUESTED	16958.00	499.71	N/C	8479.00
							DYNAMIC PREFETCH READS	16389.00	482.94	N/C	8194.50
							PAGES READ VIA DYN.PREFETCH	507.8K	15.0K	N/C	253.9K
							D.PRF.PAGES READ/D.PRF.READ	30.99			
							PREF.DISABLED-NO BUFFER	0.00	0.00	N/C	0.00
							PREF.DISABLED-NO READ ENG	0.00	0.00	N/C	0.00
							SYNC.HPOOL READ	0.00	0.00	N/C	0.00
							ASYN.HPOOL READ	0.00	0.00	N/C	0.00
							HPOOL READ FAILED	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-S	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-F	0.00	0.00	N/C	0.00
							PAGE-INS REQUIRED FOR READ	0.00	0.00	N/C	0.00

1 LOCATION: USIBMT6BOAPLX DB2 PERFORMANCE MONITOR (V5) PAGE: 2-12
 GROUP: BOAG STATISTICS REPORT - LONG REQUESTED FROM: 02/12/99 15:06:03.00
 MEMBER: NB22 TO: 02/12/99 15:44:05.00
 SUBSYSTEM: NB22 INTERVAL FROM: 02/12/99 15:09:49.64
 DB2 VERSION: V5 SCOPE: MEMBER TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BP2	WRITE OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT	BP2	SORT/MERGE	QUANTITY	/MINUTE	/THREAD	/COMMIT
	BUFFER UPDATES	0.00	0.00	N/C	0.00		MAX WORKFILES CONCURR. USED	0.00	N/A	N/A	N/A
	PAGES WRITTEN	0.00	0.00	N/C	0.00		MERGE PASSES REQUESTED	0.00	0.00	N/C	0.00
	BUFF.UPDATES/PAGES WRITTEN	N/C					MERGE PASS DEGRADED-LOW BUF	0.00	0.00	N/C	0.00
							WORKFILE REQ.REJCTD-LOW BUF	0.00	0.00	N/C	0.00
	SYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE REQ-ALL MERGE PASS	0.00	0.00	N/C	0.00
	ASYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE NOT CREATED-NO BUF	0.00	0.00	N/C	0.00
							WORKFILE PRF NOT SCHEDULED	0.00	0.00	N/C	0.00
	PAGES WRITTEN PER WRITE I/O	N/C					WORKFILE PAGES TO DESTRUCT	0.00	0.00	N/C	0.00
							WORKFILE PAGES NOT WRITTEN	0.00	0.00	N/C	0.00
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	VERTI.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	DM CRITICAL THRESHOLD	0.00	0.00	N/C	0.00						
	WRITE ENGINE NOT AVAILABLE	0.00	0.00	N/C	0.00						
	SYNC.HPOOL WRITE	0.00	0.00	N/C	0.00						
	ASYN.HPOOL WRITE	0.00	0.00	N/C	0.00						
	HPOOL WRITE FAILED	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-S	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-F	0.00	0.00	N/C	0.00						
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00	N/C	0.00						

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-13
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BP4	GENERAL	QUANTITY	/MINUTE	/THREAD	/COMMIT	BP4	READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
	CURRENT ACTIVE BUFFERS	1.77	N/A	N/A	N/A		BPOOL HIT RATIO (%)	55.12			
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	N/C	0.00		GETPAGE REQUEST	221.8K	6534.43	N/C	110.9K
	NUMBER OF DATASET OPENS	0.00	0.00	N/C	0.00		GETPAGE REQUEST-SEQUENTIAL	18427.00	542.99	N/C	9213.50
	BUFFERS ALLOCATED - VPOOL	50000.00	1473.36	N/C	25.0K		GETPAGE REQUEST-RANDOM	203.3K	5991.43	N/C	101.7K
	BUFFERS ALLOCATED - HPOOL	100.0K	2946.73	N/C	50.0K		SYNCHRONOUS READS	613.00	18.06	N/C	306.50
	HPOOL BUFFERS BACKED	51417.53	1515.13	N/C	25.7K		SYNCHRON. READS-SEQUENTIAL	64.00	1.89	N/C	32.00
	DFHSM MIGRATED DATASET	0.00	0.00	N/C	0.00		SYNCHRON. READS-RANDOM	549.00	16.18	N/C	274.50
	DFHSM RECALL TIMEOUTS	0.00	0.00	N/C	0.00		GETPAGE PER SYN.READ-RANDOM	370.36			
	HPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH REQUEST	577.00	17.00	N/C	288.50
	VPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH READS	577.00	17.00	N/C	288.50
	VPOOL OR HPOOL EXP.FAILURE	0.00	0.00	N/C	0.00		PAGES READ VIA SEQ.PREFETCH	18440.00	543.38	N/C	9220.00
	CONCUR.PREF.I/O STREAMS-HWM	0.00	N/A	N/A	N/A		S.PRF.PAGES READ/S.PRF.READ	31.96			
	PREF.I/O STREAMS REDUCTION	0.00	0.00	N/C	0.00		LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
	PARALLEL QUERY REQUESTS	0.00	0.00	N/C	0.00		LIST PREFETCH READS	0.00	0.00	N/C	0.00
	PARALL.QUERY REQ.REDUCTION	0.00	0.00	N/C	0.00		PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
	PREF.QUANT.REDUCED TO 1/2	0.00	0.00	N/C	0.00		L.PRF.PAGES READ/L.PRF.READ	N/C			
	PREF.QUANT.REDUCED TO 1/4	0.00	0.00	N/C	0.00		DYNAMIC PREFETCH REQUESTED	2515.00	74.11	N/C	1257.50
							DYNAMIC PREFETCH READS	2515.00	74.11	N/C	1257.50
							PAGES READ VIA DYN.PREFETCH	80470.00	2371.23	N/C	40.2K
							D.PRF.PAGES READ/D.PRF.READ	32.00			
							PREF.DISABLED-NO BUFFER	0.00	0.00	N/C	0.00
							PREF.DISABLED-NO READ ENG	0.00	0.00	N/C	0.00
							SYNC.HPOOL READ	0.00	0.00	N/C	0.00
							ASYN.HPOOL READ	0.00	0.00	N/C	0.00
							HPOOL READ FAILED	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-S	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-F	0.00	0.00	N/C	0.00
							PAGE-INS REQUIRED FOR READ	59.00	1.74	N/C	29.50

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-14
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----
 INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BP4	WRITE OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT	BP4	SORT/MERGE	QUANTITY	/MINUTE	/THREAD	/COMMIT
	BUFFER UPDATES	0.00	0.00	N/C	0.00		MAX WORKFILES CONCURR. USED	0.00	N/A	N/A	N/A
	PAGES WRITTEN	0.00	0.00	N/C	0.00		MERGE PASSES REQUESTED	0.00	0.00	N/C	0.00
	BUFF.UPDATES/PAGES WRITTEN	N/C					MERGE PASS DEGRADED-LOW BUF	0.00	0.00	N/C	0.00
							WORKFILE REQ.REJCTD-LOW BUF	0.00	0.00	N/C	0.00
	SYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE REQ-ALL MERGE PASS	0.00	0.00	N/C	0.00
	ASYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE NOT CREATED-NO BUF	0.00	0.00	N/C	0.00
							WORKFILE PRF NOT SCHEDULED	0.00	0.00	N/C	0.00
	PAGES WRITTEN PER WRITE I/O	N/C					WORKFILE PAGES TO DESTRUCT	0.00	0.00	N/C	0.00
							WORKFILE PAGES NOT WRITTEN	0.00	0.00	N/C	0.00
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	VERTI.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	DM CRITICAL THRESHOLD	0.00	0.00	N/C	0.00						
	WRITE ENGINE NOT AVAILABLE	0.00	0.00	N/C	0.00						
	SYNC.HPOOL WRITE	0.00	0.00	N/C	0.00						
	ASYN.HPOOL WRITE	60274.00	1776.11	N/C	30.1K						
	HPOOL WRITE FAILED	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-S	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-F	0.00	0.00	N/C	0.00						
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00	N/C	0.00						

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-15
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BP5	GENERAL	QUANTITY	/MINUTE	/THREAD	/COMMIT	BP5	READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
	CURRENT ACTIVE BUFFERS	7.28	N/A	N/A	N/A		BPOOL HIT RATIO (%)	*****			
	UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	N/C	0.00		GETPAGE REQUEST	68.00	2.00	N/C	34.00
	NUMBER OF DATASET OPENS	0.00	0.00	N/C	0.00		GETPAGE REQUEST-SEQUENTIAL	37.00	1.09	N/C	18.50
	BUFFERS ALLOCATED - VPOOL	50000.00	1473.36	N/C	25.0K		GETPAGE REQUEST-RANDOM	31.00	0.91	N/C	15.50
	BUFFERS ALLOCATED - HPOOL	200.0K	5893.45	N/C	100.0K		SYNCHRONOUS READS	7.00	0.21	N/C	3.50
	HPOOL BUFFERS BACKED	0.00	0.00	N/C	0.00		SYNCHRON. READS-SEQUENTIAL	0.00	0.00	N/C	0.00
	DFHSM MIGRATED DATASET	0.00	0.00	N/C	0.00		SYNCHRON. READS-RANDOM	7.00	0.21	N/C	3.50
	DFHSM RECALL TIMEOUTS	0.00	0.00	N/C	0.00		GETPAGE PER SYN.READ-RANDOM	4.43			
	HPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH REQUEST	8.00	0.24	N/C	4.00
	VPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00		SEQUENTIAL PREFETCH READS	8.00	0.24	N/C	4.00
	VPOOL OR HPOOL EXP.FAILURE	0.00	0.00	N/C	0.00		PAGES READ VIA SEQ.PREFETCH	64.00	1.89	N/C	32.00
	CONCUR.PREF.I/O STREAMS-HWM	0.00	N/A	N/A	N/A		S.PRF.PAGES READ/S.PRF.READ	8.00			
	PREF.I/O STREAMS REDUCTION	0.00	0.00	N/C	0.00		LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
	PARALLEL QUERY REQUESTS	0.00	0.00	N/C	0.00		LIST PREFETCH READS	0.00	0.00	N/C	0.00
	PARALL.QUERY REQ.REDUCTION	0.00	0.00	N/C	0.00		PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
	PREF.QUANT.REDUCED TO 1/2	0.00	0.00	N/C	0.00		L.PRF.PAGES READ/L.PRF.READ	N/C			
	PREF.QUANT.REDUCED TO 1/4	0.00	0.00	N/C	0.00		DYNAMIC PREFETCH REQUESTED	0.00	0.00	N/C	0.00
							DYNAMIC PREFETCH READS	0.00	0.00	N/C	0.00
							PAGES READ VIA DYN.PREFETCH	0.00	0.00	N/C	0.00
							D.PRF.PAGES READ/D.PRF.READ	N/C			
							PREF.DISABLED-NO BUFFER	0.00	0.00	N/C	0.00
							PREF.DISABLED-NO READ ENG	0.00	0.00	N/C	0.00
							SYNC.HPOOL READ	0.00	0.00	N/C	0.00
							ASYN.HPOOL READ	0.00	0.00	N/C	0.00
							HPOOL READ FAILED	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-S	0.00	0.00	N/C	0.00
							ASYN.DA.MOVER HPOOL READ-F	0.00	0.00	N/C	0.00
							PAGE-INS REQUIRED FOR READ	69.00	2.03	N/C	34.50

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-16
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

BP5	WRITE OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT	BP5	SORT/MERGE	QUANTITY	/MINUTE	/THREAD	/COMMIT
	BUFFER UPDATES	46.00	1.36	N/C	23.00		MAX WORKFILES CONCURR. USED	0.00	N/A	N/A	N/A
	PAGES WRITTEN	0.00	0.00	N/C	0.00		MERGE PASSES REQUESTED	0.00	0.00	N/C	0.00
	BUFF.UPDATES/PAGES WRITTEN	N/C					MERGE PASS DEGRADED-LOW BUF	0.00	0.00	N/C	0.00
							WORKFILE REQ.REJCTD-LOW BUF	0.00	0.00	N/C	0.00
	SYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE REQ-ALL MERGE PASS	0.00	0.00	N/C	0.00
	ASYNCHRONOUS WRITES	0.00	0.00	N/C	0.00		WORKFILE NOT CREATED-NO BUF	0.00	0.00	N/C	0.00
							WORKFILE PRF NOT SCHEDULED	0.00	0.00	N/C	0.00
	PAGES WRITTEN PER WRITE I/O	N/C					WORKFILE PAGES TO DESTRUCT	4.00	0.12	N/C	2.00
							WORKFILE PAGES NOT WRITTEN	4.00	0.12	N/C	2.00
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	VERTI.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	DM CRITICAL THRESHOLD	0.00	0.00	N/C	0.00						
	WRITE ENGINE NOT AVAILABLE	0.00	0.00	N/C	0.00						
	SYNC.HPOOL WRITE	0.00	0.00	N/C	0.00						
	ASYN.HPOOL WRITE	0.00	0.00	N/C	0.00						
	HPOOL WRITE FAILED	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-S	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-F	0.00	0.00	N/C	0.00						
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00	N/C	0.00						

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-17
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

TOT4K GENERAL	QUANTITY	/MINUTE	/THREAD	/COMMIT	TOT4K READ OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT
CURRENT ACTIVE BUFFERS	106.55	N/A	N/A	N/A	BPOOL HIT RATIO (%)	2.10			
UNAVAIL.BUFFER-VPOOL FULL	0.00	0.00	N/C	0.00	GETPAGE REQUEST	5896.5K	173.8K	N/C	2948.2K
NUMBER OF DATASET OPENS	0.00	0.00	N/C	0.00	GETPAGE REQUEST-SEQUENTIAL	5251.7K	154.8K	N/C	2625.9K
BUFFERS ALLOCATED - VPOOL	151.0K	4449.56	N/C	75.5K	GETPAGE REQUEST-RANDOM	644.8K	19.0K	N/C	322.4K
BUFFERS ALLOCATED - HPOOL	300.0K	8840.18	N/C	150.0K	SYNCHRONOUS READS	9774.00	288.01	N/C	4887.00
HPOOL BUFFERS BACKED	51417.53	1515.13	N/C	25.7K	SYNCHRON. READS-SEQUENTIAL	1499.00	44.17	N/C	749.50
DFHSM MIGRATED DATASET	0.00	0.00	N/C	0.00	SYNCHRON. READS-RANDOM	8275.00	243.84	N/C	4137.50
DFHSM RECALL TIMEOUTS	0.00	0.00	N/C	0.00	GETPAGE PER SYN.READ-RANDOM	77.92			
HPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00	SEQUENTIAL PREFETCH REQUEST	163.6K	4822.05	N/C	81.8K
VPOOL EXPANS. OR CONTRACT.	0.00	0.00	N/C	0.00	SEQUENTIAL PREFETCH READS	161.8K	4766.36	N/C	80.9K
VPOOL OR HPOOL EXP.FAILURE	0.00	0.00	N/C	0.00	PAGES READ VIA SEQ.PREFETCH	5174.3K	152.5K	N/C	2587.2K
CONCUR.PREF.I/O STREAMS-HWM	5.00	N/A	N/A	N/A	S.PRF.PAGES READ/S.PRF.READ	31.99			
PREF.I/O STREAMS REDUCTION	0.00	0.00	N/C	0.00	LIST PREFETCH REQUESTS	0.00	0.00	N/C	0.00
PARALLEL QUERY REQUESTS	1.00	0.03	N/C	0.50	LIST PREFETCH READS	0.00	0.00	N/C	0.00
PARALL.QUERY REQ.REDUCTION	0.00	0.00	N/C	0.00	PAGES READ VIA LIST PREFETCH	0.00	0.00	N/C	0.00
PREF.QUANT.REDUCED TO 1/2	0.00	0.00	N/C	0.00	L.PRF.PAGES READ/L.PRF.READ	N/C			
PREF.QUANT.REDUCED TO 1/4	0.00	0.00	N/C	0.00	DYNAMIC PREFETCH REQUESTED	19473.00	573.82	N/C	9736.50
					DYNAMIC PREFETCH READS	18904.00	557.05	N/C	9452.00
					PAGES READ VIA DYN.PREFETCH	588.3K	17.3K	N/C	294.2K
					D.PRF.PAGES READ/D.PRF.READ	31.12			
					PREF.DISABLED-NO BUFFER	0.00	0.00	N/C	0.00
					PREF.DISABLED-NO READ ENG	0.00	0.00	N/C	0.00
					SYNC.HPOOL READ	0.00	0.00	N/C	0.00
					ASYN.HPOOL READ	0.00	0.00	N/C	0.00
					HPOOL READ FAILED	0.00	0.00	N/C	0.00
					ASYN.DA.MOVER HPOOL READ-S	0.00	0.00	N/C	0.00
					ASYN.DA.MOVER HPOOL READ-F	0.00	0.00	N/C	0.00
					PAGE-INS REQUIRED FOR READ	128.00	3.77	N/C	64.00

1 LOCATION: USIBMT6BOAPLX
 GROUP: BOAG
 MEMBER: NB22
 SUBSYSTEM: NB22
 DB2 VERSION: V5

DB2 PERFORMANCE MONITOR (V5)
 STATISTICS REPORT - LONG
 SCOPE: MEMBER

PAGE: 2-18
 REQUESTED FROM: 02/12/99 15:06:03.00
 TO: 02/12/99 15:44:05.00
 INTERVAL FROM: 02/12/99 15:09:49.64
 TO: 02/12/99 15:43:45.80

----- HIGHLIGHTS -----

INTERVAL START : 02/12/99 15:09:49.64 SAMPLING START: 02/12/99 15:09:49.64 TOTAL THREADS : 0.00
 INTERVAL END : 02/12/99 15:43:45.80 SAMPLING END : 02/12/99 15:43:45.80 TOTAL COMMITS : 2.00
 INTERVAL ELAPSED: 33:56.157107 OUTAGE ELAPSED: 0.000000 DATA SHARING MEMBER: N/A

TOT4K	WRITE OPERATIONS	QUANTITY	/MINUTE	/THREAD	/COMMIT	TOT4K	SORT/MERGE	QUANTITY	/MINUTE	/THREAD	/COMMIT
	BUFFER UPDATES	46.00	1.36	N/C	23.00	MAX WORKFILES CONCURR. USED		0.00	N/A	N/A	N/A
	PAGES WRITTEN	0.00	0.00	N/C	0.00	MERGE PASSES REQUESTED		0.00	0.00	N/C	0.00
	BUFF.UPDATES/PAGES WRITTEN	N/C				MERGE PASS DEGRADED-LOW BUF		0.00	0.00	N/C	0.00
						WORKFILE REQ.REJCTD-LOW BUF		0.00	0.00	N/C	0.00
	SYNCHRONOUS WRITES	0.00	0.00	N/C	0.00	WORKFILE REQ-ALL MERGE PASS		0.00	0.00	N/C	0.00
	ASYNCHRONOUS WRITES	0.00	0.00	N/C	0.00	WORKFILE NOT CREATED-NO BUF		0.00	0.00	N/C	0.00
						WORKFILE PRF NOT SCHEDULED		0.00	0.00	N/C	0.00
	PAGES WRITTEN PER WRITE I/O	N/C				WORKFILE PAGES TO DESTRUCT		4.00	0.12	N/C	2.00
						WORKFILE PAGES NOT WRITTEN		4.00	0.12	N/C	2.00
	HORIZ.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	VERTI.DEF.WRITE THRESHOLD	0.00	0.00	N/C	0.00						
	DM CRITICAL THRESHOLD	0.00	0.00	N/C	0.00						
	WRITE ENGINE NOT AVAILABLE	0.00	0.00	N/C	0.00						
	SYNC.HPOOL WRITE	0.00	0.00	N/C	0.00						
	ASYN.HPOOL WRITE	60274.00	1776.11	N/C	30.1K						
	HPOOL WRITE FAILED	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-S	0.00	0.00	N/C	0.00						
	ASYN.DA.MOVER HPOOL WRITE-F	0.00	0.00	N/C	0.00						
	PAGE-INS REQUIRED FOR WRITE	0.00	0.00	N/C	0.00						

Appendix E. Disk Storage Server Reports

RMF EXTRACT REPORTS

CHANNEL PATH ACTIVITY

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41 CYCLE 1.000 SECONDS

IODF = 29 NO CREATION INFORMATION AVAILABLE ACT: POR MODE: BASIC CPMF: AVAILABLE

CHANNEL PATH			UTILIZATION(%)		CHANNEL PATH			UTILIZATION(%)		CHANNEL PATH			UTILIZATION(%)		
ID	TYPE	SHR PARTITION	TOTAL	ID	TYPE	SHR PARTITION	TOTAL	ID	TYPE	SHR PARTITION	TOTAL	ID	TYPE	SHR PARTITION	TOTAL
00	CNC_S		0.00	08	CNC_S		13.87	10	CNC_S		0.00				
01	CNC_S		0.02	09	CNC_S		0.02	11	CNC_S		0.02				
02	CNC_S		0.02	0A	CNC_S		12.30	12	CNC_S		13.88				
03	CNC_S		0.02	0B	CNC_S		0.05	13	CNC_S		0.03				
04	CNC_S		0.02	0C	CNC_S		0.00	14	CNC_S		0.02				
05	CNC_S		0.01	0D		OFFLINE		15	CNC_S		12.44				
06	CNC_S		0.00	0E	CNC_S		0.00	16	CNC_S		0.01				
07	CNC_S		13.75	0F	CNC_S		0.00	17	CNC_S		0.00				
1E	CNC_S		13.66	26	CNC_S		0.03	2E	CNC_S		0.03				
1F	CNC_S		0.03	27	CNC_S		0.02	2F	CNC_S		0.01				
34		OFFLINE		8D	CNC_S		0.02	95	CNC_S		12.24				
80		OFFLINE		8E	CNC_S		0.01	96	CNC_S		0.06				
84	OSA		0.02	8F	CNC_S		12.28	97	CNC_S		0.02				
88	CNC_S		0.00	90	CNC_S		0.03	98	CNC_S		0.03				
89	CNC_S		0.02	91	CNC_S		13.87	99	CNC_S		0.51				
8A	CNC_S		0.02	92	CNC_S		0.01	9A	CNC_S		0.00				
8B	CNC_S		13.63	93	CNC_S		0.02	9B	CNC_S		0.00				
8C	CNC_S		12.31	94	CNC_S		0.02	9C	CNC_S		0.04				
C0	CNC_S		0.02	C8	CNC_S		13.83	D0	CNC_S		13.63				
C1	CNC_S		12.34	C9	CNC_S		0.00	D1	CNC_S		0.03				
C2	CNC_S		0.03	CA	CNC_S		0.00	D2	CNC_S		12.24				
C3	CNC_S		12.33	CB	CNC_S		0.00	D3	CNC_S		0.01				

I/O Q U E U I N G A C T I V I T Y

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41 CYCLE 1.000 SECONDS
 TOTAL SAMPLES = 2281 IOP ACTIVITY RATE AVG Q LNPTH IODF = 29 NO CREATION INFORMATION AVAILABLE ACT: POR

LCU	CONTENTION RATE	DELAY Q LNPTH	% ALL CH PATH BUSY	CONTROL UNITS	CHAN PATHS	CHPID TAKEN	% DP BUSY	% CU BUSY
			00	111.094	0.00			
			01	17.684	0.00			
0046	0.000	0.00	0.00	2B00	07	1.988	0.07	0.15
					08	1.980	0.11	0.11
					12	1.985	0.04	0.02
					8B	1.969	0.09	0.16
				2B01	91	1.961	0.13	0.24
					1E	1.967	0.04	0.31
					C8	1.966	0.07	0.16
					D0	1.964	0.07	0.24
0047	0.000	0.00	0.00	2B40	07	1.092	0.04	0.12
					08	1.093	0.12	0.16
					12	1.087	0.08	0.08
					8B	1.095	0.16	0.20
				2B41	91	1.092	0.12	0.24
					1E	1.103	0.08	0.20
					C8	1.097	0.12	0.12
					D0	1.092	0.04	0.16
0048	0.000	0.00	0.00	2B80	07	1.766	0.05	0.12
					08	1.759	0.15	0.20
					12	1.771	0.10	0.17
					8B	1.774	0.07	0.25
				2B81	91	1.766	0.10	0.20
					1E	1.777	0.07	0.20
					C8	1.766	0.05	0.15
					D0	1.765	0.12	0.07
0049	0.000	0.00	0.00	2BC0	07	0.981	0.04	0.09
					08	1.002	0.09	0.22
					12	0.981	0.09	0.22
					8B	0.994	0.13	0.22
				2BC1	91	0.997	0.18	0.26
					1E	0.986	0.09	0.22
					C8	0.996	0.00	0.09
					D0	0.987	0.04	0.13
004A	0.000	0.00	0.00	2C00	0A	0.464	0.00	0.19
					8C	0.456	0.10	0.19
					95	0.463	0.09	0.19
					C3	0.466	0.09	0.28
				2C01	15	0.463	0.19	0.09
					8F	0.471	0.18	0.46
					C1	0.471	0.00	0.19
					D2	0.466	0.09	0.19
004B	0.000	0.00	0.00	2C40	0A	1.807	0.07	0.19
					8C	1.811	0.12	0.17
					95	1.794	0.12	0.19
					C3	1.804	0.02	0.10
				2C41	15	1.804	0.02	0.10
					8F	1.810	0.05	0.07
					C1	1.804	0.10	0.22
					D2	1.797	0.05	0.17

004C	0.000	0.00	0.00	2C80	0A	1.801	0.05	0.29
					8C	1.832	0.02	0.33
					95	1.811	0.12	0.10
					C3	1.811	0.05	0.31
				2C81	15	1.826	0.17	0.12
					8F	1.822	0.12	0.26
					C1	1.819	0.07	0.17
					D2	1.816	0.05	0.38
004D	0.000	0.00	0.00	2CC0	0A	1.464	0.00	0.27
					8C	1.452	0.06	0.36
					95	1.463	0.00	0.24
					C3	1.459	0.03	0.15
				2CC1	15	1.465	0.00	0.12
					8F	1.452	0.09	0.36
					C1	1.456	0.00	0.24
					D2	1.459	0.09	0.30
					C2	0.082	0.00	0.00

RVA 1

1

C A C H E S U B S Y S T E M A C T I V I T Y

PAGE 1

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41

SUBSYSTEM 3990-03 CU-ID 2B00 SSID 0088 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
 TYPE-MODEL 9393-002

 CACHE SUBSYSTEM STATUS

SUBSYSTEM STORAGE		NON-VOLATILE STORAGE		STATUS	
CONFIGURED	1024.0M	CONFIGURED	8.0M	CACHING	- ACTIVE
AVAILABLE	1024.0M	PINNED	0.0	NON-VOLATILE STORAGE	- ACTIVE
PINNED	0.0			CACHE FAST WRITE	- ACTIVE
OFFLINE	0.0			IML DEVICE AVAILABLE	- YES

 CACHE SUBSYSTEM OVERVIEW

TOTAL I/O 41324 CACHE I/O 41324 CACHE OFFLINE 0
 TOTAL H/R 0.991 CACHE H/R 0.991

CACHE I/O REQUESTS	-----READ I/O REQUESTS-----					-----WRITE I/O REQUESTS-----					%		
	COUNT	RATE	HITS	RATE	H/R	COUNT	RATE	FAST	RATE	HITS		RATE	H/R
NORMAL	892	0.4	547	0.2	0.613	0	0.0	0	0.0	0	0.0	N/A	100.0
SEQUENTIAL	40432	17.8	40392	17.7	0.999	0	0.0	0	0.0	0	0.0	N/A	100.0
CFW DATA	0	0.0	0	0.0	N/A	0	0.0	0	0.0	0	0.0	N/A	N/A
TOTAL	41324	18.2	40939	18.0	0.991	0	0.0	0	0.0	0	0.0	N/A	100.0

REQUESTS	-----CACHE MISSES-----						-----MISC-----		-----NON-CACHE I/O-----			
	READ	RATE	WRITE	RATE	TRACKS	RATE	COUNT	RATE	COUNT	RATE		
NORMAL	345	0.2	0	0.0	385	0.2	DFW BYPASS	0	0.0	ICL	0	0.0
SEQUENTIAL	40	0.0	0	0.0	93457	41.1	CFW BYPASS	0	0.0	BYPASS	0	0.0
CFW DATA	0	0.0	0	0.0			DFW INHIBIT	0	0.0	TOTAL	0	0.0
							ASync (TRKS)	0	0.0			

TOTAL 385 RATE 0.2

----CKD STATISTICS--- ---RECORD CACHING---

OWRITE 0 READ MISSES 0
 WRITE HITS 0 WRITE PROM 0

1

C A C H E S U B S Y S T E M A C T I V I T Y

PAGE 2

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41

OSUBSYSTEM 3990-03 CU-ID 2B00 SSID 0088 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
 TYPE-MODEL 9393-002

 CACHE SUBSYSTEM DEVICE OVERVIEW

OVOLUME	DEV	DUAL	%	I/O	---CACHE HIT RATE--			-----DASD I/O RATE-----					ASync	TOTAL	READ	WRITE	%
SERIAL	NUM	COPY	I/O	RATE	READ	DFW	CFW	STAGE	DFWBP	ICL	BYP	OTHER	RATE	H/R	H/R	H/R	READ
0*ALL			100.0	18.2	18.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.991	0.991	N/A	100.0

*

1

CACHE SUBSYSTEM ACTIVITY

PAGE 1

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
0SUBSYSTEM 3990-03 CU-ID 2B40 SSID 0089 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
TYPE-MODEL 9393-002

0

CACHE SUBSYSTEM STATUS

0SUBSYSTEM STORAGE NON-VOLATILE STORAGE STATUS
0CONFIGURED 1024.0M CONFIGURED 8.0M CACHING - ACTIVE
AVAILABLE 1024.0M PINNED 0.0 NON-VOLATILE STORAGE - ACTIVE
PINNED 0.0 CACHE FAST WRITE - ACTIVE
OFFLINE 0.0 IML DEVICE AVAILABLE - YES

0

CACHE SUBSYSTEM OVERVIEW

0TOTAL I/O 22902 CACHE I/O 22902 CACHE OFFLINE 0
TOTAL H/R 0.999 CACHE H/R 0.999

-CACHE I/O-----READ I/O REQUESTS-----						-----WRITE I/O REQUESTS-----						%	
REQUESTS	COUNT	RATE	HITS	RATE	H/R	COUNT	RATE	FAST	RATE	HITS	RATE	H/R	READ
0NORMAL	169	0.1	152	0.1	0.899	0	0.0	0	0.0	0	0.0	N/A	100.0
SEQUENTIAL	22733	10.0	22730	10.0	1.000	0	0.0	0	0.0	0	0.0	N/A	100.0
CFW DATA	0	0.0	0	0.0	N/A	0	0.0	0	0.0	0	0.0	N/A	N/A
0TOTAL	22902	10.1	22882	10.1	0.999	0	0.0	0	0.0	0	0.0	N/A	100.0

-----CACHE MISSES-----							-----MISC-----			-----NON-CACHE I/O-----		
REQUESTS	READ	RATE	WRITE	RATE	TRACKS	RATE	COUNT	RATE	COUNT	RATE	COUNT	RATE
							DFW BYPASS	0	0.0	ICL	0	0.0
NORMAL	17	0.0	0	0.0	20	0.0	CFW BYPASS	0	0.0	BYPASS	0	0.0
SEQUENTIAL	3	0.0	0	0.0	52353	23.0	DFW INHIBIT	0	0.0	TOTAL	0	0.0
CFW DATA	0	0.0	0	0.0			ASYN (TRKS)	0	0.0			
0TOTAL	20	RATE	0	0.0								

----CKD STATISTICS--- ---RECORD CACHING---

0WRITE 0 READ MISSES 0
WRITE HITS 0 WRITE PROM 0

1

CACHE SUBSYSTEM ACTIVITY

PAGE 2

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
0SUBSYSTEM 3990-03 CU-ID 2B40 SSID 0089 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
TYPE-MODEL 9393-002

0

CACHE SUBSYSTEM DEVICE OVERVIEW

0VOLUME	DEV	DUAL	%	I/O	---CACHE HIT RATE--			-----DASD I/O RATE-----					ASYN	TOTAL	READ	WRITE	%
SERIAL	NUM	COPY	I/O	RATE	READ	DFW	CFW	STAGE	DFWBP	ICL	BY	OTHER	RATE	H/R	H/R	H/R	READ
0*ALL			100.0	10.1	10.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.999	0.999	N/A	100.0
*CACHE-OFF			0.0	0.0													
*CACHE			100.0	10.1	10.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.999	0.999	N/A	100.0

CACHE SUBSYSTEM ACTIVITY

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2B80 SSID 008A CDATE 02/12/1999 CTIME 15.05.41 CINT 00.37.56
 TYPE-MODEL 9393-002

CACHE SUBSYSTEM STATUS

OSUBSYSTEM STORAGE	NON-VOLATILE STORAGE	STATUS
CONFIGURED 1024.0M	CONFIGURED 8.0M	CACHING - ACTIVE
AVAILABLE 1024.0M	PINNED 0.0	NON-VOLATILE STORAGE - ACTIVE
PINNED 0.0		CACHE FAST WRITE - ACTIVE
OFFLINE 0.0		IML DEVICE AVAILABLE - YES

CACHE SUBSYSTEM OVERVIEW

TOTAL I/O	CACHE I/O	CACHE OFFLINE											
36324	36324	0											
TOTAL H/R 0.997	CACHE H/R 0.997												
-CACHE I/O						-WRITE I/O REQUESTS-						%	
REQUESTS	COUNT	RATE	HITS	RATE	H/R	COUNT	RATE	FAST	RATE	HITS	RATE	H/R	READ
ONORMAL	5668	2.5	5556	2.4	0.980	0	0.0	0	0.0	0	0.0	N/A	100.0
SEQUENTIAL	30656	13.5	30643	13.5	1.000	0	0.0	0	0.0	0	0.0	N/A	100.0
CFW DATA	0	0.0	0	0.0	N/A	0	0.0	0	0.0	0	0.0	N/A	N/A
OTOTAL	36324	16.0	36199	15.9	0.997	0	0.0	0	0.0	0	0.0	N/A	100.0
-CACHE MISSES-						-MISC-		-NON-CACHE I/O-					
REQUESTS	READ	RATE	WRITE	RATE	TRACKS	RATE	COUNT	RATE			COUNT	RATE	
							DFW BYPASS	0	0.0	ICL	0	0.0	
NORMAL	112	0.0	0	0.0	125	0.1	CFW BYPASS	0	0.0	BYPASS	0	0.0	
SEQUENTIAL	13	0.0	0	0.0	69055	30.3	DFW INHIBIT	0	0.0	TOTAL	0	0.0	
CFW DATA	0	0.0	0	0.0			ASYN (TRKS)	0	0.0				
OTOTAL	125	RATE	0.1										
---CKD STATISTICS---			---RECORD CACHING---										
OWRITE	0	READ MISSES	0										
WRITE HITS	0	WRITE PROM	0										

CACHE SUBSYSTEM ACTIVITY

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2B80 SSID 008A CDATE 02/12/1999 CTIME 15.05.41 CINT 00.37.56
 TYPE-MODEL 9393-002

CACHE SUBSYSTEM DEVICE OVERVIEW

OVOLUME	DEV	DUAL	%	I/O	---CACHE HIT RATE--			-----DASD I/O RATE-----					ASYN	TOTAL	READ	WRITE	%
SERIAL	NUM	COPY	I/O	RATE	READ	DFW	CFW	STAGE	DFWBP	ICL	BYP	OTHER	RATE	H/R	H/R	H/R	READ
0*ALL			100.0	16.0	15.9	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.997	0.997	N/A	100.0
*CACHE-OFF			0.0	0.0													
*CACHE			100.0	16.0	15.9	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.997	0.997	N/A	100.0

CACHE SUBSYSTEM ACTIVITY

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2BC0 SSID 008B CDATE 02/12/1999 CTIME 15.05.41 CINT 00.37.56
 TYPE-MODEL 9393-002

0-----

CACHE SUBSYSTEM STATUS

OSUBSYSTEM STORAGE		NON-VOLATILE STORAGE		STATUS	
0CONFIGURED	1024.0M	CONFIGURED	8.0M	CACHING	- EXPLICIT HOST TERMINATION
AVAILABLE	1024.0M	PINNED	0.0	NON-VOLATILE STORAGE	- ACTIVE
PINNED	0.0			CACHE FAST WRITE	- ACTIVE
OFFLINE	0.0			IML DEVICE AVAILABLE	- YES

RVA 2

1

C A C H E S U B S Y S T E M A C T I V I T Y

PAGE 1

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2C00 SSID 2007 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
 TYPE-MODEL 9393-002

CACHE SUBSYSTEM STATUS

OSUBSYSTEM STORAGE	NON-VOLATILE STORAGE	STATUS
CONFIGURED 1280.0M	CONFIGURED 8.0M	CACHING - ACTIVE
AVAILABLE 1280.0M	PINNED 0.0	NON-VOLATILE STORAGE - ACTIVE
PINNED 0.0		CACHE FAST WRITE - ACTIVE
OFFLINE 0.0		IML DEVICE AVAILABLE - YES

CACHE SUBSYSTEM OVERVIEW

TOTAL I/O		CACHE I/O		CACHE OFFLINE		WRITE I/O REQUESTS		MISC		NON-CACHE I/O			
REQUESTS	COUNT	RATE	HITS	RATE	H/R	COUNT	RATE	FAST	RATE	HITS	RATE	H/R	READ
TOTAL H/R	0.999	CACHE H/R	0.999										
NORMAL	158	0.1	152	0.1	0.962	0	0.0	0	0.0	0	0.0	N/A	100.0
SEQUENTIAL	9447	4.2	9447	4.2	1.000	0	0.0	0	0.0	0	0.0	N/A	100.0
CFW DATA	0	0.0	0	0.0	N/A	0	0.0	0	0.0	0	0.0	N/A	N/A
TOTAL	9605	4.2	9599	4.2	0.999	0	0.0	0	0.0	0	0.0	N/A	100.0
REQUESTS	READ	RATE	WRITE	RATE	TRACKS	RATE							
NORMAL	6	0.0	0	0.0	6	0.0	DFW BYPASS	0	0.0	ICL			0 0.0
SEQUENTIAL	0	0.0	0	0.0	21789	9.6	CFW BYPASS	0	0.0	BYPASS			0 0.0
CFW DATA	0	0.0	0	0.0			DFW INHIBIT	0	0.0	TOTAL			0 0.0
TOTAL	6	RATE	0.0				ASYNC (TRKS)	0	0.0				

---CKD STATISTICS--- ---RECORD CACHING---

OWRITE 0 READ MISSES 0
 WRITE HITS 0 WRITE PROM 0

1

C A C H E S U B S Y S T E M A C T I V I T Y

PAGE 2

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2C00 SSID 2007 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
 TYPE-MODEL 9393-002

CACHE SUBSYSTEM DEVICE OVERVIEW

OVOLUME	DEV	DUAL	%	I/O	---CACHE HIT RATE--			-----DASD I/O RATE-----					ASYNC	TOTAL	READ	WRITE	%
SERIAL	NUM	COPY	I/O	RATE	READ	DFW	CFW	STAGE	DFWBP	ICL	BYP	OTHER	RATE	H/R	H/R	H/R	READ
0*ALL			100.0	4.2	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.999	0.999	N/A	100.0
*CACHE-OFF			0.0	0.0													
*CACHE			100.0	4.2	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.999	0.999	N/A	100.0

```

OS/390                SYSTEM ID QP02                START 02/12/1999-15.05.39  INTERVAL 000.38.01
REL. 02.06.00        RPT VERSION 2.6.0            END 02/12/1999-15.43.41
OSUBSYSTEM 3990-03  CU-ID 2C40  SSID 2008  CDATE 02/12/1999  CTIME 15.05.41  CINT 00.37.56
TYPE-MODEL 9393-002

```

0

CACHE SUBSYSTEM STATUS

```

OSUBSYSTEM STORAGE      NON-VOLATILE STORAGE      STATUS
OCONFIGURED 1280.0M     CONFIGURED 8.0M          CACHING - ACTIVE
AVAILABLE 1280.0M     PINNED 0.0              NON-VOLATILE STORAGE - ACTIVE
PINNED 0.0                CACHE FAST WRITE - ACTIVE
OFFLINE 0.0                IML DEVICE AVAILABLE - YES

```

0

CACHE SUBSYSTEM OVERVIEW

```

OTOTAL I/O 36418  CACHE I/O 36418  CACHE OFFLINE 0
TOTAL H/R 0.996  CACHE H/R 0.996
-CACHE I/O-----READ I/O REQUESTS-----WRITE I/O REQUESTS-----%
REQUESTS  COUNT  RATE  HITS  RATE  H/R  COUNT  RATE  FAST  RATE  HITS  RATE  H/R  READ
ONORMAL  9460  4.2  9340  4.1  0.987  0  0.0  0  0.0  0  0.0  N/A  100.0
SEQUENTIAL 26958 11.8 26950 11.8 1.000  0  0.0  0  0.0  0  0.0  N/A  100.0
CFW DATA  0  0.0  0  0.0  N/A  0  0.0  0  0.0  0  0.0  N/A  N/A
OTOTAL  36418 16.0 36290 15.9 0.996  0  0.0  0  0.0  0  0.0  N/A  100.0
-----CACHE MISSES-----MISC-----NON-CACHE I/O-----
REQUESTS  READ  RATE  WRITE  RATE  TRACKS  RATE  COUNT  RATE  COUNT  RATE
NORMAL  120  0.1  0  0.0  128  0.1  DFW BYPASS  0  0.0  ICL  0  0.0
SEQUENTIAL 8  0.0  0  0.0  60922 26.8  CFW BYPASS  0  0.0  BYPASS  0  0.0
CFW DATA  0  0.0  0  0.0  ASYNC (TRKS) 0  0.0  TOTAL  0  0.0
OTOTAL  128  RATE  0.1
----CKD STATISTICS--- ---RECORD CACHING---
OWRITE 0  READ MISSES 0
WRITE HITS 0  WRITE PROM 0

```

1

```

OS/390                SYSTEM ID QP02                START 02/12/1999-15.05.39  INTERVAL 000.38.01
REL. 02.06.00        RPT VERSION 2.6.0            END 02/12/1999-15.43.41
OSUBSYSTEM 3990-03  CU-ID 2C40  SSID 2008  CDATE 02/12/1999  CTIME 15.05.41  CINT 00.37.56
TYPE-MODEL 9393-002

```

0

CACHE SUBSYSTEM DEVICE OVERVIEW

```

OVOLUME  DEV  DUAL  %  I/O  ---CACHE HIT RATE--  -----DASD I/O RATE-----  ASYNC  TOTAL  READ  WRITE  %
SERIAL  NUM  COPY  I/O  RATE  READ  DFW  CFW  STAGE  DFWBP  ICL  BYP  OTHER  RATE  H/R  H/R  H/R  READ
O*ALL  100.0  16.0  15.9  0.0  0.0  0.1  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.996  0.996  N/A  100.0
*CACHE-OFF  0.0  0.0
*CACHE  100.0  16.0  15.9  0.0  0.0  0.1  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.996  0.996  N/A  100.0

```

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2CA0 SSID 2009 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
 TYPE-MODEL 9393-002

 CACHE SUBSYSTEM STATUS

OSUBSYSTEM STORAGE	NON-VOLATILE STORAGE	STATUS
CONFIGURED 1280.0M	CONFIGURED 8.0M	CACHING - ACTIVE
AVAILABLE 1280.0M	PINNED 0.0	NON-VOLATILE STORAGE - ACTIVE
PINNED 0.0		CACHE FAST WRITE - ACTIVE
OFFLINE 0.0		IML DEVICE AVAILABLE - YES

 CACHE SUBSYSTEM OVERVIEW

TOTAL I/O	38155	CACHE I/O	38155	CACHE OFFLINE	0									
TOTAL H/R	1.000	CACHE H/R	1.000											
-CACHE I/O	-----READ I/O REQUESTS-----				-----WRITE I/O REQUESTS-----				%					
REQUESTS	COUNT	RATE	HITS	RATE	H/R	COUNT	RATE	FAST	RATE	HITS	RATE	H/R	READ	
NORMAL	54	0.0	47	0.0	0.870	0	0.0	0	0.0	0	0.0	N/A	100.0	
SEQUENTIAL	38101	16.7	38098	16.7	1.000	0	0.0	0	0.0	0	0.0	N/A	100.0	
CFW DATA	0	0.0	0	0.0	N/A	0	0.0	0	0.0	0	0.0	N/A	N/A	
OTOTAL	38155	16.8	38145	16.8	1.000	0	0.0	0	0.0	0	0.0	N/A	100.0	
REQUESTS	READ	RATE	WRITE	RATE	TRACKS	RATE	-----MISC-----		-----NON-CACHE I/O-----		COUNT	RATE	COUNT	RATE
							DFW BYPASS	0	0.0	ICL			0	0.0
NORMAL	7	0.0	0	0.0	10	0.0	CFW BYPASS	0	0.0	BYPASS			0	0.0
SEQUENTIAL	3	0.0	0	0.0	87879	38.6	DFW INHIBIT	0	0.0	TOTAL			0	0.0
CFW DATA	0	0.0	0	0.0			ASYNC (TRKS)	0	0.0					
OTOTAL	10	RATE	0.0											
----CKD STATISTICS---	---RECORD CACHING---													
OWRITE	0	READ MISSES	0											
WRITE HITS	0	WRITE PROM	0											

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 2CA0 SSID 2009 CDATE 02/12/1999 CTIME 15.05.40 CINT 00.37.56
 TYPE-MODEL 9393-002

 CACHE SUBSYSTEM DEVICE OVERVIEW

OVOLUME	DEV	DUAL	%	I/O	---CACHE HIT RATE--	-----DASD I/O RATE-----	ASYNC	TOTAL	READ	WRITE	%						
SERIAL	NUM	COPY	I/O	RATE	READ	DFW	CFW	STAGE	DFWBP	ICL	BYP	OTHER	RATE	H/R	H/R	H/R	READ
*ALL			100.0	16.8	16.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000	1.000	N/A	100.0
*CACHE-OFF			0.0	0.0													
*CACHE			100.0	16.8	16.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000	1.000	N/A	100.0

SYSTEM

1

CACHE SUBSYSTEM ACTIVITY

PAGE 1

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 71C0 SSID 603C CDATE 02/12/1999 CTIME 15.05.41 CINT 00.37.56
 TYPE-MODEL 3990-006

CACHE SUBSYSTEM STATUS

OSUBSYSTEM STORAGE NON-VOLATILE STORAGE STATUS
 OCONFIGURED 256.0M CONFIGURED 32.0M CACHING - ACTIVE
 AVAILABLE 254.9M PINNED 0.0 NON-VOLATILE STORAGE - ACTIVE
 PINNED 0.0 CACHE FAST WRITE - ACTIVE
 OFFLINE 0.0 IML DEVICE AVAILABLE - YES

CACHE SUBSYSTEM OVERVIEW

OTOTAL I/O 50278 CACHE I/O 50276 CACHE OFFLINE 0
 TOTAL H/R 0.999 CACHE H/R 0.999
 -CACHE I/O -READ I/O REQUESTS- -WRITE I/O REQUESTS- %
 REQUESTS COUNT RATE HITS RATE H/R COUNT RATE FAST RATE HITS RATE H/R READ
 ONORMAL 28486 12.5 28418 12.5 0.998 20030 8.8 20030 8.8 20030 8.8 1.000 58.7
 SEQUENTIAL 32 0.0 32 0.0 1.000 1728 0.8 1728 0.8 1728 0.8 1.000 1.8
 CFW DATA 0 0.0 0 0.0 N/A 0 0.0 0 0.0 0 0.0 N/A N/A
 OTOTAL 28518 12.5 28450 12.5 0.998 21758 9.6 21758 9.6 21758 9.6 1.000 56.7

-----CACHE MISSES----- -MISC----- -NON-CACHE I/O-----
 REQUESTS READ RATE WRITE RATE TRACKS RATE COUNT RATE COUNT RATE COUNT RATE
 NORMAL 68 0.0 0 0.0 68 0.0 DFW BYPASS 0 0.0 ICL 0 0.0
 SEQUENTIAL 0 0.0 0 0.0 0 0.0 CFW BYPASS 0 0.0 BYPASS 2 0.0
 CFW DATA 0 0.0 0 0.0 DFW INHIBIT 0 0.0 TOTAL 2 0.0
 OTOTAL 68 RATE 0.0 ASYNC (TRKS) 1066 0.5

---CKD STATISTICS--- ---RECORD CACHING---

OWRITE 1 READ MISSES 68
 WRITE HITS 1 WRITE PROM 9473

1

CACHE SUBSYSTEM ACTIVITY

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41
 OSUBSYSTEM 3990-03 CU-ID 71C0 SSID 603C CDATE 02/12/1999 CTIME 15.05.41 CINT 00.37.56
 TYPE-MODEL 3990-006

CACHE SUBSYSTEM DEVICE OVERVIEW

OVOLUME DEV DUAL % I/O ---CACHE HIT RATE-- -----DASD I/O RATE----- ASYNC TOTAL READ WRITE %
 SERIAL NUM COPY I/O RATE READ DFW CFW STAGE DFEBP ICL BYP OTHER RATE H/R H/R H/R READ
 O*ALL 100.0 22.1 12.5 9.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.999 0.998 1.000 56.7
 *CACHE-OFF 0.0 0.0
 *CACHE 100.0 22.1 12.5 9.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.999 0.998 1.000 56.7

1

DIRECT ACCESS DEVICE ACTIVITY

OS/390 SYSTEM ID QP02 START 02/12/1999-15.05.39 INTERVAL 000.38.01
 REL. 02.06.00 RPT VERSION 2.6.0 END 02/12/1999-15.43.41 CYCLE 1.000 SECONDS

TOTAL SAMPLES = 2,281 IODF = 29 NO CREATION INFORMATION AVAILABLE

STORAGE GROUP	DEV NUM	DEVICE TYPE	VOLUME SERIAL	LCU	ACTIVITY RATE	AVG RESP TIME	AVG IOSQ	AVG DPB DLY	AVG CUB DLY	AVG DB DLY	AVG PEND TIME	AVG DISC TIME	AVG CONN TIME	% DEV CONN	% DEV UTIL	% DEV RESV	AVG NUMBER ALLOC	% ANY ALLOC	% MT PEND
			LCU	0046	15.778	32	0	0.0	0.0	0.0	0.2	7.1	24.5	0.60	0.78	0.0	15.0	100.0	0.0
			LCU	0047	8.752	33	2	0.0	0.0	0.0	0.2	6.7	24.7	0.34	0.43	0.0	14.0	100.0	0.0
			LCU	0048	14.145	27	1	0.0	0.0	0.0	0.2	5.4	20.6	0.45	0.57	0.0	11.0	100.0	0.0
			LCU	0049	7.923	32	0	0.0	0.0	0.0	0.2	7.0	24.7	0.31	0.39	0.0	19.0	100.0	0.0
			LCU	004A	3.721	34	0	0.0	0.0	0.0	0.2	9.3	24.2	0.14	0.20	0.0	3.0	100.0	0.0
			LCU	004B	14.431	26	2	0.0	0.0	0.0	0.2	6.3	17.8	0.40	0.54	0.0	4.0	100.0	0.0
			LCU	004C	14.540	35	0	0.0	0.0	0.0	0.2	9.1	25.1	0.57	0.78	0.0	5.0	100.0	0.0
			LCU	004D	11.668	30	0	0.0	0.0	0.0	0.2	7.6	22.3	0.41	0.55	0.0	7.0	100.0	0.0
			LCU	0055	6.963	3	0	0.0	0.0	0.0	0.4	0.1	2.6	0.06	0.06	0.0	299	100.0	0.0
RVA1		SG			90.959	31	1	0.0	0.0	0.0	0.2	7.1	22.7	0.40	0.53	0.0	78.0	100.0	0.0

IXFP EXTRACT REPORTS

XSA/REPORTER

DEVICE PERFORMANCE OVERALL SUMMARY

SUBSYSTEM 20395

SUBSYSTEM	% DEV	I/O	KBYTES	ACCESS	-I/O SERVICE TIME (MS)-			% DEV	% DEV	% DEV
SUMMARY	AVAIL	PER SEC	PER SEC	DENSITY	TOTAL	DISC	CONNECT	UTIL	DISC	CONN
PROD PARTITION	100.0	45.7	5481.4	0.1	30.5	7.3	23.3	0.5	0.1	0.4
OVERALL TOTALS	100.0	45.7	5481.4	0.1	30.5	7.3	23.3	0.5	0.1	0.4

SUBSYSTEM 20395

DISK ARRAY SUMMARY

AVG % DRIVE MODULE UTIL	COEFF OF VARIATION	NET CAPACITY LOAD %			FREE SPACE COLLECTION LOAD			COLL FREE SPC (%)			UNCOLL FREE SPC (%)		
		TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
10.6	78	0.0	56.4	56.4	0.0	0.0	0.0	0.0	42.4	42.4	0.0	1.2	1.2

SUBSYSTEM 22897

SUBSYSTEM	% DEV	I/O	KBYTES	ACCESS	-I/O SERVICE TIME (MS)-			% DEV	% DEV	% DEV
SUMMARY	AVAIL	PER SEC	PER SEC	DENSITY	TOTAL	DISC	CONNECT	UTIL	DISC	CONN
PROD PARTITION	100.0	44.5	5036.8	0.1	29.7	7.8	22.0	0.5	0.1	0.4
OVERALL TOTALS	100.0	44.5	5036.8	0.1	29.7	7.8	22.0	0.5	0.1	0.4

SUBSYSTEM 22897

DISK ARRAY SUMMARY

AVG % DRIVE MODULE UTIL	COEFF OF VARIATION	NET CAPACITY LOAD %			FREE SPACE COLLECTION LOAD			COLL FREE SPC (%)			UNCOLL FREE SPC (%)		
		TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
13.2	86	0.0	25.2	25.2	0.0	0.0	0.0	0.0	73.9	73.9	0.0	0.9	0.9

CACHE EFFECTIVENESS OVERALL SUMMARY

SUBSYSTEM NAME: 20395 (CACHE SIZE: 1024 MB NVS SIZE: 8 MB)

SUBSYSTEM	READ	WRITE	I/O	READ	READ	WRITE	I/O	DFW	STAGE	HITS/	LOW	TRACK
SUMMARY	PER SEC	PER SEC	PER SEC	RATIO	HIT %	HIT %	HIT %	CONSTR	PER SEC	STGE	REF CT	OCCUP
	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
PROD PARTITION	53.8	0.0	45.7	61329	99.3	100.0	99.3	0.0	113.2	0.5	73.7	
OVERALL TOTALS	53.8	0.0	45.7	61329	99.3	100.0	99.3	0.0	113.2	0.5	73.7	25050

SUBSYSTEM NAME: 22897 (CACHE SIZE: 1280 MB NVS SIZE: 8 MB)

SUBSYSTEM	READ	WRITE	I/O	READ	READ	WRITE	I/O	DFW	STAGE	HITS/	LOW	TRACK
SUMMARY	PER SEC	PER SEC	PER SEC	RATIO	HIT %	HIT %	HIT %	CONSTR	PER SEC	STGE	REF CT	OCCUP
	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
PROD PARTITION	50.1	0.0	44.5	57064	99.9	100.0	99.9	0.0	103.0	0.5	73.5	
OVERALL TOTALS	50.1	0.0	44.5	57064	99.9	100.0	99.9	0.0	103.0	0.5	73.5	22086

SPACE UTILIZATION SUMMARY REPORT

SUBSYSTEM 20395 (NUMBER OF FUNCTIONAL DEVICES: 256)

SELECTED DEVICES SUMMARY	FUNCTIONAL CAPACITY (MB)				% FUNCT CAPACITY		DISK ARRAY			
	SELECTED DEVICES	TOTAL FUNCTIONAL CAPACITY (MB)	NOT STORED	NOT STORED	NOT STORED	NOT STORED	-- PHYSICAL CAP USED (MB) --			COMP
			STORED	STORED	STORED	STORED	SHARED	UNIQUE	TOTAL	RATIO
PRODUCTION PARTITION:	256	726532.2	204036.4	522495.8	28.1	71.9	0.0	65964.1	65964.1	3.1
TOTALS:	256	726532.2	204036.4	522495.8	28.1	71.9	0.0	65964.1	65964.1	3.1

SUBSYSTEM 20395 SPACE UTILIZATION SUMMARY

NUMBER OF FUNCTIONAL DEVICES	DISK ARRAY CAPACITY (MB)	NET CAPACITY LOAD (%)			COLL FREE SPACE (%)			UNCOLL FREE SPACE (%)		
		TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
256	117880.2	0.0	56.4	56.4	0.0	42.4	42.4	0.0	1.3	1.3

SUBSYSTEM 22897 (NUMBER OF FUNCTIONAL DEVICES: 256)

SELECTED DEVICES SUMMARY	FUNCTIONAL CAPACITY (MB)				% FUNCT CAPACITY		DISK ARRAY			
	SELECTED DEVICES	TOTAL FUNCTIONAL CAPACITY (MB)	NOT STORED	NOT STORED	NOT STORED	NOT STORED	-- PHYSICAL CAP USED (MB) --			COMP
			STORED	STORED	STORED	STORED	SHARED	UNIQUE	TOTAL	RATIO
PRODUCTION PARTITION:	256	726532.2	39125.9	687406.3	5.4	94.6	0.0	20157.3	20157.3	1.9
TOTALS:	256	726532.2	39125.9	687406.3	5.4	94.6	0.0	20157.3	20157.3	1.9

SUBSYSTEM 22897 SPACE UTILIZATION SUMMARY

NUMBER OF FUNCTIONAL DEVICES	DISK ARRAY CAPACITY (MB)	NET CAPACITY LOAD (%)			COLL FREE SPACE (%)			UNCOLL FREE SPACE (%)		
		TEST	PROD	OVERALL	TEST	PROD	OVERALL	TEST	PROD	OVERALL
256	81609.4	0.0	25.2	25.2	0.0	73.9	73.9	0.0	0.9	0.9

Appendix F. Special Notices

This publication is intended to help managers and professionals understand and evaluate the applicability of DFSMS/MVS functions to DB2 for OS/390. It also provides disk architecture background information in order to make management and control of DB2 data sets easier. The information in this publication is not intended as the specification of any programming interfaces that are provided by DB2 for OS/390 Version 5. See the PUBLICATIONS section of the IBM Programming Announcement for DB2 for OS/390 Version 5 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain 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.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

IBM ®	DB2
MVS/ESA	S/390
DFSMS/MVS	RAMAC
RMF	ESCON
IMS	CICS

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

IXFP and SnapShot are registered trademarks in the United States and/or other countries licensed exclusively through Storage Technology Corporation.

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

Appendix G. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

G.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 243.

- *DB2 for OS/390 and Data Compression*, SG24-5261
- *DB2 UDB for OS/390 Version 6 Performance Topics*, SG24-5351
- *DFSMSHsm Primer*, SG24-5272
- *DFSMS Optimizer: The New HSM Monitor/Tuner*, SG24-5248
- *Get DFSMS FIT: Fast Implementation Techniques*, SG24-2568
- *DFSMS FIT: Fast Implementation Techniques Process Guide*, SG24-4478
- *DFSMS FIT: Fast Implementation Techniques Installation Examples*, SG24-2569
- *DFSMS/MVS V1R4 Technical Overview*, SG24-4892
- *IBM RAMAC Virtual Array*, SG24-4951
- *IBM RAMAC 3 Array Storage*, SG24-4835
- *RAMAC Virtual Array: Implementing Peer-to-Peer Remote Copy*, SG24-5338
- *Using RVA and SnapShot for Business Intelligence Applications with OS/390 and DB2*, SG24-5333
- *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268

G.2 Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at <http://www.redbooks.ibm.com/> for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

G.3 Other Publications

These publications are also relevant as further information sources:

- *DB2 for OS/390 Version 5 Administration Guide*, SC26-8957
- *DB2 for OS/390 Version 5 Command Reference*, SC26-8960
- *DB2 for OS/390 Version 5 SQL Reference*, SC26-8966
- *DB2 for OS/390 Version 5 Utility Guide and Reference*, SC26-8967
- *MVS/ESA SML: Managing Data*, SC26-3124
- *MVS/ESA SML: Managing Storage Groups*, SC26-3125
- *DFSMS/MVS V1R4 Managing Catalogs*, SC26-4914
- *DFSMS/MVS V1R4 Implementing System-Managed Storage*, SC26-3123
- *DFSMS/MVS V1R4 DFSMSrmm Guide and Reference*, SC26-4931-05
- *DFSMS/MVS V1R4 DFSMSdfp Storage Administration Reference*, SC26-4920
- *DFSMS/MVS V1R3 NaviQuest User's Guide*, SC26-7194
- *DFSMS/MVS Optimizer V1R2 User's Guide and Reference*, SC26-7047-04
- *IBM RAMAC Array Subsystem Introduction*, GC26-7004
- *IBM RAMAC Virtual Array Storage Introduction*, GC26-7168
- *IXFP Configuration and Administration*, SC26-7178
- *IXFP Subsystem Reporting*, SC26-7184
- *OS/390 RMF Report Analysis*, SC28-1950
- *OS/390 V2 R6.0 MVS Diagnosis: Tools and Service Aids*, SY28-1085
- *OS/390 V2 R6.0 RMF User's Guide*, SC28-1949

G.4 Web Sites

These Web sites provide further up-to-date information sources:

- IBM Home Page:
<http://www.ibm.com/>
- ITSO Home Page:
<http://www.redbooks.ibm.com/>
- DB2 for OS/390 Home Page:
<http://www.software.ibm.com/data/db2/os390/>
- DB2 Family:
<http://www.software.ibm.com/data/db2>
- DB2 Family Performance:
<http://www.software.ibm.com/data/db2/performance>
- DFSMS/MVS Home Page:
<http://www.storage.ibm.com/software/sms/smshome.htm>
- DFSMS/MVS White Papers:
<http://www.storage.ibm.com/software/sms/smshome.htm>

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** <http://www.redbooks.ibm.com/>

Search for, view, download, or order hardcopy/CD-ROM redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the "How to Order" section at this site: http://www.elink.ibm.ibm.com/pbl/pbl/

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibm.ibm.com/pbl/pbl/

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: http://www.elink.ibm.ibm.com/pbl/pbl/

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

List of Abbreviations

ABARS	aggregate backup and recovery support	ECSA	extended common storage area
APAR	authorized program analysis report	EDM	environment descriptor management
ARM	automatic restart manager	ERP	enterprise resource planning
BLOB	binary large objects	ESA	Enterprise Systems Architecture
BSDS	boot strap data set	FBA	fixed block architecture
CCW	channel command word	GBP	group buffer pool
CEC	central electronics complex	GB	gigabyte (1,073,741,824 bytes)
CF	coupling facility	GDG	generation data group
CFRM	coupling facility resource management	GDPS	geographically dispersed parallel sysplex
CICS	Customer Information Control System	HDA	head disk assemblies
CLI	call level interface	HLQ	high level qualifier
CMOS	complementary metal oxide semiconductor	HSM	hierarchical storage manager
CPU	central processing unit	IBM	International Business Machines Corporation
CSA	common storage area	IC	internal coupling
DASD	direct access storage device	ICB	integrated cluster bus
DB2	DATABASE 2	ICF	integrated coupling facility
DB2 PM	DB2 performance monitor	ICMF	internal coupling migration facility
DBAT	database access thread	JDBC	Java Database Connectivity
DBD	database descriptor	IFCID	instrumentation facility component identifier
DBRM	database request module	IFI	instrumentation facility interface
DCL	data control language	IMS	Information Management System
DDCS	distributed database connection services	ISMF	Interactive Storage Management Facility
DDF	distributed data facility	ISPF	Interactive System Productivity Facility
DDL	data definition language	IRLM	internal resource lock manager
DFP	Data Facility Product	I/O	input/output
DFW	dasd fast write	ITSO	International Technical Support Organization
DL/1	Data Language/1	IWTH	immediate write threshold
DML	data manipulation language	IXFP	IBM Extended Facilities Product
DMTH	data management threshold	KB	kilobyte (1,024 bytes)
DRDA	distributed relational database architecture		
DSS	data set services		
DWQT	deferred write threshold		
EA	extended addressability		
ECS	enhanced catalog sharing		

KSDS	key-sequenced data set
LCU	logical control unit
LDS	linear data set
LLQ	low level qualifier
LPAR	logically partitioned mode
LRSN	log record sequence number
LRU	least recently used
MB	megabyte (1,048,576 bytes)
MVS	Multiple Virtual Storage
NVS	non-volatile storage
ODBC	Open Data Base Connectivity
OPT	optimizer
OS/390	Operating System/390
PDF	Program Development Facility (component of ISPF)
PDS	partitioned data set
PPRC	peer-to-peer remote copy
QMF	Query Management Facility
RAID	redundant array of independent disks
RACF	Resource Access Control Facility
RBA	relative byte address
RDS	relational data system
RID	record identifier
RMM	removable media manager
RS	read stability
RR	repeatable read
RVA	RAMAC virtual array
SDM	System Data Mover
SMF	System Management Facility
SMS	system managed storage
SPTH	sequential prefetch threshold
SSID	subsystem identifier
Sysplex	system complex
UCB	unit control block
UDB	universal database
VTOC	volume table of contents
VDWQT	vertical deferred write threshold
WLM	workload manager
XRC	extended remote copy

Index

Numerics

3380 85
3390 85

A

ABARS 30
abbreviations 245
accounting report 141
ACDS 35
acronyms 245
ACS 36
ACS routines 80
active control data set 35
active log 18
 sizing 18
active log data sets
 default names 23
active log size 117
ADDVOL 30
ARCHIVE 19
ARCHIVE command 69, 193
archive installation panel 69
archive log 19
archive log and SMS 69
archive log data set 116, 117
archive log data sets
 default names 24
archive logs 191
archive to disk or tape 116
array 85
asynchronous copy 99
asynchronous write 107
AUTO BACKUP 42
automatic class selection 36
availability management 30

B

backup 30
batch database 58
benefits of SMS 5, 32
block size 33
boot strap data set 17
boot strap data sets
 default names 23
BSDS and active logs 185
BSDS and SMS 67
buffer pools 103
bypass cache 92

C

cache 90, 103
cache effectiveness report . 137
cache hit 90
cache performance 90
case study 141

catalog 15
CESTPATH 93
channel program 89
COMMDS 36
communications data set 36
compression 100
concurrent copy 94
CONVERT 28
converting DB2 data to SMS 75
converting to SMS 28
CREATE DATABASE 164
CREATE INDEX 14
CREATE STOGROUP 164
CREATE TABLESPACE 14, 163, 164

D

DASD fast write 91
data class example for DB2 47
data locality 6
data management threshold 106
data spaces 103
data striping 101
Data Warehouse database 58
database 13
DB2
 accounting trace 119
 performance trace 119
 statistics trace 119
DB2 analysis 141
DB2 Catalog 60
DB2 Directory 60
DB2 I/O operations 103
DB2 PM 119
 accounting trace report 201
 statistics report 205
DB2 recovery data
 management class 64
 storage class 63
 storage group 65
DB2 recovery data sets
 test cases 185
DB2 STOGROUP 52
DB2 storage objects 11
DB2 system table spaces 15
DB2 work database 61
DCME 92
deferred write threshold 107
DEFINE CLUSTER 164
DFDSS 26
DFHSM 26
DFP 26
DFSMS 35
DFSMS FIT 81
DFSMS/MVS 25
DFSMSdfp 22, 26
DFSMSdss 27, 94
 COPY 94

- DUMP 94
- DFSMSshsm 28
- DFSMSopt 31
- DFSMSrmm 31
- DFW 91
- directory 15
- disk architecture 85
- DMTH 106
- DSN1COPY 22
- DSNDB01 15
- DSNDB01.SYSLGRNX 16
- DSNDB06 15
- DSNDB07 15
- DSNJLOGF 115
- DSNTIPD 15
- DSNTIPE 92
- DSNTIPL 112, 116, 117
- DSNTIPN 17, 117
- DSS 27
 - components 27
 - filtering 28
- dual copy 93
- DWQT 107, 108
- Dynamic Cache Management Enhanced 92
- dynamic prefetch 105

E

- ECKD 86
- ESCON 86, 93
- EXT 25
- extended remote copy 96

F

- FBA 86
- Fiber Connection Architecture 93

G

- GDPS 97
- geographically dispersed parallel sysplex 97
- guaranteed space 40, 44, 45, 46, 57, 77

H

- HDA 86
- high level qualifier 46, 54
- hiper pools 103
- HRECALL 30
- HSM 28

I

- I/O activity report 123
- I/O suspensions 121
- ICL 92
- IDCAMS 15, 39, 40, 79, 189
 - ALLOCATE 38
 - DEFINE 38, 41
 - DEFINE CLUSTER 14
 - LISTCAT 174

- SETCACHE 91
- image copies 194
- image copy 20
- image copy and SMS 71
- image copy data sets
 - naming convention 24
- image copy options 20
- immediate write threshold 108
- index 13
- index space 13
 - creation 13
- inhibit cache load 92, 128
- instrumentation facility component identifiers 123
- interval migration 30
- introduction 3
- ISMF 26
- ISMF test cases 171
- IWTH 108
- IXFP 89, 119, 245
 - report analysis 152
 - reporting 135
- IXFP view 152

L

- LCU 90
- least recently used algorithm 91
- list prefetch 105
- log activity report 122
- log data sets
 - preformatting 115
- log read 115
- log read performance 116
- log record
 - synchronous write 112
- log records 111
 - asynchronous write 112
- log write performance 114
- logical control unit 89
- LOGLOAD 109
- LOGONLY option 22
- low level qualifier 46
- LSF 87

M

- management class example for DB2 49
- ML2 42
- MODIFY 78

N

- naming convention 22
 - table spaces and index spaces 23
- NaviQuest for MVS 81
- NOMIGRATE 42
- nonvolatile storage 90
- NVS 90

O

- online database 58

OPT 31
OUTPUT BUFFER 114

P

partitioned data sets 22
path 93
peer to peer remote copy 93
peer-to-peer remote copy 96
policy 80
PPRC 93, 96
prefetch quantity 106

Q

queuing time 128
quickwrite 92

R

RAID 85
read operations 104
read record caching 91
recall 30
RECOVER 21, 95
recovery data sets 11, 17
recovery strategy 18
remote copy 95
REORG 79
response time 128
RMF 119
 cache reports 125
 CHAN report 125
 device report 125
 IOQ report 125
 report analysis 149
 report consolidation 132
 storage server reports 223
RMF tools 133
RMM 31
RVA 45

S

sample naming structure for image copies 24
SCDBARCH 191
SCDS 35
SDM 99
sequential caching 92
sequential data striping 101
sequential prefetch 104
sequential prefetch threshold 107
service level agreemen 78
service time 128
SETCACHE 91
SMF record type 42 32
SMS
 assigning classes to DB2 53
 base configuration 35
 classes 38
 coded names 174
 control data sets 35

 converting DB2 data 78
 data class 38
 DB2 recovery data sets 63
 DB2 system databases 60
 distribution of partitioned table spaces 178
 examples of constructs 46
 examples of table space management 47
 existing names 165
 imbedding codes into names of DB2 objects 56
 implementation prerequisites 77
 management class 41
 management of DB2 databases 47
 managing partitioned table space 56
 naming standard 46
 storage class 40
 storage group 43
 storage management policy 36
 user database types 57
 user distribution of partitioned table spaces 181

SMS benefits 75
SMS configuration 35
SMS management goals 76
SMS storage group 52
SnapShot 87
source control data set 35
space management 29
space utilization report 138
SPTH 107
storage class example for DB2 48
storage device 90
storage group 13, 43
 types 43
storage group example for DB2 50
Storage Management Facility 26
storage server 89
storage server analysis 147
striping 101
subsystem identifier 90
summary of considerations 5
SUSIBM.SYSCOPY 16
suspend 99
suspend time 145
synchronous copy 98
synchronous read 104
synchronous write 108
SYSIBM.SYSLGRNX 16
System Data Mover 99
system managed storage 25

T

table 12
table space 11, 12
 creation 13
 partition sizes 12
table space allocation using SMS 165
table spaces
 system 11
 user 11
test cases 161
test database 59

timestamp 100
track size 85
two-phase commit 112

U

user defined table space 14

V

VDWQT 107, 108
vertical deferred write threshold 107
virtual concurrent copy 95
virtual volumes 88
VTOC 26

W

work database 15
work table spaces
 page sizes 15
write frequency 108
write operations 107
write quantity 108
write record caching 92
WRITE THRESHOLD 114

X

XRC 99

ITSO Redbook Evaluation

Storage Management with DB2 for OS/390
SG24-5462-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

Customer **Business Partner** **Solution Developer** **IBM employee**
 None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

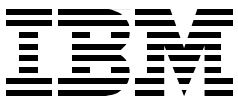
If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

SG24-5462-00

Printed in the U.S.A.



Free Manuals Download Website

<http://myh66.com>

<http://usermanuals.us>

<http://www.somanuals.com>

<http://www.4manuals.cc>

<http://www.manual-lib.com>

<http://www.404manual.com>

<http://www.luxmanual.com>

<http://aubethermostatmanual.com>

Golf course search by state

<http://golfingnear.com>

Email search by domain

<http://emailbydomain.com>

Auto manuals search

<http://auto.somanuals.com>

TV manuals search

<http://tv.somanuals.com>