OS/390

Parallel Sysplex Test Report

R9+R10

GC28-1963-18

OS/390

Parallel Sysplex Test Report

R9+R10

GC28-1963-18

Note!
Before using this information and the products it supports, be sure to read the general information under “Appendix E. Notices” on page 233.

Nineteenth Edition, September 2000

This edition applies to Parallel Sysplex environment function that includes System/390 data sharing and parallelism. The Parallel Sysplex uses the MVS/ESA SP Version 5 or the OS/390 operating system.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department B6ZH, Building 710-2, Mail Station P350
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9414
FAX (Other Countries):
Your International Access Code +1+845+432-9414

IBMLink (United States customers only): IBMUSM10(DBLISS)
Internet Email: dbliss@us.ibm.com
Internet Web Site: http://www.s390.ibm.com/os390/support/os390tst/

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

Opening Remarks

A Message from Our Team

As you read this document, keep in mind that ***we need your feedback.*** We want to hear anything you want to tell us, whether it's positive or less than positive. ***We especially want to know what you'd like to see in future editions.*** That helps us prioritize what we do in our next test phase. We will also make additional information available upon request if you see something that sparks your interest. Please use the Readers' Comments form at the back of this document.

We are a team whose combined computing experience is hundreds of years, but we have a great deal to learn from you, our customers. We will try to put your input to the best possible use. Thank you.

Al Alexsa
Loraine Arnold
Sue Barton
Frank Bellacicco
Duane Beyer
Jeff Bixler
Muriel Bixler
Dave Bliss
Dave Buehl
John Corry
Don Costello
Kevin Coyne
Luis Cruz
Tony DiLorenzo
Bob Fantom
Nancy Finn
Bobby Gardinor
Ramses Ibrahim
Joan Kelley
Fred Lates
Frank Lefevre
Kristine Logan
Dan Mackoway
Marguerite Manicone
Sue Marcotte
James Mitchell
Yva Montalvo
Bob Muenkel
Carol Muller
Deron Nix
Jim Rossi
Jim Stutzman
Johnny Tan
Patti Veith
Ami Walker

Contents

Opening Remarks . . . iii

Figures . . . ix

Tables . . . xi

Introduction—What You Can Expect From Us . . . xiii
What Is Our Starting Point? . . . xiii
What Does Our Computing Environment Look Like? . . . xiv
Before You Read On... . . . xiv
Where to Find More Information . . . xvi

Part 1. Parallel Sysplex . . . 1

Chapter 1. Understanding Our Parallel Sysplex Environment . . . 3
Our Sysplex Hardware Configuration . . . 3
Hardware Configuration Details . . . 4
Our Sysplex Software Configuration . . . 11
Our Naming Conventions . . . 13
Our VTAM Configuration . . . 13
Our Workloads . . . 14
OS/390 Base Workloads . . . 15
Application Enablement Workloads . . . 15
Networking Workloads . . . 16
Database Product Workloads . . . 17
Tiny Topics . . . 18

Chapter 2. Migrating To and Using OS/390 . . . 21
Migrating to OS/390 R9 . . . 21
OS/390 R9 Base Migration Experiences . . . 21
Setting Up Multi-System Enclave Support for WLM . . . 23
XES Enhancements . . . 24
Migrating to OS/390 R10 . . . 26
OS/390 R10 Base Migration Experiences . . . 26
Enabling Automatic Tuning of Coupling Facility Structures . . . 28
Enabling SDSF Sysplex Support . . . 32
OS/390 Performance . . . 34

Chapter 3. Setting Up Integrated Cryptographic Service Facility (ICSF) 37
Our ICSF Configuration . . . 37
Experiences Setting Up ICSF . . . 38

Chapter 4. Migrating to DB2 UDB for OS/390 Version 6 . . . 41
Planning for Migration . . . 41
Migration Considerations and Premigration Activities . . . 41
DB2 V6 Migration Experiences . . . 42

Chapter 5. Setting Up a DB2 Large Objects (LOBS) Application With Web Access . . . 45
Our Application . . . 45
Setting Up Our Environment for The Application . . . 46
HFS Setup . . . 47
Updates to the IBM HTTP Server . . . 47

Updates to DB2 47
Updates to Net.Data 51
Sample Net.Data Macros 52
Our Experiences In Setting Up DB2 LOBs Application 55

Chapter 6. Setting Up MQSeries 57
Testing the MQSeries-CICS Bridge 57

Part 2. Networking and Application Enablement 61

Chapter 7. Understanding Our Networking and Application Enablement Environment 63
Our OS/390 Networking and Application Enablement Configuration. 63
Our Ethernet LAN Configuration 64
Our ATM Configuration 65
Our Token Ring LAN Configuration 66
Comparing the Network File Systems 72
OS/390 Networking and Application Enablement Workloads 73

Chapter 8. Using The IBM WebSphere Application Server 75
Migrating to the IBM WebSphere Application Server V1.1 for OS/390 R8 . . . 75
Setting Up the IBM WebSphere Application Server to Run in a Read-Only HFS 75
Migrating to the IBM WebSphere Application Server V1.2 for OS/390 R9 . . . 81
Using the IBM WebSphere Application Server V1.2 for OS/390 R10 83
Migrating to the WebSphere Application Server Version 3.02 Standard Edition for OS/390 83
Migrating Applications for the Version 3.02 Level of the IBM WebSphere Application Server 85

Chapter 9. Using the IBM HTTP Server 89
Using the IBM HTTP Server V5.2 89
Using the HTTP Recovery Directive 89
Using Scalable Server Subsystem Mode 89
Using Net.Data to Connect the IBM HTTP Server to DB2 90
Migrating to Shared HFS for the IBM HTTP Server 91
Migrating to The IBM HTTP Server Version 5.3 97
Migrating to System SSL Support 97
Using LDAP to Protect IBM HTTP Server Resources 98
Customizing the IBM HTTP Server Configuration File for LDAP Protection 99
Customizing the LDAP Server for LDAP Protection 100
Verifying LDAP Protection for The IBM HTTP Server 101

Chapter 10. Using OS/390 Infoprint Server 103
Our Infoprint Server Configuration 103
Experiences With Infoprint Server 106
Migrating to Infoprint Server. 106
Migrating PSF to Use the Printer Inventory 106
Setting Up and Using Infoprint Server Transforms 106

Chapter 11. Managing a Hierarchical File System (HFS) 109
Highlights of Our Strategy 109
Migrating to a Shared Hierarchical File System (HFS) 109
The Shared HFS in Our Environment 110
How We Perpared Our Environment for R9 Shared HFS 119
Customizing our Shared BPXPRMxx Member 121

Customizing BPXPRMxx Members for Individual Systems 121
Creating an OMVS Couple Data Set (CDS) 122
Mounting File Systems at /usr 122
Our HFS Service Environment. 126
High-Level Steps for Servicing the HFS 128
Samples Available on Our Web Site. 130
HFS Naming Conventions 130
Our Post-Install Customization EXEC 131
Initializing File Systems in the System Specific HFSs 132
Using the Automount Facility to Manage User Home Directories 133
Performance Considerations for Shared HFS 133
Performance of Read, Write, and Copy of File Systems 134
Performance of File System Mounts 135
Recommendations for Controlling Performance 135

Chapter 12. Setting Up SAP R/3 137
Our SAP R/3 Configuration 137
Installing the Operating System Collector (SAPOSCOL) 139

Chapter 13. Migrating the OS/390 Security Server LDAP Server to OS/390 R10. 141
Migrating to the New DB2 Backend Database 141

Part 3. Systems Management 145

Chapter 14. Parallel Sysplex Automation 147
Our Automation Setup. 148
An Overview of Our Automation 149
Running a Dual NetView Environment 151
Using Command Lists (CLISTs) 151
Setting Trigger Conditions for a Subsystem 152
Displaying Status of a Trigger Condition 153
Displaying or Overriding a Service Period 154
Exploring the Sysplex Automation Functions 156
Parallel Sysplex Operation Center 156
Coupling Facility Drain 163
Experiences With SA OS/390 Sysplex Automation 165

Chapter 15. Introducing the Tivoli Management Framework for OS/390 167
Tivoli Terminology Transition 167
How OS/390 Fits Into the Tivoli Environment 168
Tivoli Management Framework for OS/390 Server and Gateway 169
Our Hardware Configuration 170
Planning, Prerequisites, and System Configuration 170
Our Experiences With Tivoli Management Framework for OS/390 Server and Gateway 174
Tivoli Management Framework for OS/390 Framework Endpoint 188
Our Hardware Configuration 188
Our Installation Experiences 190
Experiences with TME 10 User Administration for OS/390 195
Experiences with TME 10 Security Management for OS/390. 200
Where to Find More Information 203

Part 4. Appendixes 205

Appendix A. Some of Our Parmlib Members 207
Comments about IFAPRD00 208

Appendix B. Some of Our RMF Reports 211
RMF Monitor I Post Processor Summary Report 211
RMF Monitor III Online Sysplex Summary Report 211
RMF Workload Activity Report in WLM Goal Mode 212

Appendix C. Document Distribution 215

Appendix D. Useful Publications and Web Sites 217
IBM Product Publications. 217
IBM Redbooks 229
IBM Web Sites 230
Other Publications 231

Appendix E. Notices 233
Trademarks. 235

Glossary 237
Sources of Terms and Definitions. 237
Explanation of Cross-References. 237

Index 249

Figures

1. Our Sysplex Hardware Configuration . . . 4
2. Our Coupling Facility Channel Configuration . . . 9
3. Our Sysplex Software Configuration . . . 12
4. Our VTAM Configuration . . . 14
5. Our ICSF Configuration . . . 38
6. Our DB2 LOB Application . . . 46
7. Our MQSeries-CICS Bridge Application . . . 58
8. Our Networking and Application Enablement Configuration . . . 63
9. Our token-ring LAN A . . . 68
10. Our token-ring LAN B . . . 69
11. Our token-ring LAN C . . . 71
12. Our token-ring LAN D . . . 72
13. Moving Customizable Application Server Directories From the Root HFS into Read-Write HFSs 77
14. Our WAS.sh Shell Script . . . 79
15. Testing the version 3.02 level of the IBM WebSphere Application Server . . . 0
16. Putting the version 3.02 level of the IBM Application Server into Production: . . . 0
17. Enabling Shared HFS for the IBM HTTP Server . . . 92
18. Reclaiming Space from a Too-Large HFS: Steps 1 and 2 . . . 95
19. Reclaiming Space from a Too-Large HFS: Steps 3, 4, and 5 . . . 96
20. Our Print Server Configuration . . . 104
21. Adding the XFORMPRT Printer Definition . . . 107
22. XFORMPRT Allocation Values . . . 107
23. XFORMPRT Allocation Values . . . 108
24. Logical and Physical View of Shared HFS . . . 111
25. Our Sysplex Root . . . 112
26. Our Version HFS . . . 113
27. A System Specific HFS . . . 115
28. Scenario One: How a Request for /etc Resolves . . . 116
29. Scenario Two: Accessing Another System's /etc Directory . . . 117
30. Scenario Three: Accessing the Shared /bin Directory . . . 118
31. Creating a Directory in the Version HFS: This Will Work Until You IPL the System With a New Version HFS . . . 123
32. Creating a Directory in the Version HFS: This Stops Working When You IPL the System With a Different Version HFS . . . 124
33. How To Mount Serviceable File Systems at /usr . . . 125
34. Mounting a File System at /usr/spool . . . 126
35. Applying Service in Our HFS Environment: . . . 127
36. Association of a Root HFS Data Set with a SYSRES Volume . . . 128
37. Our SAP Configuration . . . 138
38. Our Automation Setup . . . 150
39. Dual NetView Environment . . . 151
40. SA OS/390 DISPSTAT Panel . . . 153
41. SA OS/390 Trigger Condition Selection List . . . 153
42. STARTUP Trigger Condition Status . . . 154
43. SHUTDOWN Trigger Condition Status . . . 154
44. SA OS/390 Service Periods Selection Panel . . . 155
45. SA OS/390 Display or Override Current Day's Service Period Schedules . . . 155
46. SA OS/390 Display or Override Seven Day Service Period Schedules . . . 156
47. SA OS/390 DISPPLEX Main Panel . . . 157
48. Example of the SA OS/390 DISPXSYS Display . . . 158
49. Example Display of the SA OS/390 DISPXSYS T Subcommand . . . 159
50. Example of the SA OS/390 DISPCF Display . . . 160
51. Example of the SA OS/390 DISPSTR Display . . . 161

52. Example of the SA OS/390 DISPCDS Display. 162
53. Example of the SA OS/390 DISPCONS Display 163
54. Example of the SA OS/390 CFDRAIN Initial Status Display 164
55. Example of the SA OS/390 CFDRAIN Rebuild Confirmation Panel 164
56. Example of the SA OS/390 CFDRAIN Progress Display 165
57. Our Tivoli Management Framework for OS/390 Server and Gateway Configuration 170
58. Our Tivoli Management Framework for OS/390 Framework Endpoint Configuration 189
59. Our Tivoli User ID Profiles 196
60. Release Notes Page 7—Removing the MAXSIZE Check from the Validation Policy for racf_tso_tlsize. 198
61. Release Notes Page 8—Removing the MAXSIZE Check from the Validation Policy for racf_tso_tmsize.. 199
62. Example RMF Monitor I Post Processor Summary Report 211
63. Example RMF Monitor III Online Sysplex Summary Report 212
64. Example RMF Workload Activity Report in WLM Goal Mode 213

Tables

1. Parallel Sysplex Planning Library Publications xvi
2. S/390 Processors and Servers (Mainframes) 5
3. Coupling Facilities 6
4. Other Sysplex Hardware Configuration Details 9
5. Our Workloads 15
6. Highlights of Some OS/390 Elements, Features, and Related Products and Functions 19
7. Additional Compatibility APARs for Migrating to OS/390 R9. 22
8. Our LOB Buffer Pool Values. 48
9. DB2 Objects for our LOBs Application 48
10. System Specific HFSs Mounted Read-Write 131
11. Performance Comparison of Reads Between File Systems Mounted Locally and Remotely. 134
12. Performance Comparison of Writes Between File Systems Mounted Locally and Remotely. 134
13. Performance Comparison of Copy Between File Systems Mounted Locally and Remotely 135
14. CLISTs We Use With NetView and SA OS/390 152
15. Tivoli Terminology Transition Summary 168
16. RDBMS Interface Module (RIM) configuration values for Tivoli Inventory 186
17. Access Permissions for the /etc/Tivoli/lcf/dat Directory Structure 191
18. Highlights of Parmlib Changes for OS/390 R9 and R10 207
19. ATM Publications 217
20. AIX Publications 217
21. BookManager READ/MVS Publications 217
22. CICS Version 4 Publications 217
23. CICS TS Publications 218
24. CICSPlex SM Publications 218
25. CICSVR Publications 218
26. COBOL Publications 218
27. DB2 Publications 219
28. DCE Publications 219
29. DFSMS/MVS Publications 219
30. DRDA Publications. 219
31. eNetwork Communications Server Publications 220
32. Firewall Technologies Publications 220
33. Hardware Publications 220
34. HCD Publications 221
35. High Level Assembler Publications 221
36. IBM HTTP Server Publications 221
37. IBM WebSphere Application Server for OS/390 Publications 221
38. IMS Publications 221
39. Intelligent Miner Publications 222
40. ISPF Publications 222
41. JES2 Publications 222
42. JES3 Publications 222
43. LAN CID Publications 223
44. Language Environment Publications 223
45. LANRES/MVS Publications 223
46. LAN Server Publications 223
47. LDAP Server Publications 223
48. MQSeries Publications 223
49. MVS Publications 224
50. NetQuestion Publications 225
51. NetView V3 Publications 225
52. Network File System Publication. 225
53. Net.Data Publications. 225

54. OSA Publications 225
55. OS/390 Publications 225
56. OS/390 IBM Communications Servers Publications 226
57. OS/390 UNIX System Services Publications 226
58. Parallel Sysplex Publications 226
59. PSF Publications 226
60. Print Server Publications 227
61. RMF Publications 227
62. SAP Publications 227
63. SA OS/390 Publications 227
64. RACF Publications 227
65. Softcopy Print Publications 228
66. SOMobjects Publications 228
67. Tivoli Publications 228
68. TSO/E Publications 229
69. VisualLift Publications 229
70. IBM Systems Center Publications (Redbooks) 230
71. Some Useful IBM Web Sites 230

Introduction—What You Can Expect From Us

This document is a test report, written from the perspective of a system programmer. A team of IBM S/390 testers and system programmers simulating a customer production environment want to continuously communicate directly with you, the S/390 customer system programmers.

We have been producing this test report quarterly since March, 1995. Parallel Sysplex was the sole focus of the four editions we produced in 1995. In 1996, we continued to run a computing environment whose core was a Parallel Sysplex, but we expanded our testing to encompass OS/390, many of whose elements are not necessarily sysplex-oriented. Our 1996 editions focused on our experiences migrating to OS/390 V1R1 and V1R2; our 1997 editions focused on our experiences migrating to OS/390 V1R3 and V2R4; our 1998 editions focused on OS/390 V2R5 and V2R6; and our 1999 editions focused on OS/390 V2R7 and V2R8. In 2000, we continue with OS/390 V2R9 and V2R10. Most often in our text, we refer to the release number only (for instance, OS/390 R9).

Continuing Our Tradition: Even though we have expanded our focus, our objectives in writing this test report remain as they were:

- Describe the cross product and integrated testing that we have done to verify that certain functions in specific versions and releases of S/390 products work together.
- Share our experiences. In a nutshell, if any of our experiences turned out to be painful, we tell you how to avoid that pain.
- Provide you with specific recommendations that are tested and verified.

We continue to acknowledge the challenge of running multiple hardware and software products and making them work together. We're taking more of that challenge upon ourselves, to shield you from as much complexity as possible.

What Is Our Starting Point?

Our testing began on an existing sysplex; with the introduction of OS/390 we added a number of workstations and local area networks (LANs) and built a highly interconnected, client/server, multi-platform, varied network environment.

At a high level, our process is much the same as it has been: to run a production shop in the same manner that customers do. We believe that only by being customers ourselves can we understand what customers actually experience when they use our products.

Our process includes simulating the workload stress levels, transaction rates, and lock contention rates that customers might experience. We stress many of the same areas of the system that customers stress.

IBM's testing of its products is and has always been extensive. ***The test process described in this document is not a replacement for other test efforts.*** Rather, it is an additional test effort with a shift in emphasis, focusing more on the customer experience, cross-product dependencies, and sensitivity to end users. When we encounter a problem, our goal is to keep systems up and running so that end users can still process work.

The results of our testing should be:

- A more stable system for you, our customers, at known, tested, and recreatable service levels
- A reduction in the time and cost of your migration to new product releases and functions.

Keep reading to find out what our environment looks like, why it looks that way, and what aspects of our setup might be advantageous in your installation.

What Does Our Computing Environment Look Like?

The Parallel Sysplex that forms the core of our test environment has grown and changed over the years. We started out with a 10-system sysplex. Later on, we expanded to 16 systems by adding 3 more production systems, 2 test systems, and a non-production system to run TPNS (which simulates users generating transactions). So, our resulting configuration had 12 production systems, 3 test systems, and 1 system for TPNS. (You can read about why we run TPNS on a non-production system in the ICF chapter of our December 1998 edition.)

Most recently, as part of our migration to OS/390 R10 and in preparation for some other planned hardware changes, we've eliminated two production systems and one test system. We currently run a 13-system sysplex that consists of 10 production systems, 2 test systems, and 1 system for TPNS. Because this edition of our test report covers both OS/390 R9 and R10, the figures and descriptions in "Our Sysplex Hardware Configuration" on page 3 still include information about our 16-way configuration.

We also have four token-ring LANs and an Ethernet LAN set up to exercise the networking and application enablement elements and features of OS/390.

The following figures illustrate each of the major pieces of our computing environment in greater detail:

- Figure 1 on page 4 illustrates our sysplex hardware configuration.
- Figure 3 on page 12 illustrates our sysplex software configuration.
- Figure 8 on page 63 illustrates our networking and application enablement configuration. We describe our networking and application enablement environment in greater detail in the sections that follow Figure 8.

Before You Read On...

Here is some important information to note before you continue reading:

Summary of Changes for Nineteenth Edition—September, 2000: This is the nineteenth edition of a document we update quarterly based on new information and experiences. If the edition you are reading is more than a few months old, consult with your IBM representative or check the IBM World Wide Web pages to see if a newer edition is available (see "Document Distribution" on page xv).

This edition contains new or changed information in the following areas:

- Our sysplex hardware configuration
- Our experiences migrating to OS/390 R10, including automatic tuning of coupling facility structures and SDSF sysplex support
- Our experiences setting up MQSeries
- Our experiences with the IBM HTTP Server, including migrating to the IBM HTTP Server version 5.3 and using LDAP to protect IBM HTTP Server resources.

- Our experiences with the IBM WebSphere Application Server, including using to the IBM WebSphere Application Server V1.2 and migrating to the IBM WebSphere Application Server 3.02.
- Our experiences setting up SAP R/3
- Our experiences migrating the OS/390 Security Server LDAP Server to OS/390 R10
- Our information on the Tivoli Management Framework for OS/390.

Note: See "Terminology" on page xv for the terminology change from OpenEdition to OS/390 UNIX System Services.

Editorial and technical updates appear throughout the text. All technical updates are marked with a change bar (|) in the left margin.

Document Distribution: Because we publish quarterly, printing hardcopy books for timely distribution of every edition is impractical. To provide you with the latest information, this report is currently available on the IBM intranet, the Internet, and through the IBMLink InfoCenter host environment.

Our Own Web Site

We've built a Web site for our team, OS/390 Integration Test, where you can access all editions and formats of our test report (including our companion publication, *OS/390 Parallel Sysplex Recovery*):

`http://www.ibm.com/s390/os390/support/os390tst/`

If you can't get to our Web site for some reason, please see "Appendix C. Document Distribution" on page 215 for alternate ways to access our reports.

Disposition of Our Prior Editions: Each quarterly edition of our test report is cumulative throughout the calendar year. At the end of each year, we freeze the content in our December edition, and we begin with a new test report in March of the following year. Printed copies of each December edition, with all of that year's information, become available through the publications ordering system sometime during the first quarter of the following year. The edition you are reading right now, which is June 2000 (GC28-1963-17), documents our sysplex test efforts during 2000 and our experiences with OS/390 R9. The information in this edition will become available in printed copy early in the first quarter of the year 2001. See "Where to Find More Information" on page xvi for the order numbers and brief descriptions of each of our other year-end editions.

Terminology:

- When you see the term ***sysplex***, understand it to mean a sysplex with a coupling facility, which is a ***Parallel Sysplex.***
- As part of the name change of OpenEdition to OS/390 UNIX System Services, occurrences of OS/390 OpenEdition have been changed to OS/390 UNIX System Services or its abbreviated name, OS/390 UNIX. OpenEdition may continue to appear in messages, panel text, and other code with OS/390 UNIX System Services.

Acronyms: All acronyms are listed in the glossary. In the interest of brevity, most product and other acronyms are not spelled out in the text.

Where to Find More Information

If you are not familiar with Parallel Sysplex terminology and concepts, you should consult the following publications first:

Table 1. Parallel Sysplex Planning Library Publications

Publication Title	Order Number
OS/390 Parallel Sysplex Overview	GC28-1860
OS/390 Parallel Sysplex Systems Management	GC28–1861
Note: This book is being removed from the library as of OS/390 V2R10. Refer to the online Parallel Sysplex Configuration Assistant at `http://www.ibm.com/s390/pso/psotool`	
OS/390 Parallel Sysplex Hardware and Software Migration	GC28–1862
Note: This book is being removed from the library as of OS/390 V2R10. Refer to the online Parallel Sysplex Configuration Assistant at `http://www.ibm.com/s390/pso/psotool`	
OS/390 Parallel Sysplex Application Migration	GC28-1863

Here are some additional key titles:

- See *OS/390 MVS Setting Up a Sysplex*, GC28-1779, for information about installing and managing OS/390 MVS systems in a sysplex.
- See *System/390 MVS Parallel Sysplex Test Report*, GC28-1236, which documents our sysplex test efforts during 1995, and forms the foundation for the OS/390 test reports. Printed copies of this edition are available for order.
- See *OS/390 Parallel Sysplex Test Report*, GC28-1963-03, which documents our sysplex test efforts during 1996, as well as our experiences with OS/390 R1 and R2. Printed copies of this edition (specifically, GC28-1963 at the -03 level) are available for order.
- See *OS/390 Parallel Sysplex Test Report*, GC28-1963-07, which documents our sysplex test efforts during 1997, as well as our experiences with OS/390 R3 and R4. Printed copies of this edition (specifically, GC28-1963 at the -07 level) are available for order.
- See *OS/390 Parallel Sysplex Test Report*, GC28-1963-11, which documents our sysplex test efforts during 1998, as well as our experiences with OS/390 R5 and R6. Printed copies of this edition (specifically, GC28-1963 at the -11 level) are available for order.
- See *OS/390 Parallel Sysplex Test Report*, GC28-1963-15, which documents our sysplex test efforts during 1999, as well as our experiences with OS/390 R7 and R8. Printed copies of this edition (specifically, GC28-1963 at the -15 level) are available for order.
- See *OS/390 Parallel Sysplex Recovery*, GA22-7286, for information about Parallel Sysplex recovery scenarios we've executed in our test environment, including system, subsystem, and coupling facility recovery. We describe how to be prepared for potential problems in a Parallel Sysplex, what the indicators are to let you know there's a problem, and what actions to take to recover. Printed copies of this publication are available for order.
- See *OS/390 Planning for Installation*, GC28-1726, for OS/390 installation information.

For information about specific products discussed in this publication, see the individual product libraries.

“Appendix D. Useful Publications and Web Sites” on page 217 lists many IBM product publications, Redbooks, Web sites, and some non-IBM publications that we use or have used.

Part 1. Parallel Sysplex

Chapter 1. Understanding Our Parallel Sysplex Environment 3
Our Sysplex Hardware Configuration 3
Hardware Configuration Details 4
S/390 Servers. 5
Coupling Facilities 6
Other Sysplex Hardware 9
Our Sysplex Software Configuration 11
Our Naming Conventions 13
Our VTAM Configuration 13
Our Workloads 14
OS/390 Base Workloads 15
Application Enablement Workloads 15
Networking Workloads 16
Database Product Workloads 17
Database Product OLTP Workloads 17
Database Product Batch Workloads 18
Tiny Topics 18

Chapter 2. Migrating To and Using OS/390 21
Migrating to OS/390 R9. 21
OS/390 R9 Base Migration Experiences 21
Setting Up Multi-System Enclave Support for WLM 23
XES Enhancements 24
Structure Full Monitoring 24
XES Event Supression 25
Minimum Required CFLEVEL 25
CFRM Policy Now Allows Up To 512 Structures 25
Recovery Enhancements 25
Rebuild Quiesce Enhancements 26
Migrating to OS/390 R10 26
OS/390 R10 Base Migration Experiences 26
Enabling Automatic Tuning of Coupling Facility Structures 28
How XCF Automatic Alter Support Works 28
Experiences With Automatic Alter 29
Enabling SDSF Sysplex Support 32
OS/390 Performance. 34

Chapter 3. Setting Up Integrated Cryptographic Service Facility (ICSF) 37
Our ICSF Configuration. 37
Experiences Setting Up ICSF 38

Chapter 4. Migrating to DB2 UDB for OS/390 Version 6 41
Planning for Migration 41
Migration Considerations and Premigration Activities 41
DB2 V6 Migration Experiences 42

Chapter 5. Setting Up a DB2 Large Objects (LOBS) Application With Web Access 45
Our Application 45
Setting Up Our Environment for The Application. 46
HFS Setup 47
Updates to the IBM HTTP Server 47
Updates to DB2 47

Creating New Buffer Pools and Group Buffer Pools 47
Setting Up DB2 Objects 48
Putting Our LOB Data Into DB2 50
Service for DB2 to Enable LOBs 50
Updates to Net.Data 51
Sample Net.Data Macros 52
Our Experiences In Setting Up DB2 LOBs Application 55

Chapter 6. Setting Up MQSeries 57
Testing the MQSeries-CICS Bridge 57

The above chapters describe the Parallel Sysplex aspects of our computing environment.

Chapter 1. Understanding Our Parallel Sysplex Environment

In this chapter we describe our Parallel Sysplex computing environment, including information about our hardware and software configurations, and descriptions of the workloads we run.

Our Sysplex Hardware Configuration

Figure 1 on page 4 is a high-level view of our sysplex hardware configuration. Broad arrows in the figure do not indicate physical point-to-point connections but indicate general connectivity between processors, the coupling facility, and other devices.

"Hardware Configuration Details" on page 4 provides additional information about our sysplex hardware configuration.

A Note About the Figures and Tables In This Chapter: The figures and tables in this chapter include an ES/9000 Model 951 (a 9021 711-based processor) and an S/390 Parallel Transaction Server 9672-E01. These servers were part of our configuration through our testing of OS/390 V2R9 and into our migration to OS/390 V2R10. As we migrated to R10, however, we removed these servers from our configuration along with the OS/390 systems that ran on them (JD0, JI0, and Z1). We did this because these older servers do not support OS/390 R10.

OS/390 R10 exploits ESA/390 architectural enhancements that were implemented on selected IBM S/390 servers. OS/390 R10 runs only on servers that implement the architectural enhancements. See the target system hardware requirements in *OS/390 Planning for Installation* for a list of supported servers for OS/390 R10.

As of our migration to OS/390 R10, we are running a 13-member Parallel Sysplex (down from 16 members previously). Shortly, we'll be adding a new server to our configuration, but we'll continue to run our sysplex with 13 members. We'll elaborate on the details as they unfold in coming editions of our test report.

Parallel Sysplex Environment

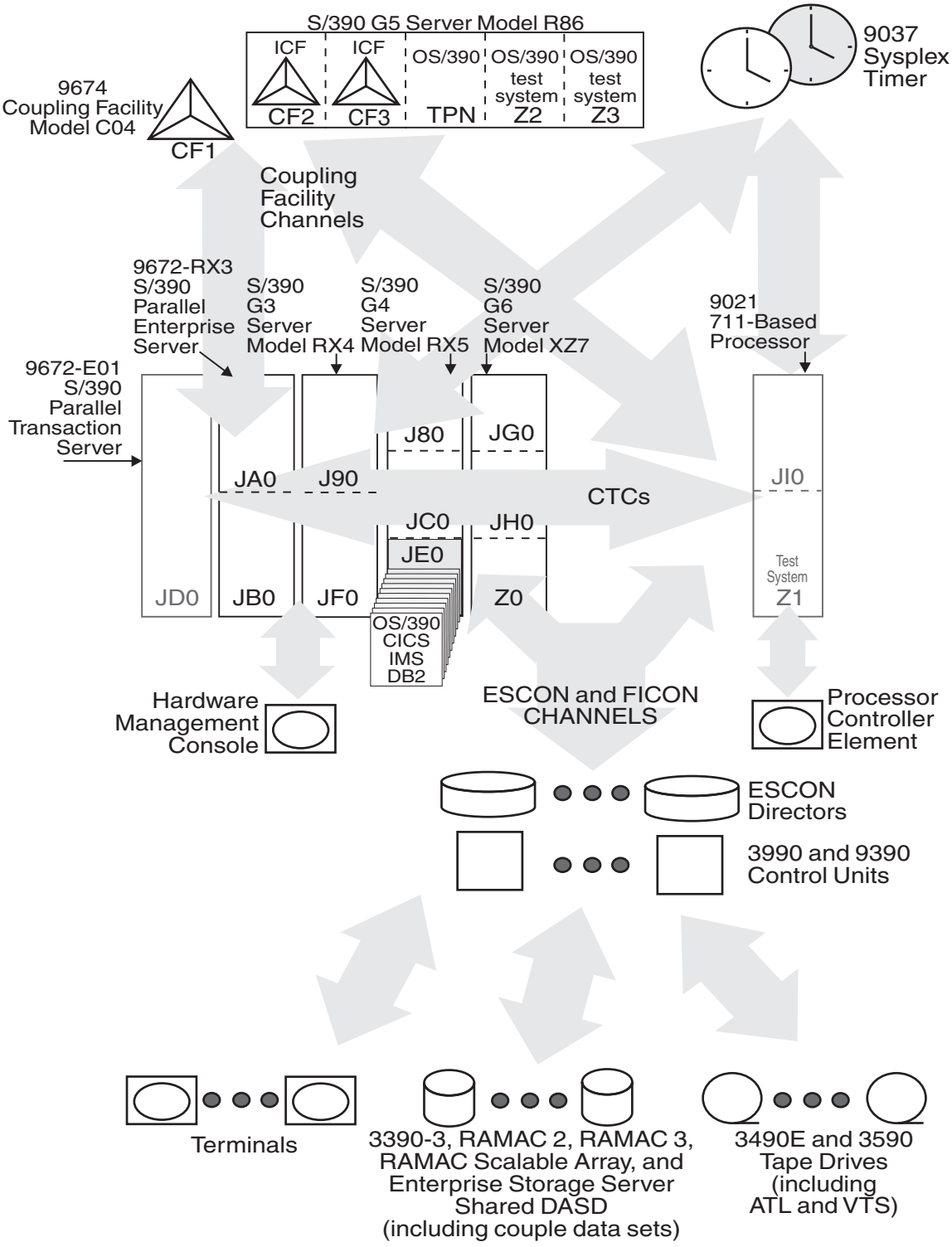


Figure 1. Our Sysplex Hardware Configuration

Hardware Configuration Details

The figures and tables in this section provide additional details about the overall hardware configuration shown in Figure 1.

S/390 Servers

Table 2 provides information about the S/390 servers in our sysplex:

Table 2. S/390 Processors and Servers (Mainframes)

Model	CPCs and CPs	Mode	Con-trolled By	HSA	Storage: Central Expanded	Systems
ES/9000 Model 951 (a 9021 711-based processor; see note 1 on page 6)	1 CPC with 5 CPs	LPAR mode with 2 LPs	PCE	48M	464M 1024M	JI0 (production LP, 5 shared CPs, weight of 100, uncapped)
					512M 1024M	Z1 (test LP, 3 shared CPs, weight of 20, capped)
S/390 9672-E01 (a 9672 Parallel Transaction Server; see note 1 on page 6)	1 CPC with 6 CPs	BASIC mode	HMC	13M	768M 256M	JD0 (production LP with 6 dedicated CPs)
S/390 9672-RX3 (a 9672 Parallel Enterprise Server)	1 CPC with 10 CPs	LPAR mode with 2 LPs	HMC	25M	1000M 1024M	JA0 (production LP with 5 dedicated CPs)
					1018M 256M	JB0 (production LP with 5 dedicated CPs)
S/390 Parallel Enterprise Server, Generation 3 (G3), Model RX4	1 CPC with 10 CPs	LPAR mode with 2 LPs	HMC	48M	2048M 128M	J90 (production LP with 6 shared CPs)
					2048M 128M	JF0 (production LP with 6 shared CPs)
S/390 Parallel Enterprise Server, Generation 4 (G4), Model RX5	1 CPC with 10 CPs	LPAR mode with 3 LPs	HMC	64M	1984M 128M	J80 (production LP with 9 shared CPs)
					1856M 128M	JC0 (production LP with 4 shared CPs)
					1856M 128M	JE0 (production LP with 4 shared CPs)
S/390 Parallel Enterprise Server, Generation 5 (G5), Model R86 (see note 4 below).	1 CPC with 8 CPs (and 2 ICF CPs, see Table 3 below.)	LPAR mode with 5 LPs (2 being used as ICFs)	HMC	96M	2048M 512M	TPN (LP to run OS/390 and TPNS; 8 shared CPs with a weight of 70)
					1024M 256M	Z2 (test LP with 2 shared CPs, weight of 10)
					1024M 256M	Z3 (test LP with 2 shared CPs, weight of 10)
S/390 Parallel Enterprise Server, Generation 6 (G6), Model XZ7 (see note 5 below).	1 CPC with 12 CPs	LPAR mode with 3 LPs	HMC	64M	2048M 256M	JG0 (production LP with 8 shared CPs, weight of 10)
					2048M 256M	JH0 (production LP with 6 shared CPs, weight of 10)
					2048M 1440M	Z0 (production LP with 10 shared CPs, weight of 10)

Table 2. S/390 Processors and Servers (Mainframes) (continued)

Model	CPCs and CPs	Mode	Con-trolled By	HSA	Storage: Central	Expanded	Systems

Notes:

1. Upon migrating to OS/390 R10, we removed the ES/9000 and E01 models from our configuration, along with systems JD0, JI0, and Z1.
2. For the E01, RX3, G3, G4, G5, and G6 models, evaluate whether you need to configure some of your storage as expanded storage based on the functions you are using. Some functions, such as Hiperbatch, require expanded storage.
3. Our complete processor configuration, excluding coupling facilities and system TPN (which simulates users generating transactions), equates to 345 MSUs.
4. We originally had one test system in our configuration, Z1. We've added two more test systems, Z2 and Z3. With three systems we have a more robust test environment, as we have set up test data sharing groups for IMS, DB2, and CICS. Our second ICF (CF3) connects only to our three test systems.
5. For our G6, we applied the IZP version of IOCP 1.7.2, which is available with the fix for APAR OW36887 (PTF UW90547).

Coupling Facilities

Table 3 provides information about the coupling facilities in our sysplex:

Table 3. Coupling Facilities

Coupling Facility Name	Model CPCs and CPs CFLEVEL Controlled By	Central Storage and Usage	Channel Distribution
CF1	S/390 9674-C04 Coupling Facility with expansion cage, feature code 2000 1 CPC with 6 CPs CFLEVEL=8 (CFCC Release 08.00, Service Level 01.01) Controlled by the HMC	2G main plus 2G expanded, for a total of 4G available for structures (We use about 1.8G for active structures, leaving about 2.2G for other coupling facility structures and future growth.)	19 TYPE=CFR channels. There are 19 corresponding TYPE=CFS channels on the following systems; "shared" indicates that the systems share that number of channels using EMIF: • J90/JF0: 4 shared (HiPerLinks) • JA0/JB0: 2 shared • J80/JC0/JE0: 4 shared (HiPerLinks) • JD0: 2 dedicated • JG0/JH0/Z0: 4 shared (HiPerLinks) • JI0/Z1: 1 shared • TPN/Z2/Z3: 2 shared (HiPerLinks) See Figure 2 on page 9 for an illustration of the coupling facility channel configuration between our OS/390 systems and our 9674-C02 Coupling Facility.

Table 3. Coupling Facilities (continued)

Coupling Facility Name	Model CPCs and CPs CFLEVEL Controlled By	Central Storage and Usage	Channel Distribution
CF2	ICF Feature on an S/390 Parallel Enterprise Server, Generation 5, Model R86 (see the December 1998 edition of our test report for more information). 2 shared ICF CPs with a weight of 100 and dynamic dispatch turned off (DYNDISP=OFF). CFLEVEL=9 (CFCC Release 09.00, Service Level 01.00) Controlled by the HMC.	2G main plus 608M expanded, for a total of 2.6G available for structures (We use about 1.2G for active structures, leaving about 1.4G for other coupling facility structures and future growth.)	13 TYPE=CFR channels. (See note 1 below.) There are 13 corresponding TYPE=CFS channels on the following systems; “shared” indicates that the systems share that number of channels using EMIF: • J90/JF0: 3 shared (HiPerLinks) • JA0/JB0: 2 shared • J80/JC0/JE0: 3 shared (HiPerLinks) • JD0: 2 dedicated • JI0/Z1: 2 shared • TPN/Z2/Z3: 1 shared (HiPerLink) 6 TYPE=CBR channels. There are 6 corresponding TYPE=CBS channels on the following systems; “shared” indicates that the systems share that number of channels using EMIF. These are integrated cluster bus (ICB) channels. (See note 2 below.) • TPN: 2 dedicated • JG0/JH0/Z0: 4 shared 1 TYPE=ICR channel. There is 1 corresponding dedicated TYPE=ICS channel on system TPN. These are internal coupling (IC) channels. (See note 3 below.)
CF3	ICF Feature on an S/390 Parallel Enterprise Server, Generation 5, Model R86. We recently added this second ICF LP on the same G5 as CF2 above. 1 shared ICF CP with a weight of 100 and dynamic dispatch turned on (DYNDISP=ON). CFLEVEL=9 (CFCC Release 09.00, Service Level 01.00) Controlled by the HMC.	256M main plus 64M expanded, for a total of 320M available for structures (used only for structures for our three test systems: Z1, Z2, and Z3)	1 TYPE=CFR channel. There is 1 corresponding dedicated TYPE=CFS channel on system Z1. 2 TYPE=ICR channels. There are 2 corresponding TYPE=ICS channels on the following systems: • Z2/Z3: 2 shared

Table 3. Coupling Facilities (continued)

Coupling Facility Name	Model CPCs and CPs CFLEVEL Controlled By	Central Storage and Usage	Channel Distribution

Notes:

1. The total number of coupling facility channels on our R86 is 17: 14 are TYPE=CFR and 3 are TYPE=CFS (as illustrated in Figure 2 on page 9). Two of the CFS links connect systems TPN, Z2, and Z3 to two CFR links on CF1 (our standalone 9674 coupling facility). The third CFS link on our R86 connects systems TPN, Z2, and Z3 to a CFR link on CF2 (one of the two ICFs on our R86). See Figure 2 on page 9 for an illustration of the connections to the remaining CFRs on our R86. (Systems Z2 and Z3 also use ICs, and system TPN also uses both ICBs and ICs, as explained in notes 2 and 3 below.)
2. The R86 and XZ7 also have channels called integrated cluster bus (ICB) channels, denoted as TYPE=CBS (cluster bus sender) and TYPE=CBR (cluster bus receiver). On the R86, two CBSs connect system TPN to two CBRs on CF2. Four CBSs connect systems JG0, JH0, and Z0 on the XZ7 to four CBRs on CF2. See Figure 2 on page 9 for an illustration. As explained in *System Overview—S/390 G5 Enterprise Servers*, ICBs reduce the cost of coupling by using a higher performing but less complex transport link. Reduced hardware and microcode path lengths improve coupling performance. To put it simply, these channels are much faster than standard coupling facility channels or even HiPerLinks. ICBs are available only on G5 and higher models. We recently replaced one of the G5s in our sysplex with a G6. Our G6 houses systems JG0, JH0, and Z0. To support the G6, you need:
   - The IZP version of IOCP 1.7.2, available with the fix for APAR OW36887 (PTF UW90547).
   - The fix for HCD APAR OW37775 (PTFs UW90602, UW90603, and UW90604).
3. The R86 also uses internal coupling (IC) channels, denoted as TYPE=ICS (internal coupling sender) and TYPE=ICR (internal coupling receiver). ICs are logical connections between a coupling facility partition and an MVS partition on the same CPC. One ICS connects system TPN to an ICR on CF2. The other two ICSs connect systems Z2 and Z3 to two ICRs on CF3. ICs require no channel or cable hardware (although CHPID numbers must still be defined in the IOCDS), thereby reducing the cost of internal coupling. Because they utilize the system bus, ICs offer improved coupling performance over standard coupling facility channels (including HiPerLinks) and ICBs. See *System Overview—S/390 G5 Enterprise Servers* for additional information and characteristics of IC channels.

Figure 2 on page 9 illustrates our coupling facility channel configuration:

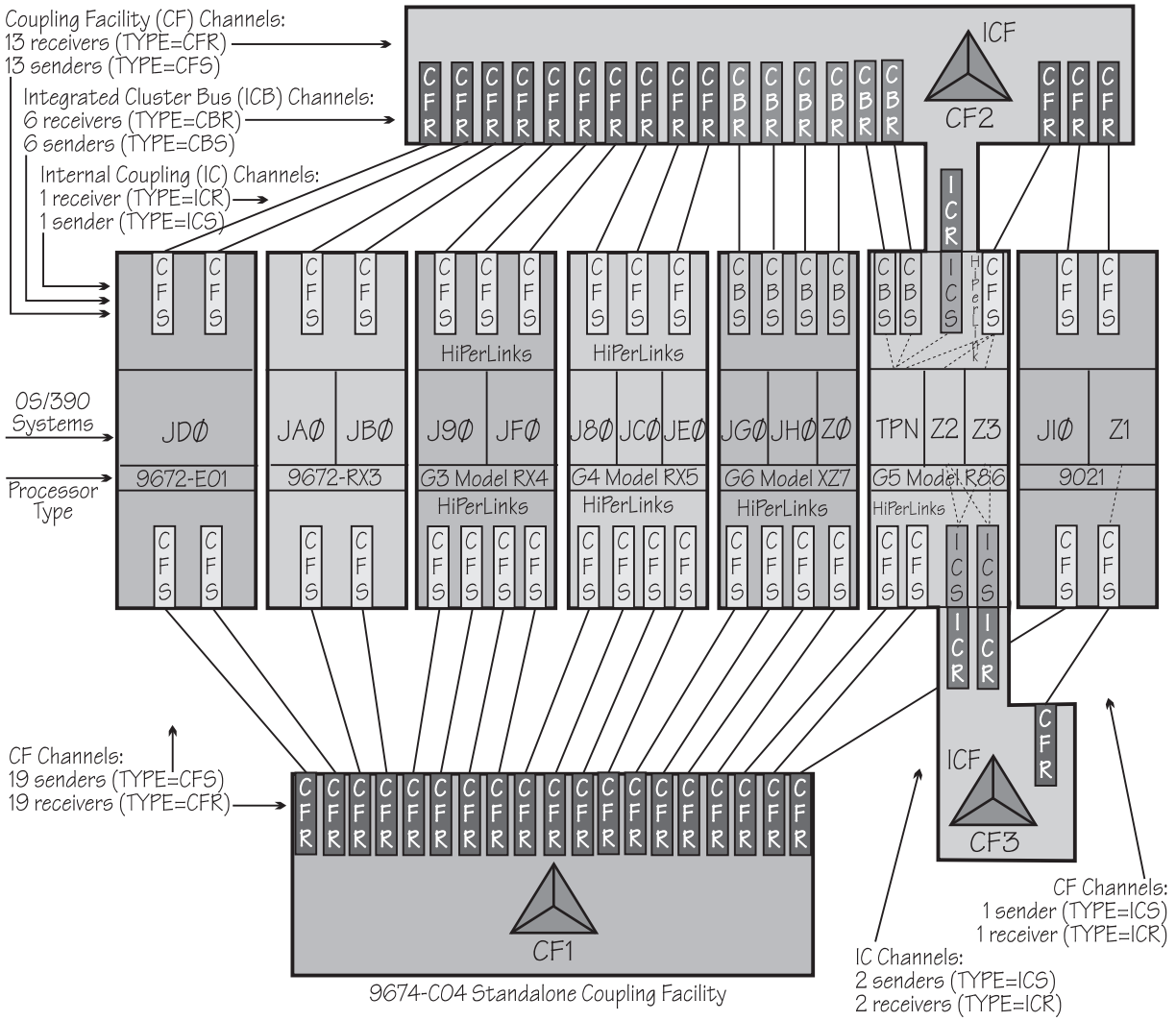


Figure 2. Our Coupling Facility Channel Configuration.

Other Sysplex Hardware

Table 4 highlights the other hardware components in our sysplex:

Table 4. Other Sysplex Hardware Configuration Details

Hardware Element	Model or Type	Additional Information
External Time Reference	Sysplex Timer (9037-001 with feature code 4048)	We use the Sysplex Timer with the Expanded Availability feature, which provides 2 9037 control units that you connect with fiber optic links. We don't have any Sysplex Timer logical offsets defined for any of the LPs in our sysplex.

Table 4. Other Sysplex Hardware Configuration Details (continued)

Hardware Element	Model or Type	Additional Information
Connectivity	CTC communications connections	Connecting every system to every other system. **Note:** All systems use both CTCs and coupling facility structures to communicate. This is strictly optional. You might choose to run with structures only, for ease of systems management. We use both means of communication because it allows us to test more code paths, and under some circumstances, XCF signalling using CTCs is faster than using structures. See *S/390 Parallel Sysplex Performance* for a comparison.
	Coupling facility, ICB, and IC channels	At least 2 connections between every system and each coupling facility, with a few exceptions: • there is 1 connection between systems Z2/Z3 and CF2. • there is 1 connection between systems JI0/Z1 and CF1. • there is 1 connection between system Z1 and CF3. • only our test systems Z1, Z2, and Z3 connect to CF3. **Notes:** 1. For the 9021, the 9672-RX3, the RX4, the RX5, the XZ7, and the R86, we use EMIF to logically share channels between LPs. 2. The coupling facility channels on the RX4, the RX5, the XZ7, and the R86 are HiPerLinks. 3. Our XZ7 has both coupling facility channels and ICB channels. 4. Our R86 has coupling facility, ICB, and IC channels.
	ESCON products	We use ESCON channels and ESCON Directors for our connectivity. Our connections are “any-to-any”, which means every system can get to every device, including tape. We do not use any parallel channels.
	FICON channels	We currently have one FICON channel between our XZ7 (shared by systems JG0, JH0, and Z0) and one of our ESCON Directors using the FICON Bridge feature. The FICON channel from the XZ7 connects to a FICON Bridge card in the ESCON Director. This enables the FICON channel to access ESCON I/O storage control units.
Control units	3990 Storage Control Model 6 9390 (RAMAC 3) Storage Control	At least 2 paths to all DASD.
DASD	3390 Model 3 RAMAC 2 (9392-2) RAMAC 3 (9392-3) RAMAC Scalable Array (9396-200) Enterprise Storage Server (2105)	All volumes shared by all systems; about 70% of our data is SMS-managed. We recently added an IBM Enterprise Storage Server. Like the RAMAC Scalable Array, the ESS contains the control unit and storage media all in one box.

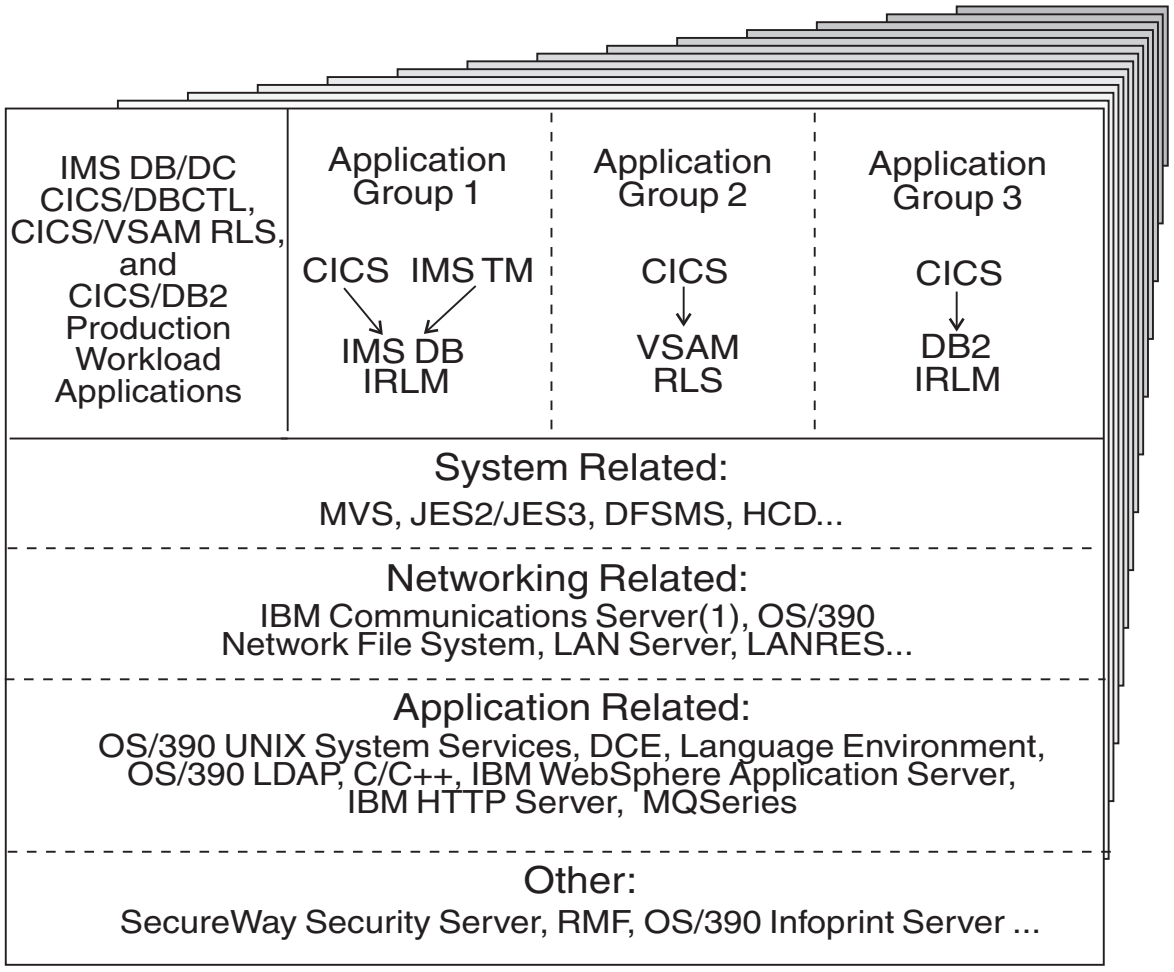
Table 4. Other Sysplex Hardware Configuration Details (continued)

Hardware Element	Model or Type	Additional Information
Tape	3490E tape drives	16 3490E tape drives that can be connected to any system.
Automated tape library (ATL)	3495 Model L40 with 16 additional 3490E tape drives and 16 3590 tape drives.	These tape drives are accessible from all systems.
Virtual Tape Server (VTS)	3494 Model L10	32 virtual 3490 tape drives accessible from all systems.
Printers	3800-3, 3835, and 3820	Connected only to systems JD0, JA0, and Z0.

Our Sysplex Software Configuration

Figure 3 on page 12 shows a high-level view of our sysplex software configuration. We run the OS/390 V2R9 operating system along with CICS Transaction Server (CICS TS) V1R2, IMS/ESA V6 with its associated IRLM, and DB2 UDB for OS/390 V6 with its associated IRLM. Note that we currently run only IBM software in our sysplex.

As you will see when you read *OS/390 Planning for Installation*, OS/390 is made up of base elements and optional features. Certain elements and features of OS/390 support something called *dynamic enablement*. When placing your order, if you indicate you want to use one or more of these, IBM ships you a tailored IFAPRDxx parmlib member with those elements or features enabled. See *OS/390 Planning for Installation* and *OS/390 MVS Product Management* for more information about dynamic enablement.



Production systems:
Z0, J80, J90, JA0, JB0, JC0, JE0, JF0, JG0 (IMS, VSAM RLS, and DB2 data sharing)
JD0, JH0, JI0 (DB2 data sharing)

Non-Production Systems:
TPN (runs TPNS scripts)
Z1, Z2, Z3 (test systems)

Note (1): The IBM Communications Server has 2 components: SNA (VTAM) and IP.

Figure 3. Our Sysplex Software Configuration

We run three separate application groups (AGs) in one sysplex. AG1 and AG2 span 9 production systems in the sysplex, while AG3 spans 12 production systems in the sysplex. Each AG has CICS as the transaction manager; AG1 also has IMS TM as a transaction manager. AG1 has IMS as the database manager; AG2 uses VSAM data; AG3 has DB2 as the database manager. AG1 and AG3 use IRLM as the lock manager; for AG2, serialization is provided through the VSAM record level sharing (RLS) function.

You can refer to Chapter 2 in our December 1995 edition, where we describe in great detail how we were set up when all our AGs were using IMS data, and AG3 was using DB2 data for some transactions. We include our subsystem jobnames and other naming conventions as they were then. We also describe how a transaction is processed in the sysplex, and we use AG3 for our example. The example of a transaction writing to both an IMS and DB2 database is still valid,

although AG3 is no longer doing that. See “Database Product OLTP Workloads” on page 17 for more information about what our AGs are now doing.

Our Naming Conventions

We designed the naming convention for our CICS regions so that the names relate to the application groups and system names that the regions belong to. This is important because:

- Relating a CICS region name to its application groups means we can use wildcards to retrieve information about, or perform other tasks in relation to, a particular application group.
- Relating CICS region names to their respective OS/390 system names means that subsystem jobnames also relate to the system names, which makes operations easier. This also makes using automatic restart management easier for us — we can direct where we want a restart to occur, and we know how to recover when the failed system is back online.

Our CICS regions have names of the form CICS*grsi* where:

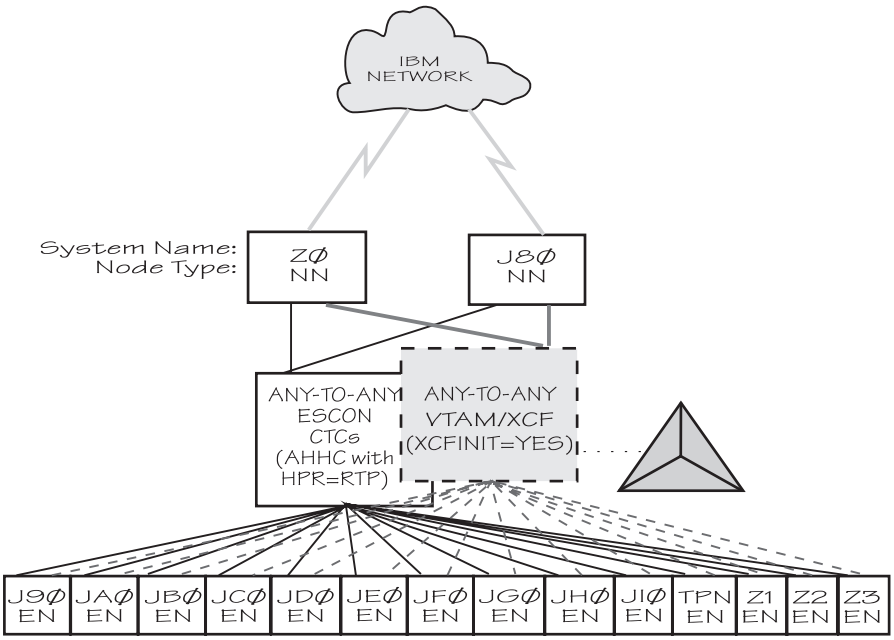
- *g* represents the application group, and can be either 1, 2, or 3
- *r* represents the CICS region type, and can be either A for AORs or T for TORs
- *s* represents the system name, and can be 0 for system Z0, 8 for J80, 9 for J90, and A for JA0 through I for JI0
- *i* represents the instance of the region and can be A, B, or C (we have 3 AORs in each application group on each system)

For example, the CICS region named CICS2A0A would be the first group 2 AOR on system Z0.

Our IMS subsystem jobnames also correspond to their OS/390 system name. They take the form IMS*s* where *s* represents the system name, as explained above for the CICS regions.

Our VTAM Configuration

The following figure is an illustration of our current VTAM configuration. (Note that we have migrated to SecureWay Communications Server SNA. See our December ’99 edition for more information.)



TPNS runs on our system TPN, and routes CICS logons to any of the other systems in the sysplex.

Our VTAM configuration is a pure any-to-any AHHC. Systems ZØ and J8Ø are the network nodes (NNs) and the remaining systems are end nodes (ENs).

We also have any-to-any communication using XCF signalling (where XCF may use either CTCs, coupling facility structures, or both). This is called dynamic definition of VTAM-to-VTAM connections.

We are configured to use both AHHC and XCF signalling for test purposes.

Figure 4. Our VTAM Configuration

Our Workloads

We run a variety of workloads in our pseudo-production environment. Our workloads are similar to those that our customers use. In processing these workloads, we perform many of the same tasks as customer system programmers. Our goal, like yours, is to have our workloads up 24 hours a day, 7 days a week (24 x 7). We have workloads that exercise the sysplex, networking, and application enablement characteristics of our configuration.

The following table shows the workloads we run, both on prime and off shifts:

Table 5. Our Workloads

Shift	OS/390 Base Workloads	Application Enablement Workloads	Networking Workloads	Database Product Workloads
Prime Shift	• Batch pipes • Automatic tape switching • JES2/JES3 printer simulators	• Shelltest (rlogin/telnet) • Shelltest (TSO) • UNIX recursive delete • IBM HTTP Server • WebSphere Application Server	• FTP workloads • MMFACTS for LFS, NFS, and LANRES • AutoWEB • TCP/IP CICS sockets	• CICS DBCTL • IMS SMQ fast path • IMS full function • CICS/RLS online • CICS/DB2 • CICS/QMF online queries • DB2 online reorganization • DB2/RRS stored procedure • QMF batch queries • DB2 Connect
Off Shift	• Random batch • Automatic tape switching • JES2/JES3 printer simulators	• Shelltest (rlogin/telnet) • Shelltest (TSO) • UNIX recursive delete • IBM HTTP Server • WebSphere Application Server	• FTP workloads • LFS video stream • MMFACTS for LFS, NFS, and LANRES	• CICS /DBCTL • CICS RLS online • CICS/DB2 • QMF online queries • CICS/RLS batch • DB2 utility • DB2 DDF • IMS utility

OS/390 Base Workloads

We run the following OS/390 base (MVS) workloads:

Batch Pipes: This is a multi-system batch workload using BatchPipes. It drives high CP utilization of the coupling facility.

Automatic Tape Switching: We run 2 batch workloads to exploit automatic tape switching and the IEFAUTOS coupling facility structure. These workloads use the Virtual Tape Server and DFSMSrmm, as described in our December 1998 edition, and consist of DSSCOPY jobs and DSSDUMP jobs. The DSSCOPY jobs copy particular data sets to tape, while the DSSDUMP jobs copy an entire DASD volume to tape.

Both workloads are set up to run under OPC so that 3 to 5 job streams with hundreds of jobs are all running at the same time to all systems in the sysplex. With WLM-managed initiators, there are no system affinities, so any job can run on any system. In this way we truly exploit the capabilities of automatic tape switching.

JES2/JES3 Printer Simulators: This workload uses the sample functional subsystem (FSS) and the FSS application (FSA) functions for JES2 and JES3 output processing.

Random Batch: This workload is a collection of MVS test cases that invoke many of the functions (both old and new) provided by MVS.

Application Enablement Workloads

We run the following application enablement workloads:

Shelltest (rlogin/telnet): In this workload, users log in remotely from an RS/6000 workstation to the OS/390 shell using either rlogin or telnet and then issue commands.

Shelltest (TSO): In this workload, simulated users driven by the Teleprocessing Network Simulator (TPNS) logon to TSO/E and invoke the OS/390 UNIX shell and issue various commands. The users perform tasks that simulate real OS/390 UNIX users daily jobs, for example:
- Moving data between the HFS and MVS data sets.
- Compiling C programs.
- Running shell programs.

UNIX Recursive Delete: This TPNS driven workload copies over 700 directories from one large HFS to another. It then deletes all directories in the copy with one command.

IBM HTTP Server: This workload, which is driven from both Windows NT and AIX/RISC workstations, accesses the following:
- DB2 LOBs through net.data
- DB2 RRS through net.data
- MVS datasets
- FastCGI programs
- Counters
- Static html pages

IBM WebSphere Application Server: We have the following IBM WebSphere Application Server workloads:
- A workload driven from an AIX/RISC workstation that invokes Java servlets.
- A workload driven from a Windows NT workstation that invokes Java servlets to verify the following for a user request:
  - RACF authentication
  - RACF authentication with Java Database Connectivity (JDBC)
  - LDAP-RACF authentication
  - LDAP-RACF authentication using Java Native Interface (JNI) and C++
  - LDAP-DB2 authentication

Networking Workloads

FTP Workloads: We run the following networking workloads:
- **FTPHFS/DB2:** This client/server workload simulates SQL/DB2 queries via an FTP client.
- **FTPHFS(Linux):** This workload simulates users logging onto a Linux client through telnet or FTP and simulates workloads between the OS/390 servers and the LINUX client.
- **FTP TPNS:** This workload uses TPNS to simulate FTP client connections to the OS/390 server.
- **FTPWL:** This client/server workload automates WINDOWS 95/NT users performing FTP file transfers across Token Ring and Ethernet networks. This workload also exercises the OS/390 Domain Name System (DNS). Files that are transferred reside in both OS/390 HFS and MVS non-VSAM data sets. Future enhancements to this workload will exploit the OS/390 workload manager DNS.

MMFACTS for NFS, LFS, and LANRES: This client/server workload is designed to simulate the delivery of multimedia data streams, such as video, across the network. It moves large volumes of randomly-generated data in a continuous, real-time stream from the server (in our case, OS/390) to the client. Data files can

range in size from 4 MB to 2 Gigabytes. A variety of options allow for variations in such things as frame size and required delivery rates.

AutoWEB: This client/server workload is designed to simulate a user working from a Web Browser. It uses the following HTML meta-statement to automate the loading of a new page after the refresh timer expires:

```
<meta http-equiv='Refresh' content='10; url=file:///xxx.htm'>
```

This workload can drive any file server, such as LANRES, LAN Server or NFS. It also can drive a Web Server by changing the URL from ″url=file:///xxx″ to ″url=http://xxx″

LFS Video Stream: This client/server workload is very similar to that of MMFACTS except that it sends actual video streams across the network instead of simulating them.

TCP/IP CICS Sockets: This TPNS workload exercises TCP/IP CICS sockets to simulate real transactions.

Database Product Workloads

Database Product OLTP Workloads

Our OS/390 sysplex OLTP workloads are our mission critical, primary production workloads. Each of our 3 application groups runs different OLTP workloads using CICS or IMS as the transaction manager:

- Application group 1—IMS data sharing, including IMS shared message queue
- Application group 2—VSAM record level sharing (RLS)
- Application group 3—DB2 data sharing (2 different OLTP workloads, as well as a batch workload).

Note that our OLTP workloads, which are COBOL, FORTRAN, PL1, or C/C++ programs, are Language Environment enabled (that is, they invoke Language Environment support).

IMS Data Sharing Workloads: In application group one, we run three IMS data sharing workloads:

- CICS/DBCTL
- IMS SMQ fast path
- IMS SMQ full function

Highlights of our IMS data sharing workloads include:

- Using full function, fast path, and mixed mode transactions
- Accessing virtual storage option (VSO) and shared sequential dependent (SDEP) databases
- Adding integrity checking on INSERT calls using SDEP journaling
- Adding a batch message processing (BMP) application to do integrity checking on REPLACE calls.

Our IMS data sharing workloads exploit new functions that are available with IMS/ESA V6, including IMS generic resources, shared message queues, shared VSO and shared SDEP.

VSAM RLS Data Sharing Workload: In application group 2, we run one VSAM/RLS data sharing workload. This workload runs transactions similar to the IMS data sharing workload, except that the transactions access VSAM files.

DB2 Data Sharing Workloads In application group 3, we run two different DB2 data sharing OLTP workloads. These workloads are also similar to the IMS data sharing workload running in application group 1.

In the first of the DB2 workloads, we execute 8 different types of transactions in a CICS/DB2 environment. This workload uses databases with simple and partitioned table spaces.

In the second of our DB2 workloads, we use the same CICS regions and the same DB2 data sharing members. However, we use different transactions and different databases. The table space layout is also different for the databases used by the second DB2 workload — it has partitioned table spaces, segmented table spaces, simple table spaces, and partitioned indexes.

Database Product Batch Workloads

We run various batch workloads in our environment, some of which we will describe here. They include:

- IMS Utility
- RLS batch
- DB2 batch workloads

We run our batch workloads under OPC control and use WLM-managed initiators. Our implementation of WLM batch management is described in our December 1997 edition.

DB2 Batch Workloads: Our DB2 batch workloads include:

- DB2 Online reorganization
- DB2/RRS stored procedure
- QMF batch queries
- DB2 utilities
- DB2 DDF

Our DB2 batch workload has close to 2000 jobs that are scheduled using OPC, so that the jobs run in a certain sequence based on their inter-job dependencies.

Tiny Topics

You'll notice as you read our report that we describe our experiences with certain OS/390 elements and features, and certain related products and functions, in greater detail than others. We devote entire sections to those OS/390 elements, features, and related products and functions that play the strongest role in our test environment, and for which we have the most experiences to share with you. Others might be mentioned only in terms of how they fit into the bigger picture.

Here are some points we'd like to mention about OS/390 elements, features, and related products and functions that do not have separate sections in the report:

Table 6. Highlights of Some OS/390 Elements, Features, and Related Products and Functions

OS/390 Element, Feature, Related Product or Function	Details
Using OSA-2 FENET cards between LPARs	We discovered a hardware design restriction with OSA-2 FENET cards whereby LPARs in the same CPC cannot use the same FENET card to communicate with each other. Specifically, in the case of TCP/IP, this means that the IP stacks on different LPARs cannot communicate with each other through a single FENET card. This presents a further problem for OMPROUTE because the LPARs will not learn routing data about each other and your routing tables will be incorrect. To get around this problem, the recommendation from the OSA developers is to install two FENET cards and define *both* cards to *each* LPAR in the CPC that you want to be able to communicate with each other. For TCP/IP, this means defining IP interfaces for both cards on each LPAR's IP stack. In this way, communication between the IP stacks actually occurs between two different FENET cards. OMPROUTE can then also build complete routing tables.
TSO/E	We are all system programmers, so we use TSO/E extensively.
SMS-managed storage	About 70% of our data is SMS-managed.
DFSMShsm	We use DFSMShsm for space and availability management of our data sets on DASD.
Automated tape library (ATL)	We have an 3495 model L40 ATL that allows us to automatically mount and SMS-manage our tapes.
Virtual Tape Server (VTS)	We have a 3494 model L10 VTS that provides 32 virtual 3490 tape drives that are accessible from all systems.
Object access method (OAM)	We use the object access method (OAM) which provides storage and retrieval management for tape volumes contained in system-managed libraries. We also use DFSMSrmm to manage and track the tape volumes used within our tape libraries. See our December 1998 edition for information about how we are using DFSMSrmm.
HCD	We use HCD in the sysplex and have 1 IODF, with combined MVSCP and IOCDS information, shared by all systems in the sysplex. We are exploiting UCBs above the 16 megabyte line. So far we've moved our tape device UCBs, some of our CTCs (those used by TCP/IP), and some of our DASD devices above the line. The DASD devices we've moved include those volumes that house our SYSRES, various logs and SYSLOGs, EREP, and LOGREC data, as well as SYS1.PARMLIB, IODFs, and SYS0.IPLPARM. When we moved the UCBs for the DASD volumes containing SYSRES, SYS1.PARMLIB, and IODFs, we could not make the changes in HCD dynamically. These changes required a POR of each CPC and IPL of each system, which we did 1 CPC at a time during a service window. **Note:** We had a problem when we dynamically activated our TCP/IP CTC UCBs above the line. The MIH value is set to 3 minutes after the dynamic change, and our devices were timing out and becoming unusable. To avoid this, we set the MIH value for these CTCs in our IECIOSST parmlib member as follows: `MIH TIME=00:00,DEV=(C40C-C40F)`
BookManager Read/MVS	We use BookManager READ/MVS to view softcopy books.
BDT networking	We established BDT networking between JES3 and JES2 nodes.
GDDM	We exercise GDDM through our use of graphics options on the RMF Monitor III reports.

Table 6. Highlights of Some OS/390 Elements, Features, and Related Products and Functions (continued)

OS/390 Element, Feature, Related Product or Function	Details
OPC	We've installed Operations Planning and Control (OPC), a production workload scheduler, to test together with SmartBatch, and to use for scheduling some of our other batch workloads. For example, we have a DB2 batch workload (described in "Our Workloads" on page 14) that has close to 300 jobs that are scheduled using OPC, so that the jobs run in a certain sequence based on their inter-job dependencies. We also use OPC to schedule DB2 image copy jobs to run twice a week, and to schedule our CICS VSAM Recovery (CICSVR) log of logs scan every day. See our companion document, *OS/390 Parallel Sysplex Recovery*, for information about how we use CICSVR.
Network Stations	We have multiple Network Stations set up on which we can start terminal sessions. The IBM Network Station is a ready-to-go network computer (NC) that connects to a LAN and can access existing business applications from any server on a network.
Dynamic LPA	We've tested the dynamic LPA support that became available with OS/390 R4. This support allows you to add modules to LPA, and remove them, after IPL.
Multiple SYSRES volumes	We've implemented indirect volume serial support so we can use multiple SYSRES volumes. See our December 1997 edition for more information. Note that our SYSRES layout follows the recommended data set placement described in *OS/390 Planning for Installation*.

Chapter 2. Migrating To and Using OS/390

In this chapter of each of our editions, we describe our experiences migrating to and using each release of OS/390. In this edition, we describe our experiences migrating to OS/390 R9 and R10. Our experiences with previous releases of OS/390 are in:

- Our December 1996 edition for OS/390 R1 and R2
- Our December 1997 edition for OS/390 R3 and R4
- Our December 1998 edition for OS/390 R5 and R6
- Our December 1999 edition for OS/390 R7 and R8

We use this chapter primarily to discuss our sysplex-related base operating system experiences, including performance aspects. We discuss our networking and application-enablement environment and experiences in “Part 2. Networking and Application Enablement” on page 61.

Migrating to OS/390 R9

OS/390 R9 Base Migration Experiences

In this section we describe experiences related only to our migration to OS/390 R9, without having implemented any new functions. Experiences related to implementing a new function are described in the section for that function.

Updating the RACF Templates: Before bringing up our first R9 system, we ran the IRRMIN00 utility with PARM=UPDATE to upgrade our existing RACF database templates to the current level of the RACF templates that are shipped with OS/390 R9. (For details, see *OS/390 SecureWay Security Server RACF System Programmer’s Guide*.)

High-Level Migration Process: The high-level process we followed to migrate our sysplex from OS/390 R8 to OS/390 R9 included the following steps:

- We applied compatibility PTFs and APARs to position ourselves for the migration. Most compatibility PTFs and APARs for OS/390 are documented in *OS/390 Planning for Installation*. On page 22 we list additional PTFs that we applied for compatibility. You should make sure you install the fixes for any APARs that relate to your configuration before you migrate.
- We brought up OS/390 R9 on our test system, Z3, and ran it there for one week.
- We then brought up OS/390 R9 on one more test system, Z2, and ran it there for about two weeks.
- We then brought up OS/390 R9 on two JES2 production systems (JB0 and JD0) and a JES3 production system (Z0) and ran it on those systems for about one week.
- We converted our remaining systems to R9 over a 5–week period, making the total time for the migration approximately nine weeks.

Running with Mixed Product Levels: Note that during our migration, we successfully operated our sysplex with mixed product levels, such as:

- OS/390 R8 and OS/390 R9
- OS/390 R8 JES3 and OS/390 R9 JES3

Updating Parmlib for OS/390 R9: We added or updated parmlib members for OS/390 R9. Table 18 on page 207 summarizes the additions and changes we

made, by parmlib member. We continue to use concatenated parmlib support; see below for details. See the samples page on our Web site for examples of some of our parmlib members.

Using Concatenated Parmlib: We created SYS1.PETR9.PARMLIB to contain all the parmlib members that changed for OS/390 R9, and we used our LOADxx member for migrating our systems one at a time. See our December 1997 edition for an example of using LOADxx to migrate individual systems.

This is a good use of concatenated parmlib; not only does it isolate all the R9 parmlib changes in one place, it also makes it easier to migrate each system. Rather than having to change many parmlib members each time we bring another system to R9, we just add the PARMLIB statements at the appropriate places in LOADxx to allow that system to use SYS1.PETR9.PARMLIB.

Data Set Names for JES2 Libraries: We made no changes to the middle level qualifiers (MLQs) for JES2 data sets (the MLQs remained as V2R8M0):

New Data Set Names for JES3 Libraries: We made the following changes to handle the new middle level qualifiers (MLQs) for JES3 data sets (the MLQs changed from V2R8M0 to V2R9M0):
- Cataloged the new JES3 libraries.
- Updated PROGxx to add the JES3 libraries to our LNKLST sets and to the APF list.
- Updated LPALSTxx to add the JES3 library SIATLPA.

Recompiling the JES2 and JES3 OPC/ESA Exits: If you use OPC, be aware that there are OPC JES2 and JES3 exit routines that have new entry points.

For JES2, the exit name is EXIT7, and the sample provided in the SEQQSAMP library is called EQQXIT74. We chose to rename the exit to OPCAXIT7.

For JES3, the exit names are IATUX19 and IATUX29.

We assemble and link-edit these exits ourselves. As a result, when we migrated to OS/390 R9 JES3, we had to recompile the exits with the new JES3 libraries to prevent abends. When we migrated to OS/390 R9 JES2, we did not have to recompile the exits because the JES2 libraries were the same in R9 as they were in R8.

A way to avoid this problem is to install the exits as SMP/E usermods in the CSI where JES is included, in which case SMP/E automatically reassembles the exits if there are changes to the JES control blocks that OPC/ESA depends on.

Recompiling REXX EXECs for Automation: We had to recompile our SA OS/390 REXX EXECs when we migrated to OS/390 R9. This is similar to what we did when we migrated to OS/390 R4, as discussed in our December 1997 edition.

HFS Migration Considerations: See “Migrating to a Shared Hierarchical File System (HFS)” on page 109 for a discussion of our OS/390 R9 shared HFS migration.

OS/390 R9 Compatibility APARs: The majority of the compatibility PTFs and APARs needed to position yourself for the migration to OS/390 R9 are documented in *OS/390 Planning for Installation*. We also want to highlight the following APARs that we feel are important:

Table 7. Additional Compatibility APARs for Migrating to OS/390 R9.

Product/Function	APAR Number	Comments
OS/390 R6 and higher (MVS element, XES component)	OW34514	Provides compatibility and coexistence support for structure full monitoring. Install the PTF on *all* downlevel systems in the sysplex before migrating *any* of the systems to OS/390 R9.
OS/390 R6 and higher (MVS element, XCF component)	OW39891	Provides compatibility support for new ARM functions. This PE APAR fixes PTFs in error from APAR OW35888.
OS/390 R6 and higher (MVS element, RRS component)	OW42042	Provides toleration support for RRS changes. This PE HIPER APAR fixes PTFs in error from APAR OW40061.
JES3	OW40166	Provides fallback and coexistence support for new initialization stream function. Note that PTFs are in error for HJS5521, HJS6601, and HJS6603 only; the fixing APAR is OW41897.

Setting Up Multi-System Enclave Support for WLM

With multisystem enclave support, enclaves can span address spaces on multiple systems in a Parallel Sysplex. As in a single system enclave, WLM manages and reports on the work as a single unit.

Before you can use multisystem enclaves, you need to define a specific coupling facility structure named SYSZWLM_WORKUNIT in the CFRM policy. Once you activate the CFRM policy containing this structure definition, WLM automatically connects to the structure, enabling the use of multisystem enclaves. Note that this function requires the coupling facility to be at CFLEVEL 9.

We defined the SYSZWLM_WORKUNIT cache structure in a CFRM policy as described in *OS/390 MVS Planning: Workload Management*. After running IXCMIAPU to load our new CFRM policy, we issued a SETXCF command to activate it:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policyname
```

Afterwards, each system in the sysplex issued the following messages as WLM automatically connected to the new structure:

```
IXL014I IXLCONN REQUEST FOR STRUCTURE SYSZWLM_WORKUNIT
WAS SUCCESSFUL.  JOBNAME: WLM ASID: 000C
CONNECTOR NAME: #Z3 CFNAME: CF2
IWM050I STRUCTURE(SYSZWLM_WORKUNIT), CONNECTED
IXL015I STRUCTURE ALLOCATION INFORMATION FOR
STRUCTURE SYSZWLM_WORKUNIT, CONNECTOR NAME #Z3
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   --------------------------------
 CF2        STRUCTURE ALLOCATED
```

The D WLM command also indicates that the structure is connected:

```
D WLM

IWM025I  09.58.16  WLM DISPLAY
  ACTIVE WORKLOAD MANAGEMENT SERVICE POLICY NAME: WLMPOL01
  ACTIVATED: 2000/01/03  AT: 16:11:29  BY: DBLISS    FROM: J90
  DESCRIPTION: WLM SERVICE POLICY FOR GOAL MODE
  RELATED SERVICE DEFINITION NAME: WLMDEF01
```

WLM Multi-System Enclaves

```
INSTALLED: 2000/01/03  AT: 16:11:10  BY: DBLISS    FROM: J90
WLM VERSION LEVEL:       LEVEL008
WLM FUNCTIONALITY LEVEL: LEVEL004
WLM CDS FORMAT LEVEL:    FORMAT 3
STRUCTURE SYSZWLM_WORKUNIT STATUS: CONNECTED
```

Thus far, we have only set up the capability to use multisystem enclaves, but we don't currently run any workloads that exploit them. See *OS/390 MVS Programming: Workload Management Services* for more information on using multisystem enclaves.

XES Enhancements

OS/390 R9 introduces several XES enhancements that we want to highlight.

Structure Full Monitoring

Structure full monitoring adds support for the monitoring of objects within coupling facility structures. Its objective is to determine the level of usage for monitored objects and issue a warning message if a structure full condition is imminent. This allows an installation to intervene, either manually or through automation, and take the appropriate diagnostic or tuning actions to avoid the structure full condition.

Structure full monitoring periodically retrieves structure statistics and calculates a structure's percent full value. When a structure's percent full value reaches or exceeds a threshold in terms of any of the structure objects that it contains, the system issues highlighted message IXC585E to the console and to the system message logs.

Originally, OS/390 R9 used a threshold value of 80 percent for all structures. However, APAR OW41051 introduces an optional `FULLTHRESHOLD(nnn)` keyword to allow you to specify different thresholds (*nnn* ranges from 0 to 100) for individual structures in the CFRM policy. The default threshold remains at 80 percent for any structures for which you do not specify a threshold.

In our testing, we lowered the threshold for the LOGGER_OPERLOG structure to 70. The following is an example of the threshold warning message:

```
IXC585E STRUCTURE LOGGER_OPERLOG IN COUPLING FACILITY CF1,
PHYSICAL STRUCTURE VERSION B3B3A462 87C8EC00,
IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF  70%.
ENTRIES:  IN-USE:      26143 TOTAL:      30536,  85% FULL
ELEMENTS: IN-USE:      54275 TOTAL:      91607,  59% FULL
```

When utilization decreases such that the percent full value for all structure objects falls below the threshold, the system deletes message IXC585E and issues message IXC586I.

If structure full monitoring is discontinued for any reason (for example, if the structure is deallocated), while there is an outstanding IXC585E message, the system deletes message IXC585E and issues message IXC587I.

In the course of our testing, we received each of the above messages as expected. No special hardware or software changes are needed. However, once you install APAR OW41051, you can update the CFRM policy to include the `FULLTHRESHOLD(nnn)` keyword to specify a threshold value on a structure-by-structure basis. Note that setting the threshold to 0 disables structure full monitoring for that structure. See *OS/390 MVS Setting Up a Sysplex* for further information.

XES Event Supression

On systems with OW38840 installed or which are at OS/390 Release 9 or higher, connectors to a cache structure may request that the system suppress certain connection and disconnection events that the connector might otherwise generate. Suppressing these events (New Connection, Existing Connection, Rebuild New Connection, Rebuild Existing Connection, and Disconnected or Failed Connection) may provide a significant performance benefit at connect time to connectors who do not need the information presented in these events. Connectors can use the new IXLCONN **SUPPRESSEVENTS** keyword to request event supression.

IMS Batch with PQ26416 and PQ26491 is one exploiter of this new function. To test this, we installed the above service and ran our IMS Batch workload. While the jobs ran, we monitored the OSAMCACHE1 structure for any failed connections, and we saw none. Previously, delays caused by excessive event overhead would result in some failed connections.

For more information, see *OS/390 MVS Programming: Sysplex Services Guide* and *OS/390 MVS Programming: Sysplex Services Reference*.

Minimum Required CFLEVEL

Applications can specify the new IXLCONN **MINCFLEVEL** keyword to specify the minimum coupling facility level in which a structure can be allocated. Although we did not specifically test for this, WLM is an example of an exploiter of this new function.

For more information, see *OS/390 MVS Programming: Sysplex Services Guide* and *OS/390 MVS Programming: Sysplex Services Reference*.

CFRM Policy Now Allows Up To 512 Structures

OS/390 R9, and earlier releases with OW19974, now allow up to 512 structures to be formatted into a CFRM couple data set and defined in a CFRM policy. This change provides constraint relief for some very large parallel sysplex configurations. However, note that the recommendation against *over* defining the number of structures in your CFRM couple data set still stands. In addition, note that falling back to a CFRM couple data set formatted for 255 structures or less requires a sysplex-wide IPL. See *OS/390 MVS Setting Up a Sysplex* for more information.

Recovery Enhancements

OS/390 R9 (and earlier releases with OW33615) provides several recovery enhancements that are generally applicable and useful for a variety of coupling facility environments, but are particularly relevant to coupling facilities that are not failure-isolated. A non-failure-isolated coupling facility is one where the coupling facility LPAR and an OS/390 LPAR in the same Parallel Sysplex reside on the same CPC.

The following is a summary of the scenarios that these recovery enhancements address:

- *When a coupling facility fails along with all OS/390 member images running on the same CPC, 100 percent loss of connectivity is not detected.*

  An enhancement now allows XES to report a 100 percent loss of connectivity if a coupling facility loss of connectivity is observed by all failure-isolated systems, but is *not* observed by all systems that are *not* failure-isolated.
- *Failed-persistent connections to a failed structure in a coupling facility that is no longer connected can prevent attempts to allocate a new structure.*

  Now, when all systems lose connectivity to a coupling facility, any attempt to connect to a structure in that coupling facility will cause the system to

automatically FORCE (deallocate) the structure. This allows the connection attempt to succeed by allocating a new structure in an accessible coupling facility.

- *Connectors in a failing state can prevent connections to a new structure until the associated cleanup completes.*

  An enhancement now allows XES to internally retry such transient connect failures for a period of time before failing the connect request back to the requester.

- *XES can reorder the CFRM preference list based on how well various coupling facilities provide the requested coupling facility attributes (such as available space, requested CFLEVEL, failure isolation, nonvolatility).*

  An enhancement provides a new CFRM policy option, ENFORCEORDER(NO|YES), to cause XES to strictly enforce a structure's preference list order. With ENFORCEORDER(YES), XES can still drop ineligible coupling facilities from the list, but it will not reorder the sequence of the coupling facilities in the preference list.

- *A losscon percentage is not calculated when SFM is not active, even if all systems lose connectivity.*

  An enhancement now presents a losscon percentage of 100% when a 100% loss of connectivity occurs, even when SFM is not active. (Note that when SFM is not active, XES still will not calculate nor present a losscon percentage for *partial* losses of connectivity.)

For more information about these recovery scenarios, see *OS/390 MVS Programming: Sysplex Services Guide* and *OS/390 MVS Setting Up a Sysplex*. Also, refer to flash W98029 on the Technical Information Web site at:

`http://www.ibm.com/support/techdocs/`

Rebuild Quiesce Enhancements

OS/390 R9, and earlier releases with OW36894, now inform connectors of the LOCATION and LESSCONNACTION attributes for the current rebuild process by adding two new parameter list fields on the Rebuild Quiesce event exit. For more information, see *OS/390 MVS Programming: Sysplex Services Guide*.

Migrating to OS/390 R10

OS/390 R10 Base Migration Experiences

In this section we describe experiences related only to our migration to OS/390 R10, without having implemented any new functions. Experiences related to implementing a new function are described in the section for that function.

Updating the RACF Templates: Before bringing up our first R10 system, we ran the IRRMIN00 utility with PARM=UPDATE to upgrade our existing RACF database templates to the current level of the RACF templates that ship with OS/390 R10. (For details, see *OS/390 SecureWay Security Server RACF System Programmer's Guide*.)

Updating the Microcode Level on the IBM Enterprise Storage Server: If you use an IBM Enterprise Storage Server (also known as "Shark"), make sure you install the latest microcode level before you bring up your first OS/390 R10 system. We didn't know this when we began our migration and it caused us some problems. If you have any doubts, check with your IBM support representative for the appropriate microcode level for your environment.

High-Level Migration Process: The high-level process we followed to migrate our sysplex from OS/390 R9 to OS/390 R10 included the following steps:

- We applied the necessary coexistence service (also known as compatibility or toleration PTFs) to position ourselves for the migration. Coexistence service for OS/390 releases is documented in *OS/390 Planning for Installation*. You should make sure you install the fixes for any APARs that relate to your configuration before you migrate.
- We brought up OS/390 R10 on one of our test systems (Z2) and ran it there for about one week.
- We then brought up OS/390 R10 on another test system (Z3), a JES2 production system (JA0), and a JES3 production system (JF0, which happens to be a JES3 local) and ran it on those systems for about one week.
- We then brought up OS/390 R10 on three more systems (JH0, J80, and Z0) and ran it there for about one week.

  At about this time, we also shut down three of our systems (JD0, JI0, and Z1) and removed them from our sysplex configuration. We did this because the processors on which those systems ran do not support OS/390 R10. For more information, see "Our Sysplex Hardware Configuration" on page 3.
- We migrated our remaining six systems to R10 over the next two weeks, making the total time for the migration approximately five weeks.

Running with Mixed Product Levels: Note that during our migration, we successfully operated our sysplex with mixed product levels, such as:

- OS/390 R9 and OS/390 R10
- OS/390 R8 JES2 and OS/390 R10 JES2
- OS/390 R9 JES3 and OS/390 R10 JES3

Updating Parmlib for OS/390 R10: We added or updated parmlib members for OS/390 R10. Table 18 on page 207 summarizes the additions and changes we made, by parmlib member. We continue to use concatenated parmlib support; see below for details. See the samples page on our Web site for examples of some of our parmlib members.

Using Concatenated Parmlib: We created SYS1.PETR10.PARMLIB to contain all the parmlib members that changed for OS/390 R10, and we used our LOADxx member for migrating our systems one at a time. See our December 1997 edition for an example of using LOADxx to migrate individual systems.

This is a good use of concatenated parmlib; not only does it isolate all the R10 parmlib changes in one place, it also makes it easier to migrate each system. Rather than having to change many parmlib members each time we bring another system to R10, we just add the PARMLIB statements at the appropriate places in LOADxx to allow that system to use SYS1.PETR10.PARMLIB.

Data Set Names for JES2 Libraries: We made the following changes to handle the new middle level qualifier (MLQ) for JES2 data sets (since JES2 did not change in OS/390 R9, the MLQ changed from V2R8M0 to V2R10M0):

- Cataloged the new JES2 libraries
- Updated PROGxx to add the JES2 libraries to our LNKLST sets and to the APF list

New Data Set Names for JES3 Libraries: We made the following changes to handle the new middle level qualifier for JES3 data sets (the MLQ changed from V2R9M0 to V2R10M0):

- Cataloged the new JES3 libraries
- Updated PROGxx to add the JES3 libraries to our LNKLST sets and to the APF list
- Updated LPALSTxx to add the JES3 library SIATLPA

Recompiling the JES2 and JES3 OPC/ESA Exits: If you use OPC, be aware that there are OPC JES2 and JES3 exit routines that have new entry points.

For JES2, the exit name is EXIT7, and the sample provided in the SEQQSAMP library is called EQQXIT74. We chose to rename the exit to OPCAXIT7.

For JES3, the exit names are IATUX19 and IATUX29.

We assemble and link-edit these exits ourselves. As a result, when we migrated to OS/390 R10 JES2 and JES3, we had to recompile the JES2 and JES3 exits with the new JES2 and JES3 libraries, respectively, to prevent abends. (Note that in OS/390 R10, the high level qualifier for the High Level Assembler changes from HLA to ASM.)

A way to avoid this problem is to install the exits as SMP/E usermods in the CSI where JES is included, in which case SMP/E automatically reassembles the exits if there are changes to the JES control blocks that OPC/ESA depends on.

Recompiling REXX EXECs for Automation: We had to recompile our SA OS/390 REXX EXECs when we migrated to OS/390 R10. This is similar to what we did when we migrated to OS/390 R4, as discussed in our December 1997 edition.

OS/390 R10 Compatibility APARs: The compatibility PTFs and APARs needed to position your systems for the migration to OS/390 R10 are documented in *OS/390 Planning for Installation* in the chapter on ensuring coexistence. We applied the compatibility service as prescribed in the book. We also want to highlight the following additional information about one of the listed APARs.

APAR OW36418 requires that you first install the fix for APAR OW43304. APAR OW36418 fixes a potential problem with contention management in a GRS STAR environment while OW43304 provides compatibility support to allow systems in a sysplex to migrate to the corrected contention management processing without requiring a sysplex-wide IPL. Therefore, you must install and IPL the fix for OW43304 on ***every*** system in the sysplex before you install and IPL the fix for OW36418 on ***any*** system in the sysplex. See the APAR descriptions for details.

Enabling Automatic Tuning of Coupling Facility Structures

When we reported on our migration to OS/390 R9, we introduced structure full monitoring which added support to monitor objects within coupling facility structures and to issue a warning message if a structure full condition is imminent (see “Structure Full Monitoring” on page 24). Now, OS/390 R10 adds new function to allow the system to automatically tune structures that are at or over their percent full thresholds.

How XCF Automatic Alter Support Works

Given the increasingly dynamic nature of workloads, it is difficult to know in advance what the actual usage of entries, elements, and event monitor controls (EMCs) will be for a given structure. Along with structure full monitoring, the XCF automatic alter support new in OS/390 R10 helps relieve the complex tasks of planning and calculating structure sizes by allowing you to make an educated guess about the minimum, maximum, and initial structure sizes. XCF then monitors and

automatically tunes the structure size and entry-to-element ratio in response to changing structure object usage. It does this in real time by calculating a structure's size, ratio, and EMC storage percentage and issuing an alter request if the structure is at or above its percent full threshold.

New CFRM Policy Parameters for Automatic Alter: OS/390 R10 adds two new parameters for structure definitions in the CFRM policy to support automatic alter:

- ALLOWAUTOALT indicates whether the structure's size is allowed to be automatically tuned. The default is ALLOWAUTOALT(NO).
- MINSIZE defines the lower bound on the structure size such that XCF will not allocate or alter a structure to be smaller than this value. If you do not specify a value for MINSIZE, it defaults to 50% of INITSIZE; if you do not specify INITSIZE, then MINSIZE defaults to 50% of SIZE.

Structure Eligibility for Automatic Alter: A coupling facility structure is eligible to be automatically altered when all of the following conditions are true:

- The structure resides in a coupling facility running at CFLEVEL 8 or higher.
- All connectors to the structure allow the structure to be altered.
- Structure full monitoring is active (that is, the active CFRM policy specifies a non-zero FULLTHRESHOLD value for the structure).
- The active CFRM policy specifies ALLOWAUTOALT(YES) on the structure definition.
- The structure is not currently being rebuilt. (The exception is that a structure in the duplex established phase of a user-managed duplexing rebuild is still considered eligible.)

Coupling Facility Storage Constraint Relief: XCF can also use automatic alter to dynamically relieve a coupling facility storage constraint by contracting one or more eligible structures whose storage is not being productively used. XCF considers a coupling facility to be storage constrained when it reaches 80% full. Only those structures that are eligible for automatic alter can be candidates for contraction. Furthermore, a structure cannot be contracted to a size that is less than the minimum size (MINSIZE) defined in the CFRM policy.

Migration and Coexistence With Earlier Releases of OS/390: Automatic alter requires no compatibility PTFs for migration or coexistence. When a system running OS/390 R10 exists in the same sysplex with systems running earlier releases of OS/390, the following behavior applies:

- When an OS/390 R10 system gains connectivity to a coupling facility that contains structures that are being monitored by an OS/390 R9 system, the R10 system will take over monitoring responsibilities and will provide automatic alter support for eligible structures.
- Systems running OS/390 R8 or earlier cannot perform structure full monitoring. Therefore, the OS/390 R10 system will automatically assume monitoring responsibilities and will provide automatic alter support for eligible structures.

Experiences With Automatic Alter

We enabled the following structures to be automatically altered:

DB2 IRLM lock
DB2 group buffer pools
DB2 SCA
DFSMS enhanced catalog sharing
DFSMS VSAM RLS lock
IMS IRLM lock
IMS OSAM cache
IMS VSAM cache
WLM multi-system enclaves
XCF signalling

Automatic Alter

As an example, here's what we did to enable the DFSMS enhanced catalog sharing (ECS) structure, SYSIGGCAS_ECS, for automatic alter. We defined the ECS structure in the CFRM policy with an initial size of 256 KB, a full threshold of 75%, and automatic alter enabled, as follows:

```
STRUCTURE NAME(SYSIGGCAS_ECS)
            INITSIZE(256)
            SIZE(512)
            PREFLIST(CF2,CF1)
            FULLTHRESHOLD(75)
            ALLOWAUTOALT(YES)
```

We activated the new CFRM policy and then displayed the SYSIGGCAS_ECS structure:

```
IXC360I  hh.mm.ss  DISPLAY XCF
STRNAME: SYSIGGCAS_ECS
 STATUS: ALLOCATED
  POLICY SIZE    : 512 K
  POLICY INITSIZE: 256 K
  POLICY MINSIZE : 128 K
  FULLTHRESHOLD  : 75
  ALLOWAUTOALT   : YES
  REBUILD PERCENT: N/A
  DUPLEX         : DISABLED
  PREFERENCE LIST: CF2      CF1
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

Notice that the structure display also indicates the POLICY MINSIZE. We did not specify a MINSIZE for the structure in the CFRM policy, so the system used the default of 50% of the INITSIZE.

Take Note!

The first time you start a new CFRM policy generated under OS/390 R10, you will see messages that indicate that policy changes are pending for all currently allocated structures. This is because the system adds a MINSIZE value—either an explicit value that you specify in the policy or a default value that it calculates as described earlier—for every structure.

The first time we started a new CFRM policy under R10, we were surprised to see that there were 94 policy changes pending:

```
IXC511I START ADMINISTRATIVE POLICY IXCALLST FOR CFRM ACCEPTED
IXC512I POLICY CHANGE IN PROGRESS FOR CFRM
TO MAKE IXCALLST POLICY ACTIVE.
94 POLICY CHANGE(S) PENDING.
```

We did not expect to see any policy changes pending because the only change we made was to add three new structures. So, we naturally assumed we had a problem. What we didn't realize was that the system had automatically added a default MINSIZE to all of the other allocated structures and that this accounted for the pending policy changes. Unaware of this at the time, we looked up the explanation of the IXC512I message to see if it would give us a clue about why we suddenly had so many policy changes pending.

In the description of the IXC512I message, the system programmer response suggests that you might need to format new CFRM couple data sets to accommodate a larger number of STR records; however, this action does not apply in this particular case. (We tried it, and it had no effect on the number of pending policy changes.) Instead, we just had to rebuild each structure to allow the pending changes to complete. (You can use the D XCF,STR,STAT=(POLCHG) command to find out which structures have policy changes pending.)

We learned afterwards that this only happens the first time you start a new CFRM policy under OS/390 R10. Once the policy is in effect and all structures have a MINSIZE, subsequently starting new CFRM policies proceeds as you would normally expect.

We let the system run and waited to see what would happen as the usage of the ECS structure increased. When structure full monitoring determined that the ECS structure was at or over its percent full threshold (indicated by highlighted message IXC585E), an automatic alter was initiated (new message IXC588I):

```
JC0      2000200 02:00:37.84
  *IXC585E STRUCTURE SYSIGGCAS_ECS IN COUPLING FACILITY CF2
   PHYSICAL STRUCTURE VERSION B4570FBB CA3B6A41,
   IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF  75%.
   ENTRIES:  IN-USE:         42 TOTAL:         61,  68% FULL
   ELEMENTS: IN-USE:         42 TOTAL:         42, 100% FULL

   IXC588I AUTOMATIC ALTER PROCESSING INITIATED
   FOR STRUCTURE SYSIGGCAS_ECS.
   CURRENT SIZE:        256 K
   TARGET SIZE:         512 K
   TARGET ENTRY TO ELEMENT RATIO:          1 :          1
```

Automatic Alter

When alter processing completed, the system deleted the outstanding IXC585E message and issued the following messages to indicate that the alter was successful (new message IXC590I) and structure usage was below the percent full threshold:

```
JC0      2000200 02:00:46.93
   IXC590I AUTOMATIC ALTER PROCESSING FOR STRUCTURE SYSIGGCAS_ECS
   COMPLETED.  TARGET ATTAINED.
   CURRENT SIZE:                512 K  TARGET:        512 K
   CURRENT ENTRY COUNT:         122    TARGET:        122
   CURRENT ELEMENT COUNT:       103    TARGET:        103
   CURRENT EMC COUNT:             0    TARGET:          0

JC0      2000200 02:01:40.96
   IXC586I STRUCTURE SYSIGGCAS_ECS IN COUPLING FACILITY CF2,
   PHYSICAL STRUCTURE VERSION B4570FBB CA3B6A41,
   IS NOW BELOW STRUCTURE FULL MONITORING THRESHOLD.
```

If automatic alter processing cannot complete, the system instead issues message IXC589I to indicate the reason for the failure.

For more information about structure full monitoring and automatic alter, see *OS/390 MVS Setting Up a Sysplex*. Also, for information about how to code connectors to exploit these functions, see *OS/390 MVS Programming: Sysplex Services Guide* and *OS/390 MVS Programming: Sysplex Services Reference*.

Enabling SDSF Sysplex Support

In OS/390 R10, SDSF adds capability for sysplex-wide job output displays. Specifically, this involves adding sysplex-wide support for the PR and INIT device panels so that they now show all printers and initiators for all systems in the MAS, no matter which system you're logged on to. SDSF's browse function, which lets you browse a job's output and the syslog, has also been enhanced to include data not yet written to spool from any system in the sysplex. This means that SDSF's browse function will always show the most complete and current information for active jobs and the syslog from any system in the sysplex.

In addition, the sysplex support for the PR panel requires that your destination definitions in the JES2 initialization parameters be consistent for all systems that are to participate.

Note that at present, SDSF does not support MQSeries clustering.

We did the following to enable SDSF sysplex support:

Installed MQSeries for OS/390 on Each Participating System: On each system where we wanted to use the SDSF sysplex support, we made sure we had MQSeries for OS/390 at the 2.1 level. See "Chapter 6. Setting Up MQSeries" on page 57 for details.

Setup the SDSF Server: SDSF sysplex support requires that you have an SDSF server set up on each system participating in SDSF sysplex support. The SDSF server is an address space that SDSF uses to process ISFPARMS statements and provide sysplex data on the PR, INIT, and browse panels. We used the *OS/390 SDSF Customization and Security* book to do this.

Added A Server Group Definition to ISFPARMS: SDSF sysplex support requires that we define a server group for our SDSF servers in ISFPARMS. A server group

is the group of SDSF servers that provide sysplex data. Within the server group definition, each SDSF server is specifically defined to the group. For each server in the group, you must define:

- The name of the server (note that we've named all of our servers the same thing - SDSF)
- The system where the server resides
- The related primary or secondary JES2s that participate in a sysplex-wide request
- The method of communication between the servers

Note that all SDSF servers in the group must be in the same sysplex, and all associated JES's must be in the same MAS.

For our environment, we defined the following servers to our server group:

```
    /*******************************************/
    /* Server group definitions                */
    /*******************************************/

SERVERGROUP
SERVER NAME(SDSF),SYSNAME(J80),JESNAME(JES2),MEMBER(J80),COMM(COMM8)
SERVER NAME(SDSF),SYSNAME(J90),JESNAME(JES2),MEMBER(J90),COMM(COMM9)
SERVER NAME(SDSF),SYSNAME(JA0),JESNAME(JES2),MEMBER(JA0),COMM(COMMA)
SERVER NAME(SDSF),SYSNAME(JB0),JESNAME(JES2),MEMBER(JB0),COMM(COMMB)
SERVER NAME(SDSF),SYSNAME(JC0),JESNAME(JES2),MEMBER(JC0),COMM(COMMC)
SERVER NAME(SDSF),SYSNAME(JE0),JESNAME(JES2),MEMBER(JE0),COMM(COMME)
SERVER NAME(SDSF),SYSNAME(JF0),JESNAME(JES2),MEMBER(JF0),COMM(COMMF)
SERVER NAME(SDSF),SYSNAME(JG0),JESNAME(JES2),MEMBER(JG0),COMM(COMMG)
SERVER NAME(SDSF),SYSNAME(JH0),JESNAME(JES2),MEMBER(JH0),COMM(COMMH)
SERVER NAME(SDSF),SYSNAME(TPN),JESNAME(JES2),MEMBER(TPN),COMM(COMMT)
SERVER NAME(SDSF),SYSNAME(Z0),JESNAME(JES2),MEMBER(Z0),COMM(COMM0)
SERVER NAME(SDSF),SYSNAME(Z2),JESNAME(JES2),MEMBER(Z2),COMM(COMM2)
SERVER NAME(SDSF),SYSNAME(Z3),JESNAME(JES2),MEMBER(Z3),COMM(COMM3)
COMM NAME(COMM8),TYPE(MQS),QMGR(CSQ8),QPREFIX(ISF)
COMM NAME(COMM9),TYPE(MQS),QMGR(CSQ9),QPREFIX(ISF)
COMM NAME(COMMA),TYPE(MQS),QMGR(CSQA),QPREFIX(ISF)
COMM NAME(COMMB),TYPE(MQS),QMGR(CSQB),QPREFIX(ISF)
COMM NAME(COMMC),TYPE(MQS),QMGR(CSQC),QPREFIX(ISF)
COMM NAME(COMME),TYPE(MQS),QMGR(CSQE),QPREFIX(ISF)
COMM NAME(COMMF),TYPE(MQS),QMGR(CSQF),QPREFIX(ISF)
COMM NAME(COMMG),TYPE(MQS),QMGR(CSQG),QPREFIX(ISF)
COMM NAME(COMMH),TYPE(MQS),QMGR(CSQH),QPREFIX(ISF)
COMM NAME(COMMT),TYPE(MQS),QMGR(CSQT),QPREFIX(ISF)
COMM NAME(COMM0),TYPE(MQS),QMGR(CSQ0),QPREFIX(ISF)
COMM NAME(COMM2),TYPE(MQS),QMGR(CSQ2),QPREFIX(ISF)
COMM NAME(COMM3),TYPE(MQS),QMGR(CSQ3),QPREFIX(ISF)
```

Note that if a system is not defined to the server group, SDSF does not gather the data from that system.

Communication Between Systems: In order to use SDSF sysplex support, each participating system in the sysplex must have MQSeries connectivity to every other participating system to enable communication between SDSF servers and queue managers, including remote queues. We ensured this connectivity using queue manager aliases. For each remote queue manager, we create a queue manager aliases definition. This one definition takes care of routing requests to any of the queues managed by that remote queue manager because it relates a queue manager name to a transmission queue, which stores the messages until they can be transmitted to a remote queue manager. This saves us some work - we define one queue manager alias instead of creating a separate remote queue for each queue SDSF uses on a remote queue manager. For an explanation of

communication between queue managers, start with the *OS/390 SDSF Customization and Security* book. Then, for complete details, see *MQSeries Intercommunication*.

For each participating system, we needed the following MQSeries definitions:

- One sender channel
- One receiver channel
- One transmit queue
- One process
- One queue manager alias

Use the MQSeries operations and control panels or other MQSeries methods, to create these definitions. See the *MQSeries Command Reference* for details.

Recycling SDSF After an MQSeries Failure: If MQSeries fails, you may need to recycle SDSF (by exiting and reentering the SDSF panel on TSO/E) after you restart MQSeries in order to restore communication between MQSeries andthe SDSF servers. This is because after an MQSeries failure, SDSF loses access to MQSeries, which provides the communication between servers. This access may not always come back up when you restart MQSeries. Without access to MQSeries, SDSF will not display current sysplex information on the PR and INIT device panels.

If MQSeries fails, you can check to see whether communications are up using the SDSF WHO command to display user information. When communications are up (meaning that SDSF **does** have access to MQSeries), the output from the display command might look as follows:

```
USERID=PETIMS1,PROC=WLMRMF52,TERMINAL=T6FDTN2D,GRPINDEX=1,GRPNAME=ISFSPROG,
MVS=OS/390 02.10.00,JES2=OS 2.10,SDSF=HQX7703,ISPF=5.0,RMF/DA=610,SERVER=YES,
SERVERNAME=SDSF,JESNAME=JES2,MEMBER=Z3,SYSNAME=Z3, COMM=ENABLED
```

When communications are down, (SDSF **does not** have access to MQSeries), the output from the WHO command might look as follows:

```
USERID=PETIMS1,PROC=WLMRMF52,TERMINAL=T6FDTN2D,GRPINDEX=1,GRPNAME=ISFSPROG,
MVS=OS/390 02.10.00,JES2=OS 2.10,SDSF=HQX7703,ISPF=5.0,RMF/DA=610,SERVER=YES,
SERVERNAME=SDSF,JESNAME=JES2,MEMBER=Z3,SYSNAME=Z3, COMM=NOTAVAIL
```

If, after an MQSeries failure, the output of the WHO command shows that communications are not available, you must recycle SDSF to restore communications between MQSeries and the SDSF servers.

Security for SDSF Sysplex Support: SAF security for SDSF's use of MQSeries for OS/390 is described in the *OS/390 SDSF Customization and Security* book. You can also create security definitions using the SDSF configuration assistant, which you can find at the following URL:

```
http://www.s390.ibm.com/os390/wizards
```

OS/390 Performance

How our OS/390 systems perform is an important issue for us, just as it is for you. If we are to be customer-like, we must pay attention to meeting the goals in our service level agreements.

Here is what we do in each phase of our testing, and what we plan to periodically report to you in our test reports:

- Monitor our performance in terms of our service level agreements

This is not formal performance testing for purposes of publishing performance statistics for OS/390. It is a way for us to establish and report on reasonable goals for response times and transaction rates for the various types of work we are doing, just as a customer would do to create a service level agreement (SLA).

Note: Our goal for our sysplex workloads has been, and continues to be, 90% CP utilization across the systems in the sysplex, with an internal response-time goal of 80% of CICS transactions completed in less than 0.6 seconds.

At one time, we attempted to push our OLTP workload to 95% CP utilization, but the result was increased contention for resources, and our response time suffered. So, we are staying within the general guideline of 90% CP utilization for OLTP work, and filling in the remaining 10% with batch work and various additional types of users (OS/390 UNIX System Services (OS/390 UNIX) users, TSO users, and workstation clients).

- Identify performance problems in our environment, find solutions to those problems, and report the information to you.
- Provide you with periodic performance snapshots of our environment, in the form of RMF reports, to provide pertinent information such as how many transactions we process per second and what our response times are for various workloads. You'll find those reports in "Appendix B. Some of Our RMF Reports" on page 211.

Chapter 3. Setting Up Integrated Cryptographic Service Facility (ICSF)

In this chapter, we'll discuss how we set up Integrated Cryptographic Service Facility (ICSF) on the R8 level of OS/390. ICSF is a software element of OS/390 that works with the hardware cryptographic coprocessor feature and the OS/390 Security Server, Resource Access Control Facility (RACF), to provide secure, high-speed cryptographic services in the OS/390 environment.

- The ICSF software provides application programming interfaces by which applications request the cryptographic services.
- The cryptographic coprocessor feature is secure, high-speed hardware that performs the actual cryptographic functions. The cryptographic coprocessor feature is available as part of the processor hardware on IBM S/390 Multiprise 2000 and IBM S/390 Parallel Enterprise Servers at Generation 3 level or higher with feature code 0800. See *OS/390 ICSF System Programmer's Guide* for complete information on cryptographic hardware feature codes.

We set ICSF up using the following books:

- *OS/390 ICSF System Programmer's Guide*
- *OS/390 ICSF Administrator's Guide*
- *S/390 Support Element Operations Guide*

Our ICSF Configuration

We set up ICSF on a S/390 G6 server with three LPAR system images. The cryptographic coprocessor feature on our machine has two cryptographic coprocessors (CPs), each of which is attached to a central processor in the same machine. Each crypto CP has sixteen master key register sets, referred to as domains. The domain is simply an index into a set of master key registers which allows the system image on an LPAR to access the cryptographic coprocessor feature.

You can configure ICSF in two different ways; single-image mode or logical partition (LPAR) mode. (See *OS/390 ICSF System Programmer's Guide* for more information on these configurations.) Because the S/390 G6 server that we use runs in LPAR mode, we set up our ICSF configuration in LPAR mode. Each LPAR or system image accesses the same domain number on each of the crypto CPs. The figure below shows system image JH0's connections to domain 1 on both cryptographic coprocessors:

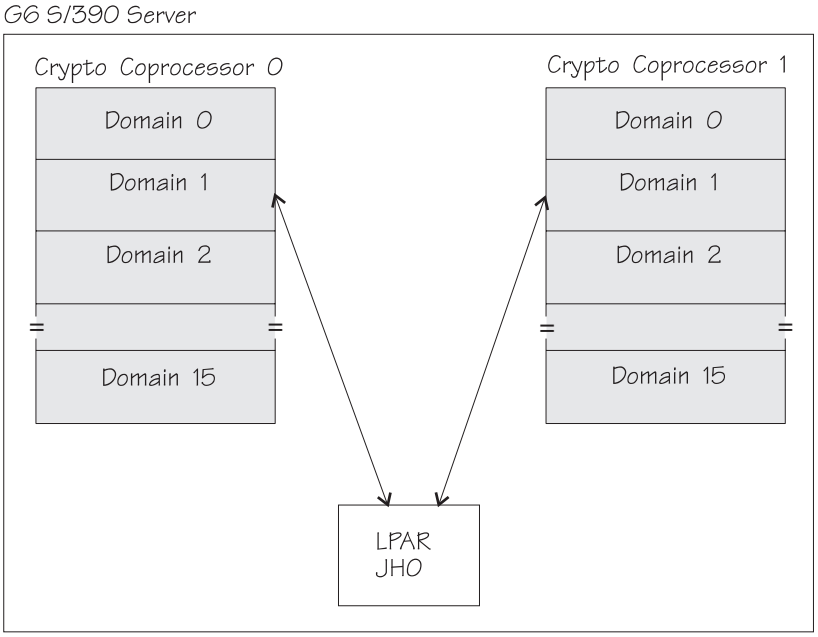


Figure 5. Our ICSF Configuration

To assign each image its crypto CP domain number, you specify it in the usage domain index on the Support Element Customize Image Profile page. See the *S/390 Support Element Operations Guide*.

Experiences Setting Up ICSF

We have the following experiences to share:

Setting Up the Cryptographic Hardware Feature: The cryptographic coprocessor hardware feature is included on your server when you order it with feature code 0800. (See *OS/390 ICSF System Programmer's Guide* for complete information on cryptographic hardware feature codes.) If you order the feature at a later time, an IBM customer engineer will install and enable this feature on your server. Once the feature is installed, there is some hardware set up for each LPAR that must be done for the cryptographic coprocessor feature on the server. This includes assigning domains for each system image using the usage domain index on the Support Element Customize Image Profile page. Information on setting up your cryptographic hardware feature is documented in the *S/390 Support Element Operations Guide*.

Defining Domain Numbers in the Installation Setup Options: To set up the ICSF software, you'll need to know the domain number that was assigned to each LPAR or system image on your machine during hardware setup. The domain number specified for the usage domain index during hardware set up must correspond to the domain number you specify with the DOMAIN(n) keyword in the installation options data set (CSVPRMxx member). You can find out what domain numbers were assigned for each partition using the hardware management console (HMC). When we set up the installation options data set for ICSF, we made sure that the domain number for each system image matched the one displayed on the HMC.

When we mistakenly specified a domain for a system image in the installation options data set that **did not** match the one displayed on the HMC (reflecting the

one specified in the usage domain index for that image), we received system abend X'18F' with a reason code of X'3C' (the specified domain index is not valid) .

Verifying The Status of the Cryptographic Hardware Feature: Before starting ICSF, our very first step was to check the status of the cryptographic hardware feature. This is important because you cannot set up ICSF unless the hardware is installed and ready to go. We verified the status of the cryptographic hardware on the HMC. For example, **before** pass phrase initialization, the status of the cryptographic coprocessor feature displayed on the HMC read:

```
Pseudo-Random number not initialized
```

This indicates that pass-phrase initialization has not completed.

Verifying Pass Phrase Initialization: We also needed to verify the status of the cryptographic hardware feature after ICSF had been started to verify that pass phrase initialization had successfully completed. There are three different ways to do this:

- Using the HMC: When we checked the status of the feature on the HMC after pass phrase initialization was complete, the status displayed for the cryptographic coprocessor was:

  ```
  Initialized
  ```

  This indicates that pass phrase initialization had completed correctly.
- Monitoring system messages: After pass phrase initialization and CKDS initialization steps of ICSF initialization, the system issues the following messages to the console:

  ```
  IEE504I CRYPTO(0),ONLINE
  CSFM400I CRYPTOGRAPHY - SERVICES ARE NOW AVAILABLE.
  IEE504I CRYPTO(1),ONLINE
  ```
- Issuing a display command: We Issued D M=CPU to display the status of the crypto coprocessor. The example below shows the output of the command after ICSF has been started. The cryptographic coprocessor feature status is displayed as ONLINE, as indicated by the plus signs (+) in the CR (crypto facility) column:

  ```
  D M=CPU
  IEE174I 13.44.06 DISPLAY M 695
  PROCESSOR STATUS
  ID  CPU  CR              SERIAL
  0    +   +               0301049672
  1    +   +               1301049672
  2    +   .               2301049672
  3    +   .               3301049672
  4    +   .               4301049672
  5    +   .               5301049672
  6    +   .               6301049672
  7    +   .               7301049672
  8    +   .               8301049672
  9    +   .               9301049672

  CPC ND = 009672.XZ7.IBM.02.000000050104
  CPC SI = 9672.XZ7.IBM.02.0000000000050104
  CPC ID = 00

  + ONLINE    - OFFLINE    . DOES NOT EXIST

  CR      CRYPTO FACILITY
  CPC ND  CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
  CPC SI  SYSTEM INFORMATION FROM STSI INSTRUCTION
  CPC ID  CENTRAL PROCESSING COMPLEX IDENTIFIER
  ******************************** BOTTOM OF DATA ***
  ```

Chapter 4. Migrating to DB2 UDB for OS/390 Version 6

We run DB2 data sharing workloads in our application group 3 across all 12 production systems in our sysplex (as described in “Our Workloads” on page 14). We run a 12-way data sharing group one day a week and a 4-way data sharing group the other six days of the week.

In this chapter, we discuss the migration of our data sharing group from DB2 Version 5 to DB2 Version 6 and highlight some of the important points to consider to ensure a smooth migration.

Planning for Migration

We can’t stress enough that the single most important thing you must do to ensure a successful migration is to ***carefully read the installation documentation and related information, and follow them to the letter.*** At a minimum, this includes the following:

- *DB2 UDB for OS/390 V6 Installation Guide*, which provides comprehensive procedures for both installing and migrating to DB2 V6. Note that you can only migrate to V6 from V5. If you currently run any version of DB2 other than V5, you must follow the *installation* procedure rather than the *migration* procedure.
- *DB2 UDB for OS/390 V6 Data Sharing: Planning and Administration*, which specifically addresses migrating a data sharing group to a new release.
- *DB2 Program Directory* that comes with your DB2 V6 software. Also, use the keyword specifications in the program directory to check for any PSP updates.
- Information APAR II11442, which addresses migration and fallback for DB2 V6, including a list of required service for toleration, coexistence, and compatibility.

You might also want to refer to *DB2 UDB for OS/390 V6 Release Planning Guide* for descriptions of the new functions and enhancements in DB2 V6.

As you develop your migration plan, you might find that you do not need to perform each and every step in the migration procedure, as some steps are optional and might not apply to your installation. Still, be sure to read and understand each step and then deliberately decide whether or not you need to take any action. You will experience problems if you make assumptions or skip over steps.

Migration Considerations and Premigration Activities

Two topics in *DB2 UDB for OS/390 V6 Installation Guide* which you’ll want to pay particular attention to as you begin planning your migration are “Migration Considerations” and “Premigration Activities”. These topics identify all of the changes and differences in DB2 V6 which might impact your migration or subsequent DB2 operations. Although, as we caution above, you must review the information in its entirety to ensure that you don’t miss anything, we wanted to highlight a few of the considerations that we found particularly applicable to our environment.

Type 2 Indexes Are Now Required: DB2 no longer supports type 1 indexes. You must convert all indexes to type 2 before migrating to V6. Catalog migration will fail if any type 1 indexes exist. The installation guide explains how to identify any type 1 indexes and how to convert them to type 2.

Data Set Passwords Are Not Supported: You must remove all data set passwords before migrating to V6. Catalog migration will fail if any data set passwords exist. Instead, use OS/390 Security Server (or an equivalent security product) to protect your data sets. The installation guide explains how to remove data set passwords by using the SQL ALTER statement.

Shared Read-Only Data Is Not Supported: You must eliminate shared read-only data. Catalog migration will fail if any shared read-only data exists. The installation guide explains how to do this. Instead of shared read-only data, consider a data sharing group as a more substantial and usable method to share data.

Stored Procedures No Longer Reside in SYSIBM.SYSPROCEDURES: Catalog table SYSIBM.SYSPROCEDURES is no longer used to define stored procedures to DB2. During catalog migration, all rows in SYSIBM.SYSPROCEDURES are automatically migrated to the new SYSIBM.SYSROUTINES catalog table and to SYSIBM.SYSPARMS. The installation guide explains how to modify the SYSPROCEDURES catalog table to match the updates to SYSROUTINES and SYSPARMS in order to provide for fallback or to prepare for release coexistence in a data sharing environment.

Temporarily Remove Views on Two Catalog Tables: You must drop all views on catalog tables SYSIBM.SYSCOLDIST and SYSIBM.SYSCOLDISTSTATS before running the catalog migration job DSNTIJTC. Catalog migration will fail if any views exist on these tables. After you successfully migrate your catalog to V6, you can recreate the views.

Host Variables Require a Preceding Colon: In application programs, all host variable references in SQL statements now must begin with a colon, as in this example:

```
EXEC SQL SELECT MAX(SALARY) INTO :MAXSALRY FROM DSN8610.EMP;
```

The host variable *:MAXSALRY* above contains the required colon. The colon was optional in previous releases, but now is required. Error DSNH315E results if any host variables lack a leading colon.

Migrating a Data Sharing Group: DB2 supports the coexistence of V5 and V6 members in a data sharing group. You must first apply the fallback SPE to all V5 members, as directed in the installation guide. You can then migrate the members to the new release, one at a time. Keep in mind, however, that new V6 functions are not available to group members that have not yet migrated. Review the information on release coexistence and migrating a data sharing group in the above publications and plan to completely migrate the group over as short a time as possible.

DB2 V6 Migration Experiences

We followed the migration procedure in *DB2 UDB for OS/390 V6 Installation Guide* and the guidelines on migrating a data sharing group in *DB2 UDB for OS/390 V6 Data Sharing: Planning and Administration*. We also have the following experiences to share:

Staging Our Migration: As we described at the beginning of this chapter, we run DB2 in both a 4-way and a 12-way data sharing environment. Therefore, we decided to migrate to DB6 V6 in two phases: first, the four members that compose our 4-way data sharing group, followed by the remaining eight members.

We began the first phase with all 12 data sharing members up. We quiesced the four members we wanted to migrate and allowed our workloads to continue running on the other eight members. We then proceeded to migrate the first of the four members to V6. Once that was complete, we continued with the next three members.

Once we successfully migrated the four members, we continued running 12-way data sharing in coexistence mode—with four members running on DB2 V6 and eight members still running DB2 V5—for one week. After successfully running for a week, we proceeded to migrate the eight remaining members.

To help position ourselves for a staged migration, we again used the dynamic LNKLST capability and created a temporary logon procedure pointing to the new V6 libraries—much the same as we did for our previous DB2 migration from V4 to V5 (see our December 1997 edition for a detailed explanation of how we do this).

Running the Installation CLIST: We ran the DB2 installation CLIST and reviewed and verified the parameter values from our previous run. For now, we simply accepted the default values for any new V6 parameters.

Minor Problems We Could Have Avoided: We only encountered a couple of fairly minor problems, but we could have completely avoided them had we followed our own advice and read the installation guide a little more carefully. The following two points are clearly documented, but we somehow overlooked them or simply lost track of them and it caused us a bit of unnecessary trouble:

- **Catalog Migration Failed the First Time.** As we highlighted in the previous section, DB2 V6 no longer supports type 1 indexes. We didn't check for the existence of any type 1 indexes and, as (bad) luck would have it, we had a few hanging around. Sure enough, when we got to the step where we ran DSNTIJTC to migrate the catalog, it failed as documented. We had to go back and identify the offending type 1 indexes and get rid of them. We converted some to type 2 indexes and dropped some others that we didn't need anymore. We then re-ran DSNTIJTC and it successfully completed.
- **RECOVER INDEX Did Not Work.** DB2 V6 now uses REBUILD INDEX to perform the function previously known as RECOVER INDEX. We tried to issue `RECOVER INDEX (ALL) TABLESPACE xxx.yyy` and it failed. In V6, we instead should have issued **`REBUILD`** `INDEX (ALL) TABLESPACE xxx.yyy` to accomplish what we were trying to do. REBUILD INDEX reconstructs indexes by scanning the table that they reference, while RECOVER INDEX (with altered syntax) restores indexes from a full image copy. Any jobs from previous DB2 releases that specify RECOVER INDEX must now specify REBUILD INDEX instead. The installation guide clearly identifies this release incompatibility and *DB2 UDB for OS/390 V6 Utility Guide and Reference* fully describes the syntax and function of these commands. We changed RECOVER to REBUILD, and it worked as expected.

Summing Up Our Migration: Overall, we're pleased with how smoothly we migrated to DB2 V6. However, we must reiterate that the key to a successful migration—no matter how many times you've done it before—is to rigorously follow the procedures in the installation documentation. As faithfully as we thought we were doing that, our experiences above illustrate how easy it is to miss something important.

Finally, we should note that while DB2 V6 includes many interesting new features, our activities to date only concern the actual migration to the V6 product. We continue to run our existing workloads and do not yet exploit any new V6 functions.

However, we have plans underway to develop workloads to exercise some of the new functions, and we hope to share those experiences with you in our future test reports.

Chapter 5. Setting Up a DB2 Large Objects (LOBS) Application With Web Access

This section describes some of our experiences in setting up a banking application that lets Web users access and update data stored in DB2 table spaces on OS/390. While we were setting up this application, we decided to take advantage of the new LOB DB2 function. LOBs are a new type of DB2 object that you can use to store large amounts of data of varying types, from large text documents to video files, in a varying-length character string. They can contain almost 2 GB of data. There are three types of LOBs: BLOBs, CLOBs and DBCLOBs. In our banking application we used only BLOBs, which store binary data such as pictures, audio recordings, and mixed media files. See *DB2 Release Guide* for information on LOBs.

In order to implement LOBs, you must have the following products:

- IBM HTTP Server V5.2. See our December '99 edition.
- Net.Data V2R2. See "Using Net.Data to Connect the IBM HTTP Server to DB2" on page 90.
- DB2 UDB V6. See "Chapter 4. Migrating to DB2 UDB for OS/390 Version 6" on page 41.

You must also have the fix for APAR PQ24314 installed on your system in order to use LOBs. In addition, note that the way we implemented our application is built on our shared HFS environment. See "Chapter 11. Managing a Hierarchical File System (HFS)" on page 109.

Our Application

The following picture shows our banking application and follows a user request for banking data through the application:

DB2 LOBS Application

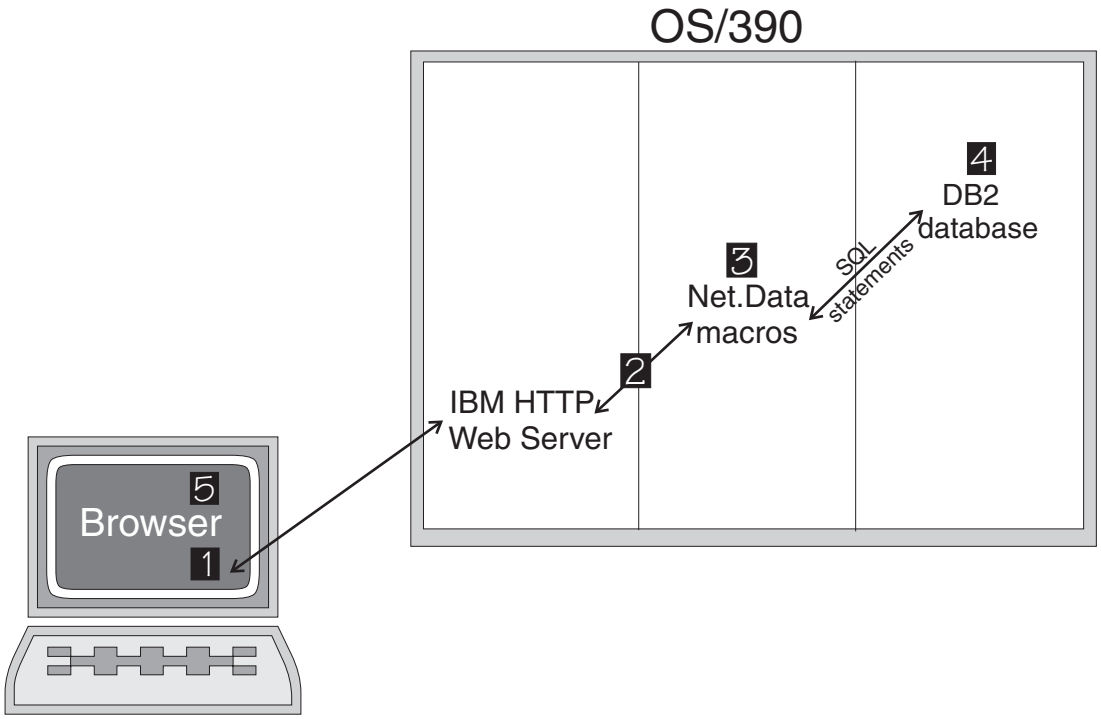


Figure 6. Our DB2 LOB Application

A user's request for their banking data from our application flows as follows:

- 1 - A user accesses the URL address for our application and through our Web pages, issues a request to get or update their bank data. To run this application, we use a product on our workstation that simulates multiple Web users accessing our application.
- 2 - The IBM HTTP Server is our Web server, the interface between the browser and OS/390. The IBM HTTP Server passes the request to Net.Data.
- 3 - We coded Net.Data macros for our application that:
  - Build the Web pages for our application.
  - Issue the SQL statements (select, insert, delete, and update) that extract and update our banking data stored in DB2 databases on OS/390.
- 4 - DB2 UDB on OS/390 contains all the data (including the LOBs) for our banking application in table spaces that are accessed by our Net.Data macros.
- 5 - The data that the user requests now flows back the same way it came, from DB2 to Net.Data. Net.Data passes the data and the appropriate Web page back to the Web user on the workstation through the IBM HTTP Server. The new Web page with the user's data is displayed on the workstation. Many of our pages also display a picture or graphic, which is LOB data.

Setting Up Our Environment for The Application

We set up our web enabled banking application as follows:

- "HFS Setup" on page 47
- "Updates to the IBM HTTP Server" on page 47
- "Updates to Net.Data" on page 51
- "Sample Net.Data Macros" on page 52
- "Updates to DB2" on page 47
- "Our Experiences In Setting Up DB2 LOBs Application" on page 55

HFS Setup

We created a new shared read-write HFS for temporary storage of the LOBs that this application handles. Net.Data stores the LOBs in this HFS after extracting them from DB2. The IBM HTTP Server retrieves the LOB from this HFS in order to return it to the user on the Web browser.

We named the data set for the HFS hlq.TMPLOB.llq and mounted it read-write at the /http_shared/netdata/tmplobs. /http_shared is the mount point for the shared customizable HFS we set up for our IBM HTTP Server (see “Migrating to Shared HFS for the IBM HTTP Server” on page 91). We set the permission bits for the files in this HFS to 777, to give universal write access.

Because the LOBs are large (they range in size from 2K to 16Mb), we wanted to make sure that we don’t fill up this HFS with our LOBs. To ensure that we don’t fill our HFS, we configured Net.Data to use dynamic Web page caching so that Net.Data deletes the LOBs from the HFS after ten seconds. See “Updates to Net.Data” on page 51 for how we did this.

Updates to the IBM HTTP Server

The IBM HTTP Server passes both the user request to OS/390 and the response back to the workstation. Because we use Net.Data macros for other applications, we had already defined the paths in our http.conf file for the IBM HTTP Server to access the Net.Data macros:

```
PASS /netdata-cgi/*  /http_shared/netdata/macros/*
```

We had to add a Pass directive to enable our IBM HTTP Server to access the new HFS /tmplobs directory. We added the following Pass directive to our http.conf file:

```
PASS /tmplobs/*  /http_shared/netdata/tmplobs/*
```

Updates to DB2

We used the following books to make our DB2 updates for our banking application:

- *DB2 UDB for OS/390 Release Planning Guide*
- *DB2 UDB for OS/390 Administration Guide*
- *DB2 UDB for OS/390 Command Reference*
- *DB2 UDB for OS/390 SQL Reference*

This section covers the following setup steps we performed for DB2 for our LOBs application. Most of the changes we made were to enable the use of LOBs:

- “Creating New Buffer Pools and Group Buffer Pools”
- “Setting Up DB2 Objects” on page 48
- “Putting Our LOB Data Into DB2” on page 50
- “Service for DB2 to Enable LOBs” on page 50

Creating New Buffer Pools and Group Buffer Pools

We created three new buffer pools and corresponding group buffer pools for our banking application. As recommended by the *DB2 Release Guide*, we created a separate buffer pool/group buffer pool for our LOB data. We named them as follows:

- buffer pool BP2, group buffer pool GBP2 for our table spaces.
- buffer pool BP9, group buffer pool GBP9 for our table space indexes.
- buffer pool BP32K, group buffer pool GBP32K for our LOB tablespace.

We defined the following ALTER BUFFERSPACE command values for our buffer pools:

Table 8. Our LOB Buffer Pool Values

Buffer Pool	VP Size	HP Size	Cast Out	VP SEQT	VP PSEQT	HP SEQT	DWQT	VDWQT
BP2	2000	0	Yes	80	50	80	50	10,0
BP9	2000	0	Yes	80	50	80	50	10,0
BP32K	2000	0	Yes	80	50	80	50	10,0

For the meaning of each of the values, see the *DB2 Utility Guide and Reference*.

We defined the following coupling facility cache structures for our group buffer pools in the CFRM policy:

```
STRUCTURE NAME(DSNDB1G_GBP2)
          SIZE(50000)
          INITSIZE(39600)
          DUPLEX(ENABLED)
          PREFLIST(CF2,CF1)
          REBUILDPERCENT(1)
          FULLTHRESHOLD(90)

STRUCTURE NAME(DSNDB1G_GBP9)
          SIZE(50000)
          INITSIZE(39600)
          DUPLEX(ENABLED)
          PREFLIST(CF2,CF1)
          REBUILDPERCENT(1)
          FULLTHRESHOLD(90)

STRUCTURE NAME(DSNDB1G_GBP32K)
          SIZE(50000)
          INITSIZE(20000)
          DUPLEX(ENABLED)
          PREFLIST(CF2,CF1)
          REBUILDPERCENT(1)
          FULLTHRESHOLD(90)
```

Setting Up DB2 Objects

Our LOBs application required a number of DB2 objects, such as a database, table spaces, tables, and indexes. In addition, for our LOB table space, LOBPHOTO, we defined an auxiliary table. (LOBs must be stored in an auxiliary table rather than a table.) We defined one database, LOBDBTST, and one storage group, LOBSGTST, for our application. The following table shows the other objects we defined and how they relate to each other:

Table 9. DB2 Objects for our LOBs Application

Tablespace Name	Table Name	Auxiliary Table	Index Name
LOBTSTST	LOBS.LOGON.TAB		LOBS.XUSERID
LOBTSTST	LOBS.OP_BAL		LOBS.XCUSTNBR
LOBTSTST	LOBS.TRAN		LOBS.XTRANNO
LOBPHOTO (type LS)		LOBS.CUST_PHOTO_TAB	LOBS.XCUST_PHOTO
LOBTSTST	LOBS.OP_BAL1		LOBS.XCUSTNBR1

To create these DB2 objects for our LOBs application, we used the following SQL statements:

```
CREATE DATABASE LOBDBTST
       STOGROUP LOBSGTST
     BUFFERPOOL BP2
        INDEXBP BP9;
```

```
CREATE TABLESPACE LOBTSTST
              IN LOBDBTST
   USING STOGROUP LOBSGTST
           PRIQTY 80000
           SECQTY 40000
            ERASE NO
    LOCKSIZE PAGE LOCKMAX SYSTEM
       BUFFERPOOL BP2
         GBPCACHE NONE
           CLOSE NO;
CREATE TABLE LOBS.LOGON_TAB (USERID CHAR(8) NOT NULL,
                         PASSWORD CHAR(10),
                       PRIMARY KEY (USERID))
              IN LOBDBTST.LOBTSTST;

CREATE UNIQUE INDEX LOBS.XUSERID
                 ON LOBS.LOGON_TAB
            (USERID ASC)
     USING STOGROUP LOBSGTST
             PRIQTY 12
              ERASE NO
          BUFFERPOOL BP9
            GBPCACHE NONE
               CLOSE NO;
              (CUST_NBR INT  NOT NULL,
               OPEN_AMT DECIMAL(18,2) CONSTRAINT
                     OP_AMT_CONST CHECK(OPEN_AMT >= 500.00),
               CURRENT_BAL DEC(18,2) CONSTRAINT
                     CURRENT_BAL_CONST CHECK(CURRENT_BAL >= 500.00),
               ROW_ID      ROWID GENERATED ALWAYS,
               CUST_PHOTO  BLOB(100M),
               PRIMARY KEY (CUST_NBR))
            IN LOBDBTST.LOBTSTST;
CREATE UNIQUE INDEX LOBS.XCUSTNBR
                 ON LOBS.OP_BAL
           (CUST_NBR ASC)
     USING STOGROUP LOBSGTST
             PRIQTY 12
              ERASE NO
          BUFFERPOOL BP9
            GBPCACHE NONE
               CLOSE NO;
CREATE TABLE LOBS.TRAN
                (TRAN_CODE  CHAR(02),
                   TRAN_NO  SMALLINT,
                  CUST_NBR  INT NOT NULL,
                  TRAN_DATE  DATE,
               TRAN_AMOUNT  DECIMAL(18,2) CONSTRAINT
            TRAN_AMT_CONST  CHECK (TRAN_AMOUNT > 0),
             FOREIGN KEY    RTO(CUST_NBR) REFERENCES LOBS.OP_BAL)
       IN LOBDBTST.LOBTSTST;
CREATE UNIQUE INDEX LOBS.XTRANNO
                 ON LOBS.TRAN
            (TRAN_NO ASC)
     USING STOGROUP LOBSGTST
             PRIQTY 12
              ERASE NO
          BUFFERPOOL BP9
            GBPCACHE NONE
               CLOSE NO;

CREATE LOB TABLESPACE LOBPHOTO
                   IN LOBDBTST
       USING STOGROUP LOBSGTST
                PRIQTY 80000
                SECQTY 40000
                BUFFERPOOL BP32K
                   LOG NO;

CREATE AUXILIARY TABLE LOBS.CUST_PHOTO_TAB
                    IN LOBDBTST.LOBPHOTO
                 STORES LOBS.OP_BAL
                 COLUMN CUST_PHOTO;
```

```
CREATE TYPE 2 UNIQUE INDEX LOBS.XCUST_PHOTO
                       ON LOBS.CUST_PHOTO_TAB
                  GBPCACHE NONE;
```

Note that our LOB table space, LOBPHOTO, is not partitioned, so we only need one auxiliary table for our LOBs. If you have multiple partitions in a LOB table space you must have one auxiliary table for each partition. For complete information on defining auxiliary tables, see the introduction to LOBs in the *DB2 Application Programming and SQL Guide*.

Putting Our LOB Data Into DB2

The LOBs for our banking application consist of .gif and .jpeg graphic files. We've already defined the table space (LOBPHOTO) and auxiliary table for our LOBs. Now we have to put our LOB data into DB2. Because we have more than 32KB of LOB columns, we had to do this in two phases, using both the LOAD utility and the INSERT function. The LOAD utility can only load LOBs if the total size of all columns is under 32 KB, (including 4 bytes for a length attribute).

1. We load the first 32KB of LOB columns for our .gif and .jpeg files using the LOAD utility as follows:

   ```
   LOAD DATA INDDN(LOADINDD)
         RESUME(YES) LOG(NO)
       INTO TABLE LOBS.OP_BAL
       (CUST_NBR    POSITION (1) INTEGER,
         OPEN_AMT    POSITION  (05) DECIMAL,
         CURRENT_BAL POSITION   (15) DECIMAL,
           CUST_PHOTO POSITION(25) BLOB)
   ```

   Note that although the LOB data is actually stored in the auxiliary table, the target of our LOAD utility statement specifies the name of the base table for the auxiliary table, LOBS.OP_BAL (see the CREATE LOB TABLESPACE statement in "Setting Up DB2 Objects" on page 48).

2. To load the rest of our LOB columns after the first 32KBs, we wrote a COBOL program that inserted the files into DB2. We needed this program to insert the files because the DB2 base table row size is restricted to 32K. See *DB2 Release Guide*, in the section on object-relational extensions and active data for more information.

   Our COBOL program issues the following SQL statement to insert the LOB data:

   ```
   EXEC    SQL
    INSERT  INTO    LOBS.OP_BAL
    (CUST_NBR,OPEN_AMT,CURRENT_BAL,ROW_ID,CUST_PHOTO)
      VALUES(:WS-CUST-NO,
             :WS-OPEN-AMT,
             :WS-CURR-BAL,
             DEFAULT,
             :WS-PHOTO)
   END-EXEC
   ```

   Again, note that although the LOB data is actually stored in the auxiliary table, the target of our INSERT statement specifies the name of the base table for the auxiliary table, LOBS.OP_BAL .

Service for DB2 to Enable LOBs

In order to enable LOBs, we needed the fixes for the following APARs while testing our DB2 LOBs application:

- APAR PQ36964–We encountered the following S0C4 abend code while issuing select SQL statements against the LOB base table (auxiliary table LOBS.CUST_PHOTO_TAB) through SPUFI:

```
ABEND0C4-10 IN DSNICUBD+0270
```

Installing the fix for APAR PQ36964 resolves this problem.

- APAR PW36953–We received the following error message while we were running the CHECK LOB utility:

```
DSNU787I   DSNUK001 - LOB WITH
                ROWID=X'00756B457C1CD6F4840901910642'
                      VERSION=X'0001' PAGE=X'0000BE43'
                      IS IN ERROR FOR REASON=X'07'
```

Installing the fix for APAR PQ36953 resolves this problem.

Updates to Net.Data

For our banking application, we created Net.Data macros in our shared HFS on OS/390. These Net.Data macros were coded to:

- Build the Web pages for our application.
- Issue the SQL statements (select, insert, delete, and update) that extract and update our banking data stored in DB2 table spaces on OS/390.

See “Sample Net.Data Macros” on page 52.

To make our updates to our Net.Data configuration and write our Net.Data macros, we used the *Net.Data Administration and Programing Guide for OS/390* which is available at the following URL:

```
http://www.ibm.com/software/data/net.data/library.html
```

We made the following updates to our Net.Data configuration for our DB2 LOBs application:

Ensuring that Net.Data Macros Are Processed as ICAPIs: We ensured that our Net.Data macros were processed as ICAPIs rather than CGIs, because LOBs are not supported with CGIs.

Updates to the Net.Data Initialization File: We made changes to our Net.Data initialization file, db2www.ini, to support our LOBs application, including:

- ***** - We had already defined the path to our Net.Data macros in the shared IBM HTTP Server HFS because we use Net.Data macros for other applications. This path statement lets Net.Data know where our macros are stored.
- **1** - Adding the path for temporary storage for the LOBs in the /tmplobs directory.
- **2** - Adding a statement to enable Net.Data dynamic caching.
- The next two statements enable Net.Data to automatically manage cached Web pages and LOBs. Statement **3** specifies that Net.Data delete cached Web pages every 20 seconds. Statement **4** specifies that Net.Data automatically delete LOBs after 10 seconds.

Our db2www.ini file looks as follows:

```
%{ Net.Data 2.2 ini file    %}
* MACRO_PATH /usr/lpp/netdata/macros;/http_shared/netdata/macros;
EXEC_PATH /usr/lpp/netdata/testcmd;
FFI_PATH /usr/lpp/netdata/file-data;
%{This is the path where temporary LOBS will reside%}
1 HTML_PATH /http_shared/netdata/tmplobs;
DB2SSID DB1G
DB2PLAN DTWGAV22
DTW_SHOWSQL YES
DTW_DIRECT_REQUEST YES
%{Configured NET.DATA for Dynamic Web Page Caching%}
```

```
 2 DTW_CACHE_PAGE /* 20 PUBLIC
%{Added these statements for NET.DATA to automatically manage cached Web pages and LOBS%}
 3 DTW_CACHE_MANAGEMENT_INTERVAL 20
%{ Delete LOBS in 10 seconds %}
 4 DTW_LOB_LIFETIME 10
%{ Tracing and logging data       %}
%{DTW_TRACE_LOG_DIR /dgw_pet/netdata/logs
DTW_TRACE_LOG_LEVEL SERVICE %}
DTW_ERROR_LOG_DIR /dgw_pet/netdata/logs
DTW_ERROR_LOG_LEVEL ALL
ENVIRONMENT (DTW_SQL) dtwsqlv6 (IN LOCATION)
ENVIRONMENT (DTW_SYSTEM) sysdll ()
ENVIRONMENT (DTW_PERL) perldll ()
ENVIRONMENT (DTW_REXX) rexxdll ()
ENVIRONMENT (DTW_FILE) filedll ()
```

Sample Net.Data Macros

The following samples show how we might code Net.Data macros to build a Web page and issue SQL statements to access or update our application data in DB2:

Sample Net.Data Macro that Verifies USERID and Password: The following macro displays our LOGON page which elicits the user's USERID and password and then checks for a match in a DB2 LOB table space on OS/390:

```
%{/* <P>       This file is <I>$(DTW_MACRO_FILENAME)</>,
/*               and was updated on $(DTW_MACRO_LAST_MODIFIED)
/* <P> The Net.Data executable file is $(DTW_MP_PATH)
/* <P> This Web application uses $(DTW_MP_VERSION)
/*    SQLSTATE = <STRONG>$(SQL_STATE)</STRONG>
/*    0  : "Everything Went OK"
/*<P>SERVER  IS $(SERVER_NAME)  %}
%{*****************************Comment Section***********************/
/* FileName: ban00.d2w                                               */
/* Description: This Net.Macro file checks the userid and password in */
/*              the logon table.                                     */
/* SQL Function block included in this file is:                      */
/*    formquery0: Selects userid column from the database.           */
/* HTML blocks included in this file are:                            */
/*         INPUT                                                     */
/*         REPORT                                                    */
/*                                                                   */
/*********************************************************************%}
%{***************************** Define Section  **********************%}
%DEFINE {
MACRO_NAME="BAN00.D2W"
DTW_HTML_TABLE="YES"
LOBTAB="LOBS.LOGON_TAB"
dtw_default_mreport="YES"
DTW_SET_TOTAL_ROWS="NO"
col1=""
SERVER_NAME=%ENVVAR
%}
%{********************** Function Definition Section ******************%}
%FUNCTION(DTW_SQL) formquery0() {
     SELECT  USERID,PASSWORD FROM $(LOBTAB) WHERE USERID='$(userid)'
                            and PASSWORD='$(passwd)'
%report {
  <P>
 %row{ <B><CENTER>HELLO WELCOME <BLINK>$(V1)</BLINK></CENTER></B> %}
         <P>
 <a href="http://ip_addr/netdata-cgi/db2www/ban01.d2w/input">
                <strong>Customer Number Check</strong></a>
<br> <p>
 <a href="http://ip_addr/netdata-cgi/db2www/out.d2w/input">
                <strong>Good Bye</strong></a>
<br> <P>
 <a href="http://ip_addr/netdata-cgi/db2www/help.d2w/input">
                <strong>HELP</strong></a>
<P>
SQLSTATE = <STRONG>$(SQL_STATE)</STRONG>
        %}
         %MESSAGE {
            0  : "Everything Went Okay!"
         -204 : "Error -204: Table not found"
         -104 : "Error -104: Syntax error"
          100 : "Logon Invalid - Logon again"
```

```
       +default: "Warning $(RETURN_CODE)" : continue
       -default: "Unexpected SQL error $(RETURN_CODE)" : exit
        %}
%}
%{************************ HTML Input Section *************************%}
%HTML (input) {
<HTML>
<head>
%{<META HTTP-EQUIV=Refresh CONTENT='30; URL=/netdata-cgi/db2www/ban00.d2w/input'> %}
<META HTTP-EQUIV=Refresh>
<image src=http://ip_addr/images/powergif.gif>
<image src=http://ip_addr/images/b6zhmast.gif>
<img src=http://ip_addr/images/ball.gif>
<title>XYZ BANK</title>
</head>
</center>
  <body bgcolor=#00000 TEXT=#00FF00>
 %{<BODY BGCOLOR=#FFFFFF BACKGROUND=http://ip_addr/images/bghome.gif> %}
 <center>
<h8><strong>BAON AUTOMATION</TEXT></strong>
<br>
 <strong><MARQUEE><FONT COLOR=RED>TEAM - John Doe, Jane Doe,Joe Shmoe</font></MARQUEE></strong>
<p>
 Use Of This Application Is For
<br>
IBM Management Approved Purposes Only
<br>
</h5>
<font size=100><strong>IBM</strong></font>
          </center>
<p>
DATE : @DTW_rDATE("U") <BR>
TIME : <BLINK>@DTW_rTIME("L")</BLINK>
 <p>This Web application uses<strong>$(DTW_MP_VERSION)</strong> from the server
<strong>$(SERVER_NAME)</strong>
<strong><hr size=12></strong>
<img src=http://ip_addr/images/colorbar.gif>
<form method="post" action="report">
   <p>
<pre>
 Fill in your USERID and PASSWORD (Note:Upper Case only)
 <strong>
 USERID<img src=http://ip_addr/images/hand.gif> : <input type="text" name="userid" size=8 maxlength=8>
 PASSWORD<img src=http://ip_addr/images/hand.gif>: <input type="password" name="passwd" size=10 maxlength=10>
 </strong>
 <p>
 <input type="submit" Value="Submit Query">
  </pre>
   %{<p>
     SHOWSQL: <INPUT TYPE="radio" Name="SHOWSQL" VALUE="YES"> Yes
              <INPUT TYPE="radio" Name="SHOWSQL" VALUE="" CHECKED> No %}
 </form>
<br> <P>
 <a href="http://ip_addr/netdata-cgi/db2www/help.d2w/input">
               <strong>HELP</strong></a>
 <br>
<br> <p>
 <a href="http://ip_addr/netdata-cgi/db2www/out.d2w/input">
               <strong>Good Bye</strong></a>
  <br>
 Greetings from<br>
 <a href="http://w3.s390.ibm.com/solution/b6zh/b6zhpict.htm">IBM CORP</a>
 <cite>DB2 V6 MIGRATION TEAM</cite>
 <br>&copy;IBM CORP
 <P> This file is available in <I>$(DTW_CURRENT_FILENAME)</>,
               and was updated on $(DTW_CURRENT_LAST_MODIFIED)
 <P>
 </body>
 </HTML>
 %}
 %{************************ HTML Report Section ************************%}
 %HTML(report){
 <HTML>
 <TITLE>Macro $(MACRO_NAME) Results</TITLE>
 <META HTTP-EQUIV=Refresh>
 %{body bgcolor="00FF00"  text="0000FF"> %}
 <BODY BGCOLOR=#FFFFFF BACKGROUND=http://ip_addr/bghome.gif>
 <image src=http://ip_addr/images/b6zhmast.gif>
 <h1 align=center> Report Section</h1>
 DATE : @DTW_rDATE("U") <BR>
 TIME :<BLINK> @DTW_rTIME("L")</BLINK>
 <hr size=12>
 <br>
  @formquery0()
```

```
<br> <p>
</body>
</HTML>
%}
```

Sample Net.Data Macro That Displays Customer Information: After the user's USERID and password have been verified, the following macro elicits a customer number, then retrieves information about that customer, such as current balance and a customer photo, from a DB2 LOB table space on OS/390. The information is disp layed on a web page:

```
%{*****************************COMMENT SECTION***********************/
/* FILENAME: BAN01.D2W                                                */
/* DESCRIPTION: THIS NET.MACRO file selects Customer number, Open     */
/*              opening balance, Current balance, Customer Photo      */
/*              From Op_bal Table
/* SQL FUNCTION BLOCK INCLUDED IN THIS FILE IS:                       */
/*   FORMQUERY01                                                      */
/* HTML BLOCKS INCLUDED IN THIS FILE ARE:                             */
/*   INPUT                                                            */
/*   REPORT                                                           */
/*                                                                    */
/**********************************************************************%}
%{***************************** DEFINE SECTION  ***********************%}
%DEFINE {
MACRO_NAME="BAN01.D2W"
DTW_HTML_TABLE="YES"
LOBTAB="LOBS.OP_BAL"
DTW_DEFAULT_REPORT="YES"
DTW_SET_TOTAL_ROWS="YES"
DTW_SET_NUM_ROWS="YES"
%}
%{********************** FUNCTION DEFINITION SECTION ******************%}
%FUNCTION(DTW_SQL) FORMQUERY01() {
  SELECT CUST_NBR,OPEN_AMT,CURRENT_BAL,CUST_PHOTO FROM $(LOBTAB)
    WHERE CUST_NBR = $(custno)
%REPORT {
 <TABLE BORDER=3 CELLPADDING=4 ALIGN=CENTER
  BGCOLOR=FUSCHIA
  BORDERCOLORLIGHT=LIME
  BORDERCOLORDARK=GREEN
  BORDERCOLOR=RED>
<TR><TH>$(N1)</TH><TH>$(N2)</TH><TH>$(N3)</TH>
 %ROW{
 <TR><TD>$(V1)</TD><TD>$(V2)</TD><TD>$(V3)</TD>
%}
   <IMG ALIGN="RIGHT" SRC="$(V4)" BORDER=5 HSPACE=20>
   <P>
%{<IMG ALIGN="RIGHT" SRC="HTTP://ip_addr/tmplobs/jerry" BORDER=5H SPACE=20>
   <IMG SRC="$(V4)" BORDER=5 HSPACE=20 ALIGN=TEXTTOP HEIGHT=200
   WIDTH=190>  PICT 2
   <IMG SRC="$(V4)" BORDER=5 HSPACE=20 ALIGN=BASELINE HEIGHT=200
   WIDTH=190> PICT 3  %}
 </TABLE>
<P>
 <strong> CLICK - TRANSACTION OR CURRENT BALANCE </STRONG>
<P>
 <P>
<A HREF="HTTP://ip_addr/netdata-cgi/db2www/ban01.d2w/input">
                          <STRONG>BACK</STRONG></A>
 <P>
<A HREF="HTTP://ip_addr/netdata-cgi/db2www/ban04.d2w/input?custno=$(custno)">
                          <STRONG>TRANSACTION</STRONG></A>
<P>
<A HREF="HTTP://ip_addr/netdata-cgi/db2www/ban05.d2w/input?custno=$(custno)">
                         <STRONG>CURRENT BALANCE</STRONG></A>
<P>
 <a href="http://ip_addr/netdata-cgi/db2www/out.d2w/input">
                <strong>Good Bye</strong></a>
<BR> <P>
 SQLSTATE = <STRONG>$(SQL_STATE)</STRONG>
 %}
 %MESSAGE {
           0  : "Everything Went Okay!"
```

```
         -204 : " ERROR -204: TABLE NOT FOUND"
         -104 : "ERROR -104: Sql syntex error"
          100 : "PLEASE OPEN AN ACCOUNT-OPENING BALANCE"
      +DEFAULT: "WARNING $(RETURN_CODE)" : CONTINUE
      -DEFAULT: "UNEXPECTED SQL ERROR $(RETURN_CODE)" : EXIT
%}
%}
%{************************ HTML INPUT SECTION ************************%}
%HTML (INPUT) {
<HTML>
<HEAD>
<META HTTP-EQUIV=Refresh>
%{<META HTTP-EQUIV=Refresh CONTENT=';URL=/netdata-cgi/db2www/ban00.d2w/input'> %}
<TITLE>XYZ BANK - CUSTOMER NUMBER CHECK </TITLE>
</HEAD>
<body bgcolor=#00000  TEXT=#00FF00>
<CENTER>
<IMAGE SRC=http://ip_addr/images/powergif.gif>
<IMAGE SRC=http://ip_addr/images/b6zhmast.gif>
<IMG SRC=http://ip_addr/images/ball.gif>
<H2>XYZ BANK - CUSTOMER NUMBER CHECK</H2></CENTER>
DATE : @DTW_RDATE("U") <BR>
TIME : <BLINK> @DTW_RTIME("L") </BLINK>
<HR SIZE=12>
<FORM METHOD="POST" ACTION="REPORT">
<PRE>
    CUSTOMER NUMBER: <INPUT TYPE=TEXT NAME=custno SIZE=7 MAXLENGTH=7>
     <P>
    <INPUT TYPE="SUBMIT" VALUE="SUBMIT QUERY">
     <P>
    </PRE>
</FORM>
<P>
 <a href="http://ip_addr/netdata-cgi/db2www/out.d2w/input">
               <strong>Good Bye</strong></a>
<P>
GREETINGS FROM<BR>
<A HREF="http://w3.s390.ibm.com/solution/b6zh/b6zhpict.htm">IBM CORP</A>
<CITE>DB2 V6 MIGRATION TEAM</CITE>
<BR>&copy;IBM CORP
</BODY>
</HTML>
%}
%{************************ HTML REPORT SECTIO N************************%}
%HTML (REPORT){
<HTML>
<TITLE>MACRO BAN01.D2W RESULTS</TITLE>
<META HTTP-EQUIV=Refresh>
<BODY BGCOLOR=#FFFFFF BACKGROUND=http://ip_addr/images/bghome.gif>
<IMAGE SRC=http://ip_addr/images/b6zhmast.gif>
<H1 ALIGN=CENTER> REPORT SECTION</H1>
DATE : @DTW_RDATE("U") <BR>
TIME : <BLINK> @DTW_RTIME("L") </BLINK>
<HR SIZE=12>
<BR>
<A HREF="HTTP://ip_addr/netdata-cgi/db2www/ban02.d2w/input?custno=$(custno)">
                                                    OPENING BALANCE</A>
<BR>
<A HREF="HTTP://ip_addr/netdata-cgi/db2www/ban01.d2w/input">
                           <STRONG>BACK</STRONG></A>
<P>
@FORMQUERY01()
</BODY>
</HTML>
%}
```

Our Experiences In Setting Up DB2 LOBs Application

We have the following experiences to share about running our DB2 LOB application:

Web Traffic Express (WTE) Caching Clashed With server-side include (SSI) Caching: We had the WTE caching directive set on, which kept the SSIs within our

LOB application from being cached. This caused the IBM HTTP Server to come down. We had to turn WTE caching off in our http.conf file.

IBM HTTP Server Logging Filled Up our Logging HFS: While we were running with our application in production mode, we left tracing on, which then filled up our logging HFS, bringing the IBM HTTP Server down. To prevent this problem, you should turn tracing off in the Net.Data initialization file (db2www.ini) and limit logging while running in production mode.

Chapter 6. Setting Up MQSeries

IBM MQSeries is a family of products for cross-platform communication, using messages and queues. Using MQSeries products, programs can talk to each other across a network of unlike components, including processors, operating systems, subsystems, and communication protocols, using a simple and consistent application programming interface. This is all explained in *MQSeries: An Introduction to Messaging and Queuing*.

We installed MQSeries for OS/390 V2R1M0 to verify that it works in our environment. Our future goal is to develop MQSeries client/server workloads that cross platform and sysplex boundaries.

We used the following IBM MQSeries publications to help us with setting up and using MQSeries:

- *MQSeries Intercommunication*
- *MQSeries for OS/390 V2R1 System Management Guide*
- *MQSeries Application Programming Guide*
- *MQSeries Application Programming Reference*
- *MQSeries Command Reference*
- *MQSeries for OS/390 V2R1 Messages and Codes*

We installed MQSeries for OS/390 V2R1M0 on all the production systems in our sysplex. Each system has a queue manager, which provides the messaging and queuing services to applications, through a message queue interface (MQI) program call. We also have MQSeries 5.1 installed on Windows NT and Windows 95 workstations.

See also “Enabling SDSF Sysplex Support” on page 32 for MQSeries updates we made in our environment for SDSF.

Testing the MQSeries-CICS Bridge

After installing MQSeries, we tested the MQSeries-CICS bridge in our environment. The MQSeries-CICS bridge enables an application, running outside of CICS, to run a program or transaction on CICS and get a response back. This non-CICS application can be run from any environment that has access to an MQSeries network that encompasses MQSeries for OS/390.

Right now, because we’re just testing the bridge, we’re running our non-CICS application from an OS/390 system. In the future, we’ll use the MQSeries-CICS bridge to kick off a CICS transaction or program from a workstation, Web, or other client application.

To setup our MQSeries-CICS bridge, we used the *MQSeries System Administration* book. We adapted an application that used the MQSeries-CICS bridge written by another test group to run in our environment. For details on the application, see the PDF called *Online Catalog to Enterprise Data in a Secure Environment* at the following Web address:

`http://www.ibm.com/software/ebusiness/cpit2.html`

We used the information in steps 5 and 6 of Section Five of this PDF to customize the OS/390 application for the MQSeries-CICS bridge.

MQSeries

The following picture shows how our MQSeries-CICS bridge application works:

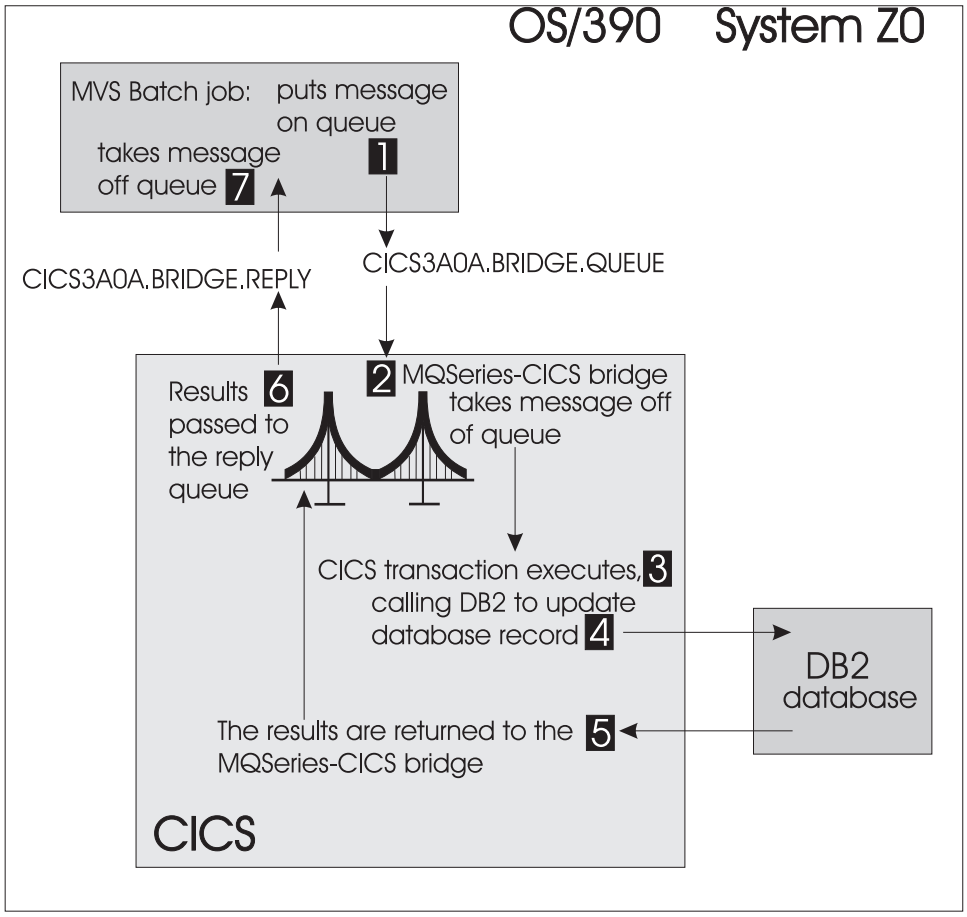


Figure 7. Our MQSeries-CICS Bridge Application

- 1 — In this step, our MVS batch job puts a message on the CICS bridge request queue, CICS3A0A.BRIDGE.QUEUE.
- 2 — The MQSeries-CICS bridge takes the message off the queue. Contained in this message is both the name of the CICS transaction and the parameters for the transaction.
- 3 — The CICS transaction executes.
- 4 — The CICS transaction calls DB2 to update a database record.
- 5 — The results from DB2 are returned to the MQSeries-CICS bridge and, 6 , passed to the CICS bridge reply queue, CICS3A0A.BRIDGE.REPLY.
- 7 — Finally, the MVS batch job retrieves the results from the CICS bridge reply queue.

We specified the MQSeries definitions for our MQSeries-CICS bridge in an initialization data set identified by the CSQINP2 DDname in the MQSeries subsystem started task procedure. In the example below, we show our MQSeries definitions for the MQSeries-CICS bridge:

```
*********************************************
* Common definitions for CICS Bridge/Adapter         *
*********************************************
 DEFINE PROCESS('CICS_BRIDGE') REPLACE +
        DESCR('CICS BRIDGE MONITOR') +
        APPLICID('CKBR') +
        APPLTYPE(CICS) +
```

```
|                          USERDATA('AUTH=LOCAL,WAIT=20')
|                        *
|
|                        ***********************************************
|                        * Definitions for CICS Bridge/Adapter         *
|                        ***********************************************
|                        * This queue name should match the queue name in the
|                        * CICS system initialization table or the SYSIN override in the
|                        *
|                        *  INITPARM=(CSQCPARM='IQ=CICS3A0A.INITQ, ...
|                        *
|                        * statement.
|                        *
|                        ******
|                        DEFINE QLOCAL( 'CICS3A0A.INITQ' ) +
|                                 DESCR( 'CKTI initiation queue' ) +
|                                 PUT( ENABLED ) +
|                                 DEFPRTY( 5 ) +
|                                 DEFPSIST( YES ) +
|                                 CLUSTER( ' ' ) CLUSNL( ' ' ) DEFBIND( OPEN ) +
|                                 GET( ENABLED ) +
|                                 SHARE +
|                                 DEFSOPT( EXCL ) +
|                                 MSGDLVSQ( FIFO ) +
|                                 RETINTVL( 999999999 ) +
|                                 MAXDEPTH( 100 ) +
|                                 MAXMSGL( 1000 ) +
|                                 NOHARDENBO +
|                                 BOTHRESH( 0 ) +
|                                 BOQNAME( ' ' ) +
|                                 STGCLASS( 'SYSTEM' ) +
|                                 USAGE( NORMAL ) +
|                         INDXTYPE( NONE ) +
|                         QDPMAXEV( ENABLED ) +
|                         QDPHIEV( DISABLED ) +
|                         QDEPTHHI( 80 ) +
|                         QDPLOEV( DISABLED ) +
|                         QDEPTHLO( 40 ) +
|                         QSVCIEV( NONE ) +
|                         QSVCINT( 999999999 ) +
|                         NOTRIGGER +
|                         TRIGTYPE( NONE ) +
|                         TRIGMPRI( 0 ) +
|                         TRIGDPTH( 1 ) +
|                         TRIGDATA( ' ' ) +
|                         PROCESS( ' ' ) +
|                         INITQ( ' ' )
|
|                        DEFINE QLOCAL( 'CICS3A0A.BRIDGE.REPLY' ) REPLACE +
|                                 DESCR( 'MQ reply to queue' ) +
|                                 PUT( ENABLED ) +
|                                 DEFPRTY( 0 ) +
|                                 DEFPSIST( YES ) +
|                                 CLUSTER(CLUSTER) +
|                                 GET( ENABLED ) +
|                                 NOSHARE +
|                                 DEFSOPT( EXCL ) +
|                                 MSGDLVSQ( PRIORITY ) +
|                                 RETINTVL( 999999999 ) +
|                                 MAXDEPTH( 999999999 ) +
|                                 MAXMSGL( 4194304 ) +
|                                 NOHARDENBO +
|                                 BOTHRESH( 0 ) +
|                                 BOQNAME( ' ' ) +
|                                 STGCLASS( 'DEFAULT' ) +
|                                 USAGE( NORMAL ) +
|                                QDPMAXEV( ENABLED ) +
```

```
        QDPHIEV( DISABLED ) +
        QDEPTHHI( 80 ) +
        QDPLOEV( DISABLED ) +
        QDEPTHLO( 40 ) +
        QSVCIEV( NONE ) +
        QSVCINT( 999999999 ) +
        NOTRIGGER +
        TRIGTYPE( FIRST ) +
        TRIGDPTH( 1 ) +
        TRIGMPRI( 0 ) +
        TRIGDATA( ' ' ) +
        PROCESS( ' ' ) +
        INITQ( ' ' )

************************************************************
***** CICS BRIDGE DEFINITIONS                       *****
************************************************************
 DEFINE QLOCAL('CICS3A0A.BRIDGE.QUEUE') REPLACE +
        DESCR('CICS BRIDGE REQUEST QUEUE') +
         SHARE +
         MSGDLVSQ(FIFO) +
         CLUSTER(CLUSTER) +
         DEFPSIST(YES) +
         HARDENBO +
         TRIGGER TRIGTYPE(FIRST) +
         PROCESS('CICS_BRIDGE') +
         INITQ('CICS3A0A.INITQ')
```

You can use the samples shipped with MQSeries in data set *hlq*.SCSQPROC as a basis for your own definitions for MQSeries. We used the following *hlq*.SCSQPROC data set members:

- CSQ4CKBM contains the MQSeries-CICS bridge samples.
- CSQ4INYG contains sample definitions for customizing your own objects, including a sample initiation queue, CICS01.INITQ definitions.

Part 2. Networking and Application Enablement

Chapter 7. Understanding Our Networking and Application Enablement Environment 63
Our OS/390 Networking and Application Enablement Configuration. 63
Our Ethernet LAN Configuration 64
Our ATM Configuration 65
Our Token Ring LAN Configuration 66
More About Our Backbone Token Ring 66
What's Happening in LAN A? 67
What's Happening in LAN B? 68
What's Happening in LAN C? 70
What's Happening in LAN D? 71
Comparing the Network File Systems 72
OS/390 Networking and Application Enablement Workloads 73

Chapter 8. Using The IBM WebSphere Application Server 75
Migrating to the IBM WebSphere Application Server V1.1 for OS/390 R8 . . . 75
Setting Up the IBM WebSphere Application Server to Run in a Read-Only HFS 75
Migrating to the IBM WebSphere Application Server V1.2 for OS/390 R9 . . . 81
Using the IBM WebSphere Application Server V1.2 for OS/390 R10 83
Migrating to the WebSphere Application Server Version 3.02 Standard Edition for OS/390 83
Migrating Applications for the Version 3.02 Level of the IBM WebSphere Application Server 85

Chapter 9. Using the IBM HTTP Server 89
Using the IBM HTTP Server V5.2 89
Using the HTTP Recovery Directive 89
Using Scalable Server Subsystem Mode 89
Using Net.Data to Connect the IBM HTTP Server to DB2 90
Migrating to Shared HFS for the IBM HTTP Server 91
Setting up Shared HFS for the IBM HTTP Server 93
Migrating to The IBM HTTP Server Version 5.3 97
Migrating to System SSL Support 97
Using LDAP to Protect IBM HTTP Server Resources 98
Customizing the IBM HTTP Server Configuration File for LDAP Protection 99
Customizing the LDAP Server for LDAP Protection 100
Verifying LDAP Protection for The IBM HTTP Server 101

Chapter 10. Using OS/390 Infoprint Server 103
Our Infoprint Server Configuration 103
Experiences With Infoprint Server 106
Migrating to Infoprint Server. 106
Migrating PSF to Use the Printer Inventory 106
Setting Up and Using Infoprint Server Transforms 106

Chapter 11. Managing a Hierarchical File System (HFS) 109
Highlights of Our Strategy 109
Migrating to a Shared Hierarchical File System (HFS) 109
The Shared HFS in Our Environment 110
The Sysplex Root HFS 111
The Version HFS. 113
The System Specific HFSs 114

Three Scenarios — How Requests Are Resolved In the Shared HFS Environment . . . 116
Scenario One: Accessing /etc Directory . . . 116
Scenario Two: Accessing Another System's /etc . . . 117
Scenario Three: Accessing Shared Code in /bin . . . 118
Where Am I? Using the set -o logical Command . . . 119
How We Perpared Our Environment for R9 Shared HFS . . . 119
Customizing our Shared BPXPRMxx Member . . . 121
Customizing BPXPRMxx Members for Individual Systems . . . 121
Creating an OMVS Couple Data Set (CDS) . . . 122
Mounting File Systems at /usr . . . 122
Mounting Product File Systems at /usr . . . 122
Mounting File Systems at /usr/spool . . . 125
Our HFS Service Environment . . . 126
High-Level Steps for Servicing the HFS . . . 128
Samples Available on Our Web Site . . . 130
HFS Naming Conventions . . . 130
Our Post-Install Customization EXEC . . . 131
Initializing File Systems in the System Specific HFSs . . . 132
Using the Automount Facility to Manage User Home Directories . . . 133
Performance Considerations for Shared HFS . . . 133
Performance of Read, Write, and Copy of File Systems . . . 134
Performance of File System Mounts . . . 135
Recommendations for Controlling Performance . . . 135

Chapter 12. Setting Up SAP R/3 . . . 137
Our SAP R/3 Configuration . . . 137
Installing the Operating System Collector (SAPOSCOL) . . . 139

Chapter 13. Migrating the OS/390 Security Server LDAP Server to OS/390 R10 . . . 141
Migrating to the New DB2 Backend Database . . . 141

The above chapters describe the networking and application enablement aspects of our computing environment.

Chapter 7. Understanding Our Networking and Application Enablement Environment

In this chapter we describe our networking and application enablement environment, including a high-level view of our configurations and workloads. We discuss networking and application enablement together because the two are greatly intertwined. You need the networking infrastructure in place before you can run many of the application enablement elements and features.

Our OS/390 Networking and Application Enablement Configuration

Figure 8 illustrates at a high level our networking and application enablement configuration. In the figure, the broad arrows indicate general network connectivity of a given type, rather than specific, point-to-point, physical connections.

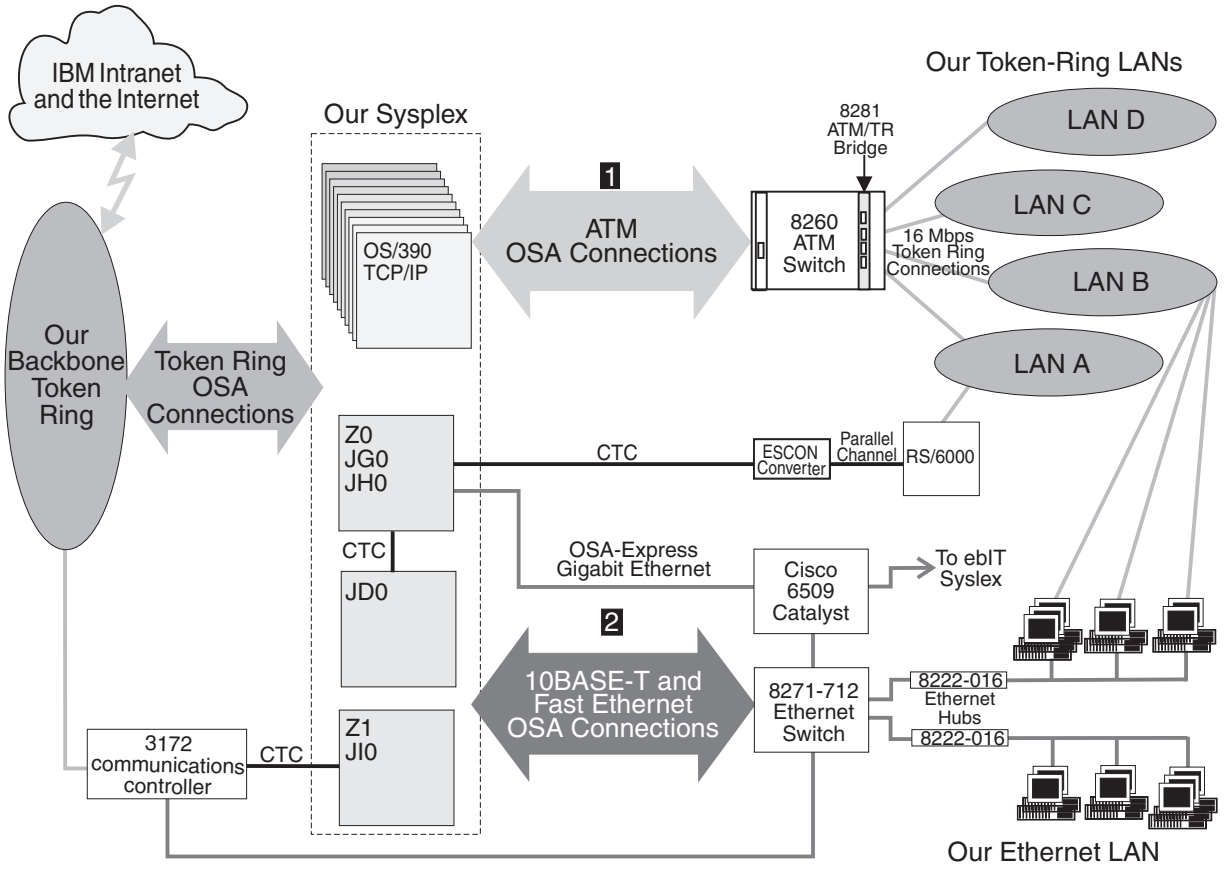


Figure 8. Our Networking and Application Enablement Configuration

Note the following about Figure 8:

- We use the following OSA features to connect our OS/390 systems to our LANs:

  – OSA-2
    - ATM (Asynchronous Transfer Mode)
    - ENTR (Ethernet/Token Ring)
    - FENET (Fast Ethernet)

  – OSA-Express
    - ATM
    - FENET
    - Gigabit Ethernet (GbE)

Networking/Applications Environment

- All ATM connections (1) use ATM LAN emulation; we have no native ATM connections. Although not shown, some of our CPCs that do not have an ATM connection instead use OSA-2 ENTR to directly connect to each of our token-ring LANs.
- Host system Ethernet connections (2) use either OSA-2 ENTR 10BASE-T, OSA-2 FENET, OSA-Express FENET, or OSA-Express Gigabit Ethernet features depending on the CPC model and the type of adapter it supports. Systems Z1 and JI0 do not directly connect to the 8271 Ethernet switch; instead, they use a CTC connection to the 3172 communications controller which then connects to the Ethernet switch.
- All network connectivity for system JD0 routes through systems Z0/JG0/JH0 using a CTC connection between these two CPCs.
- Although not shown, all RS/6000s on LAN A also have a direct connection to the backbone token ring.
- We recently replaced our Gigabit Ethernet switch with a Cisco 6509 Catalyst switch. This new switch handles all of our Gigabit Ethernet and some of our Ethernet connections. Eventually, all of our Ethernet traffic will flow through the Cisco 6509 and we'll remove the 8271-712 Ethernet switch. (For technical details about the Cisco 6500 Catalyst family, go to http://www.cisco.com.)
- We also added Gigabit Ethernet connectivity between our sysplex and a remote sysplex owned by the e-business Integration Test (ebIT) team. The ebIT team focuses on the migration and testing of OS/390 e-business products and applications, with an emphasis on exercising middleware functionality. The connectivity between our two sysplexes allows the team to simulate both local e-business and remote business-to-business environments. Like us, the ebIT team produces a test report to document their results. You can find the *OS/390 e-business Integration Test (ebIT) Report* on the Web at http://www.ibm.com/s390/nc/techinfo/ereports.html.

For an illustration of our VTAM configuration, see "Our VTAM Configuration" on page 13.

If you are familiar with our test reports, then you know that we have always described our networking configuration in exacting detail, as we felt that the complexity of our environment required a great deal of explanation. For example, at one time we were very specific about which of our OS/390 systems could access which of our network resources. However, as we progress, we are concentrating more and more on TCP/IP and expanding our use of Ethernet. As a result, things are becoming more similar than dissimilar and connectivity between our host systems and network resources is approaching any-to-any.

Accordingly, we have shifted our networking discussion to a somewhat more conceptual level and focus on how our infrastructure enables us to test and exploit new features and functions. We will continue to highlight specific aspects of our configuration as significant changes occur and we introduce new technologies.

Our Ethernet LAN Configuration

Our network configuration includes an Ethernet LAN. We primarily use it for FTP testing from Windows 95 and Windows NT clients and for VIPA testing. (For more information about VIPA, see our December '99 edition.) Many of our Ethernet client workstations also contain a token-ring adapter that connects the workstations to our token-ring LAN B as well. We use an OS/2 LAN Server on LAN B to drive the FTP testing on the Ethernet clients. You can read more about this setup in "What's Happening in LAN B?" on page 68.

Our OS/390 systems' Ethernet connectivity includes a combination of 10BASE-T, Fast Ethernet, and Gigabit Ethernet connections using OSA-2 ENTR, OSA-2 and OSA-Express FENET, and OSA-Express Gigabit Ethernet features, respectively. (See page 19 for an important note about using OSA-2 FENET features to communicate between IP stacks in multiple logical partitions on the same CPC.) Instead of OSA features, systems Z1 and JI0 connect to the 3172 communications controller, and the 3172 then connects to our Ethernet LAN.

Note that the connections between our OSA-Express Gigabit Ethernet features and our Cisco 6509 Catalyst switch operate at 1000 Mbps. The Fast Ethernet connections between our OSA-2 and OSA-Express FENET features and our 8271 Ethernet switch, as well as those between our Cisco 6509 and the 8271, operate at 100 Mbps. The 10BASE-T connections from the 8271 to the 8222 Ethernet hubs and client workstations operate at 10 Mbps.

Introducing the OSA-Express FENET Feature: We've added the OSA-Express FENET feature to our configuration, in addition to our OSA-2 FENET features. The OSA-Express FENET feature operates at either 10 or 100 Mbps in half- or full-duplex mode and supports auto-negotiation with its attached Ethernet hub, router, or switch. We used the latest edition of *OSA-Express Customer's Guide and Reference* and the OSA/SF GUI for Windows to install and configure the OSA-Express FENET feature. (See our December 1999 edition for our experiences installing the OSA/SF GUI for Windows.) Using the GUI, we configured the OSA-Express FENET to operate at 100 Mbps in half-duplex mode, and created the OSA Address Table (OAT). (Unlike the OSA-Express Gigabit Ethernet which automatically creates its own OAT, you must manually create an OAT for the OSA-Express FENET, and the OSA/SF GUI simplifies this task.) We also recommend that you check with your IBM support representative to ensure that you have the latest microcode level for this feature.

Our ATM Configuration

As we note above, our configuration includes OSA-2 and OSA-Express ATM features operating in LAN emulation mode only. Therefore, when you see the term *ATM* in this chapter, understand it to mean *ATM LAN emulation*. (See our December 1998 edition for details on our ATM implementation.)

We use ATM for high-speed, bi-directional, asynchronous connectivity between our OS/390 systems and our 8260 ATM switch. The 8260 then connects to the 8281 LAN bridge and provides access to all four of our token-ring LANs. The ATM links operate at 155 Mbps while the token-ring LANs still operate at 16 Mbps. Therefore, the maximum combined token-ring traffic from all four LANs is only 64 Mbps, which *each* ATM link easily accommodates.

Note that because of the wide variety of hardware we employ, not every CPC in our sysplex has an ATM connection. For those CPCs that do not, we use OSA-2 ENTR to provide direct connections to each of our token-ring LANs. Either way, it's all transparent to the end user.

Introducing the OSA-Express ATM Feature: We recently replaced one of our OSA-2 ATM features with an OSA-Express ATM feature. We used the latest edition of *OSA-Express Customer's Guide and Reference* (specifically, SA22-7403 at the -01 level) and the OSA/SF GUI for Windows to install and configure the OSA-Express ATM feature. (See our December 1999 edition for our experiences installing the OSA/SF GUI for Windows). Configuring the OSA-Express ATM using OSA/SF is straight forward, especially if you have any previous OSA experience. In

fact, we found the configuration process to be very similar to our previous OSA-2 ATM and, overall, we used most of the same settings. However, certain settings are switch dependent. To determine the correct values for your installation, be sure to have handy the installation guide or user's guide for your particular ATM switch.

Our Token Ring LAN Configuration

As Figure 8 illustrates, we have a total of five token-ring LANs: a backbone ring and four test LANs that use various:

- Communications protocols (TCP/IP, SNA, NetBIOS, and Internet Packet Exchange (IPX))
- Workstation operating systems (AIX, OS/2, Novell NetWare, PC DOS, and Microsoft Windows 3.1, Windows 95, and Windows NT)
- Workstation types (RS/6000s and various types of PCs)

Note that NetBIOS and IPX are not shown in Figure 8. The OS/2 LAN Servers and requesters (clients) in LANs B and C use NetBIOS to communicate with one another. The Novell NetWare server and clients in LAN D use IPX (Novell NetWare's communications protocol) to communicate with one another.

LANs A, B, C, and D in Figure 8 use only token-ring LAN protocol. The four LANs connect to our OS/390 systems through the 8281 LAN bridge and 8260 ATM switch as described above. (You can read about our ATM experiences in our December 1998 edition.) For host systems running on CPCs that do not have an ATM connection, we instead use OSA-2 ENTR features to provide direct token-ring connections to each of the four LANs (these connections are not shown in Figure 8).

Note that we also have:

- A TCP/IP connection using a CTC from one of our RS/6000 workstations to system Z0. We use a 370 Parallel Channel Adapter in the RS/6000 to channel-attach that workstation to system Z0. We have also configured TCP/IP on system Z0 to use the CLAW protocol to give us a fast point-to-point connection.
- An OS/2 LAN Server with a CLAW protocol channel adapter that connects to system JE0 for LAN Server. This is not shown in Figure 8; see Figure 11 on page 71 for an illustration of this.
- Although also not shown in Figure 8, systems Z0 and JG0 have an OSA-2 OSA token-ring connection to LANs B and D in addition to the OSA-2 ATM feature.

All the systems in our sysplex can connect to the IBM SNA network using VTAM as long as either system Z0 or system J80 (the network node server) is available. In addition, all the systems except JD0 (which is routed through system Z0) can get to the IBM TCP/IP network directly through our backbone token ring.

We discuss how we use the backbone and LANs A, B, C, and D in greater detail in the following sections.

More About Our Backbone Token Ring

The token-ring backbone connects our test environment to the IBM corporate network or intranet and, beyond that, to the Internet. Rather than exist as an isolated entity, our ability to connect to the rest of the corporation and to the outside world yields us a much more viable and robust test environment. Some specific advantages include:

- We are able to access our network resources from our offices or while working from home, instead of having to be on the test floor all the time. This convenience and flexibility allows us to be more productive.
- We can perform more complete and realistic test scenarios with products and features, such as:
  - ADSM
  - Firewall
  - NFS
  - Print Server access
  - Rlogin
  - Telnet
  - Web access
- When we encounter a complex problem, we are able to have product developers from our local site and from other IBM locations work with us in our own test environment to help diagnose and resolve the problem.
- We keep our own documentation, such as test plans and run procedures, on our Web server and can access it from anywhere. As a result, we also implicitly test our networking environment just by performing our day-to-day administrative work.
- We install our configuration tools (for Firewall and OSA/SF, for example) on workstations attached to the backbone so that we can provide central access to the tools and share them across multiple systems.

For many of the same reasons, we also recently switched from running our RS/6000 workloads on LAN A to running them on the backbone, mostly to allow greater access to other resources and provide more realistic testing. See the next section for more about LAN A.

What's Happening in LAN A?

Our RS/6000 workstations reside on LAN A but they also directly connect to the backbone. This additional connectivity allowed us to shift a majority of the workloads that we once performed exclusively on LAN A over to the backbone. LAN A itself still exists in our environment, but we don't use it for anything special from a functional standpoint.

Figure 9 on page 68 depicts our OS/390 UNIX DCE test configuration in LAN A, including the connections from the RS/6000s to the backbone (which, for clarity, are not shown in Figure 8 above).

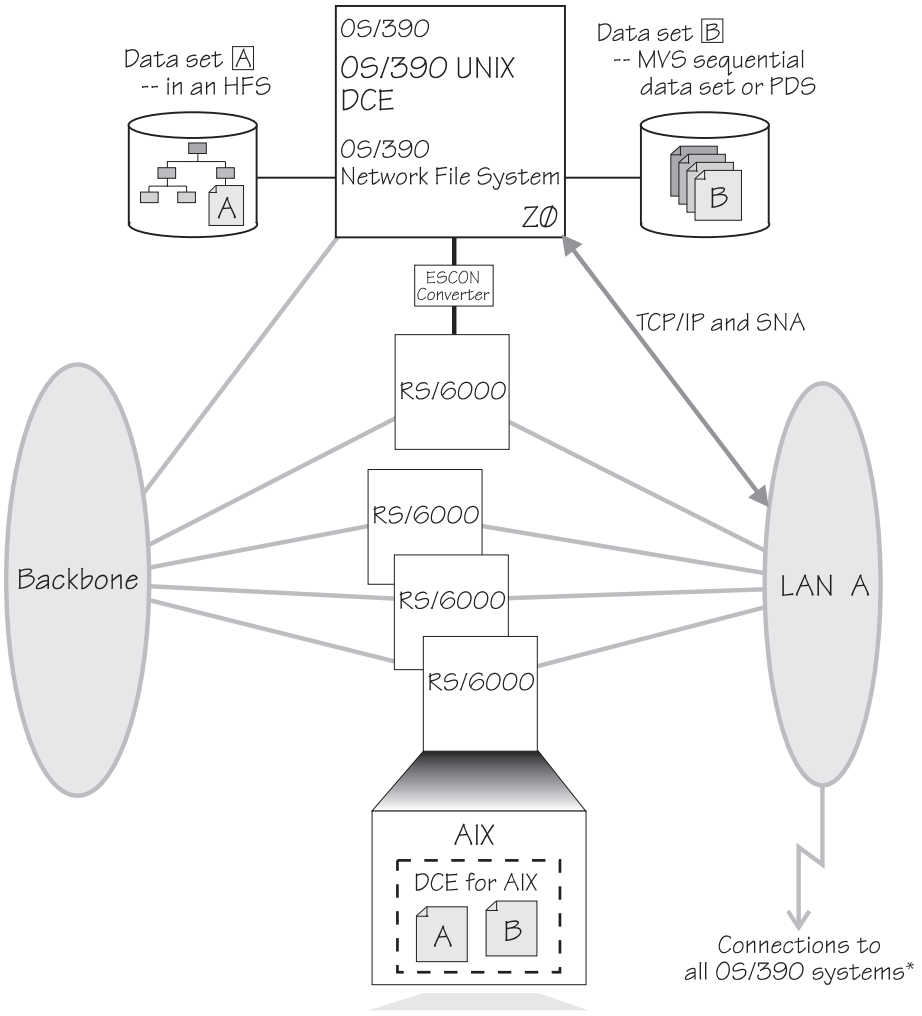


Figure 9. Our token-ring LAN A

The RS/6000 workstations on LAN A all run the AIX operating system. We use them to exercise the OS/390 UNIX, DCE, and DFS functions. See “Our Workloads” on page 14 for a more detailed description of these workloads. For more information about DCE and DFS, see “DCE and DFS Publications” in “Appendix D. Useful Publications and Web Sites” on page 217.

What’s Happening in LAN B?

You might recall from our December 1996 edition that our LANs B and C started out as two functionally separate LANs. Later on, we combined their functionality and collectively referred to them as logical LAN BC. Well, we’ve now come full circle. For better performance and throughput, we are back to using LANs B and C as two functionally separate LANs.

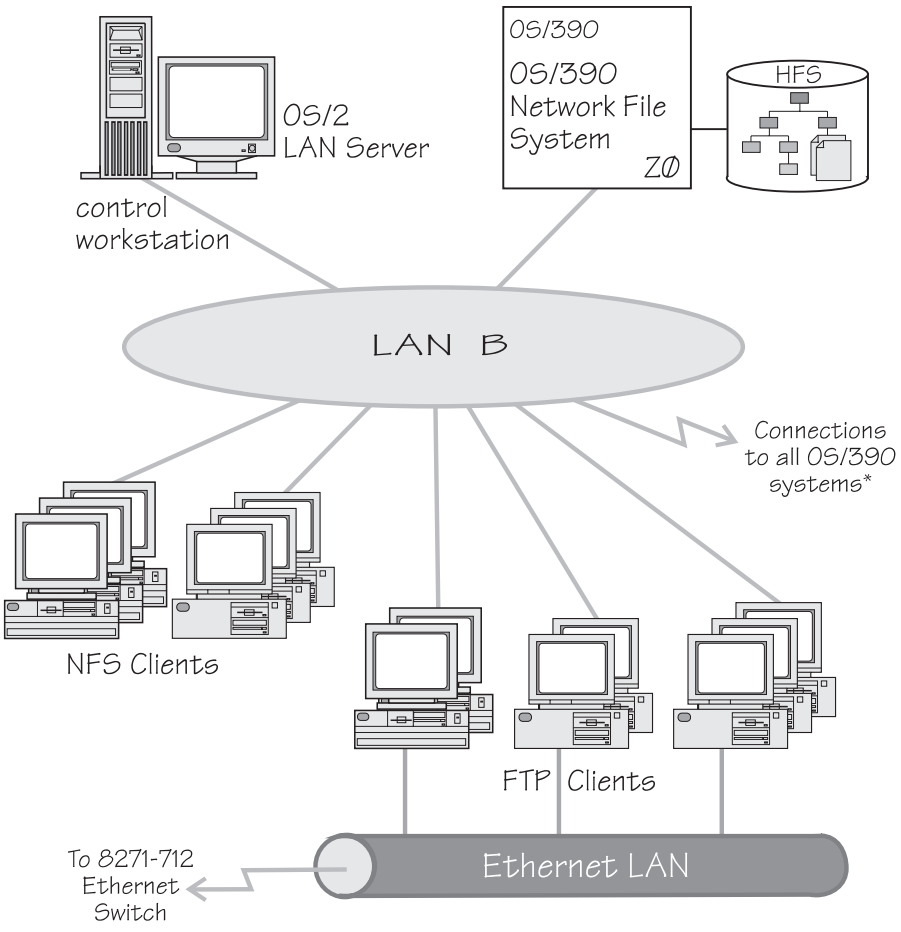
LAN B has an OS/2 NFS function and an FTP function using TCP/IP. The OS/2 LAN Server on LAN B acts as a control workstation for our NFS and FTP workloads. The control data consists of the commands that start, stop, and otherwise regulate the execution of the workloads. The test data or workload data is the actual data that the workloads manipulate.

The workstations that run the FTP workloads connect to both our token-ring LAN B and to our Ethernet LAN. The FTP control data comes from the OS/2 server to the clients over LAN B. The test data that the workloads manipulate travels over the Ethernet LAN.

The NFS function communicates with OS/390 NFS using TCP/IP, and both the control data and the workload data travel over LAN B. (See “Comparing the Network File Systems” on page 72 for a description of the different types of NFSs we use.)

We recently upgraded many of the client workstations on LAN B to be Pentium 266 MHz machines with 64 MB of RAM, running either Windows 95 or Windows NT along with ADSM, FTP, NFS, and Web browsers. In addition, we still have some older 386 and 486 machines with 16 to 32 MB of RAM, running OS/2 along with NFS, LFS, and LANRES client software to give us more workload diversity.

Figure 10 depicts our NFS and FTP test configuration in LAN B. For more information about NFS, see “Network File System Publications” in “Appendix D. Useful Publications and Web Sites” on page 217.



*Except JD0, JI0 and Z1. Connections are either through the 8281 ATM LAN bridge or directly to ENTR OSAs. (Systems JI0 and Z1 only connect to the backbone, and JD0 only routes through Z0.)

Figure 10. Our token-ring LAN B

What's Happening in LAN C?

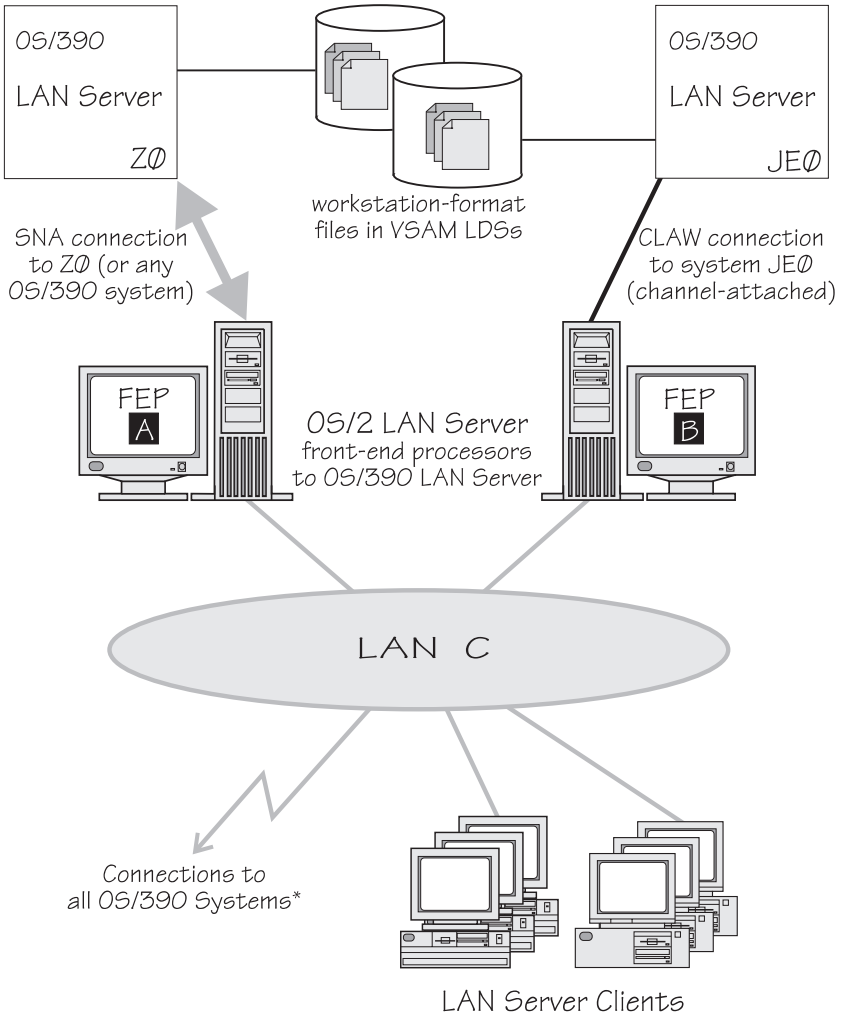
LAN C runs two different types of LAN Server scenarios using OS/2 LAN Servers as front-end processors (FEPs) to OS/390 LAN Server. OS/390 LAN Server expands the file storage capability of the OS/2 LAN Servers by storing workstation-format files in VSAM linear data sets on the OS/390 host. These data sets are not readable by MVS users, but appear to the clients as though they are stored on the OS/2 LAN Servers.

First, we have an OS/2 LAN Server acting as a FEP (**A**) with a SNA connection to LAN Server in system Z0. We could conceivably connect the OS/2 LAN Server to any OS/390 system, but we currently happen to be using Z0. We use Communications Manager/2 for the SNA connection and APPC communications.

We also have another OS/2 LAN Server acting as a FEP (**B**) with a CLAW protocol connection to LAN Server in system JE0.

Typically, a LAN file server contains one or more large-capacity hard disk drives on which it stores files for access by the clients (or requesters). However, in our setup, the OS/2 LAN Servers do not store any workload-related programs or data on their own hard disks for use by the clients. All the workload-related programs and data reside on the OS/390 system. This is completely transparent to the requesters, as they are only aware of the OS/2 LAN Servers which, in turn, interact with OS/390 LAN Server on the host. The OS/2 servers do keep setup files, automation programs, and workstation configuration files on their own local disk drives.

Figure 11 on page 71 depicts our LAN Server test configuration in LAN C. For more information about LAN Server, see "LAN Server Publications" in "Appendix D. Useful Publications and Web Sites" on page 217.



*Except JD0, JI0 and Z1. Connections are either through the 8281 ATM LAN bridge or directly to ENTR OSAs. (Systems JI0 and Z1 only connect to the backbone, and JD0 only routes through Z0.)

Figure 11. Our token-ring LAN C

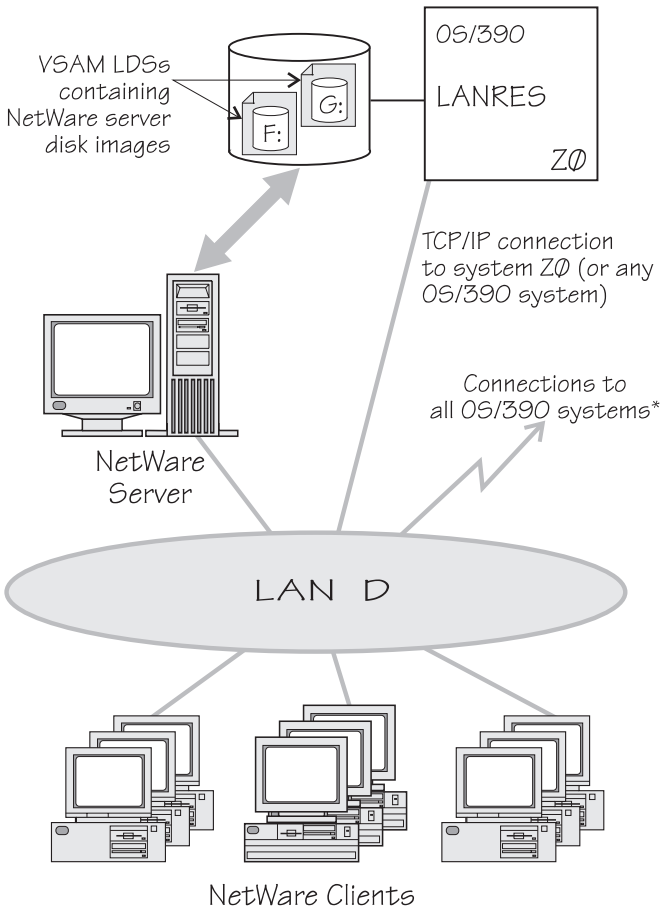
What's Happening in LAN D?

LAN D consists of a Novell NetWare server and clients that access files using OS/390 LANRES (LAN Resource Extension and Services). The server is a Pentium 166 MHz system with 64 MB of RAM and runs the NetWare 4.1 server operating system. The clients are a combination of PC DOS, OS/2, and Windows 95 machines running NetWare client software. The server and clients use IPX to communicate with each other.

The NetWare server uses TCP/IP to communicate with LANRES on the OS/390 system. Currently, we have the NetWare server connected to LANRES running on system Z0; however, our TCP/IP connectivity allows us to conceivably run it on any OS/390 system, or multiple systems. The EWXCOMM definition in LANRES identifies and controls the connection to the NetWare server.

Networking/Applications Environment

Figure 12 depicts our NetWare and LANRES test configuration in LAN D. For more information about LANRES, see “LANRES Publications” in “Appendix D. Useful Publications and Web Sites” on page 217.



*Except JD0, JI0 and Z1. Connections are either through the 8281 ATM LAN bridge or directly to ENTR OSAs. (Systems JI0 and Z1 only connect to the backbone, and JD0 only routes through Z0.)

Figure 12. Our token-ring LAN D

LAN D is similar to the LAN Server function of LAN C described earlier in that, for the purposes of processing the workloads, the NetWare clients communicate only with the NetWare server, and only the server communicates with LANRES on the OS/390 host. The NetWare server does not store any workload-related programs or data on its own disk drives for use by the clients. Instead, it gets all workload-related programs and data from the host, and it is transparent to the clients which host system is processing their requests. However, the server does keep on its own workstation things like logon scripts, setup data, and spooled print jobs.

Comparing the Network File Systems

If you are a faithful reader of our test report, you might have noticed that we have changed our Network File System (NFS) approach a number of times, depending on the circumstances at the moment. Currently, we have the OS/390 NFS (called DFSMS/MVS NFS in OS/390 releases prior to R6) on system Z0, and the LAN Server NFS on system JE0. These two different NFSs cannot be connected to the same TCP/IP address space at the same time because they use the same port. So,

a good way to avoid that is to run them on different systems. When we last reported on this configuration, we indicated that we ran them on the same system, using the same TCP/IP stack, but used them at different times. If you run your NFSs on different systems, you won't have to do that. The reason we changed to using them on different systems is that we implemented NFS server and client support for managing shared user home directory HFSs. See the HFS information in our December 1998 edition for more information.

Both of the NFSs allow files to be transferred between the server and the workstation clients. To the clients, the data appears to reside on a workstation fixed disk, but it actually resides on the OS/390 server.

With OS/390 NFS, data that resides on the server for use by the workstation clients can be either of the following:

- OS/390 UNIX files that are in a hierarchical file system (HFS). The OS/390 NFS is the only NFS that can access files in an HFS. You need to have OS/390 NFS on the same system as OS/390 UNIX and its HFS if you want to use the NFS to access files in the HFS.
- Regular MVS data sets such as PS, VSAM, PDSs, PDSEs, sequential data striping, or direct access.

With the LAN Server NFS, the data sets to be transferred to the workstations can reside only in special VSAM linear data sets that MVS users cannot read.

Migrating to the OS/390 NFS: We plan to implement some of the new functions available in OS/390 NFS, such as file locking over the OS/390 NFS server and file extension mapping support. You can read descriptions of these new functions in *OS/390 Network File System Customization and Operation* and *OS/390 Network File System User's Guide*. In addition, you can read about WebNFS support in our December 1999 edition. We hope to have additional experiences with these new functions to share with you in a future test report.

In the meantime, we'd like to highlight one aspect of the migration to the OS/390 NFS. ***Pay attention to the following words*** in the section on allocating the mount handle data sets in *OS/390 Network File System Customization and Operation*: "Delete and allocate the mount handle data sets before running any new versions of the Network File System. If an old mount handle data set is used, the server issues a message and shuts down." We somehow missed this and attempted to migrate without deleting our old data sets and recreating them. When the server shut down, we had a difficult time figuring out why.

Note that APAR OW40134 recommends a change to the SHAREOPTIONS specified in the sample JCL for the IDCAMS job used to allocate the mount handle data sets. This sample JCL is both shipped in *hlq*.NFSSAMP(GFSAMHDJ) and illustrated in *OS/390 Network File System Customization and Operation*. The sample JCL currently uses SHAREOPTIONS(3 3). However, the APAR instead recommends SHAREOPTIONS(1 3). While the sample code does work as it stands, it allows programs other than NFS to update the files. Using SHAREOPTIONS(1 3) limits the possibility of corruption to the mount handle database.

OS/390 Networking and Application Enablement Workloads

For information about our networking and application enablement workloads, see "Our Workloads" on page 14.

Chapter 8. Using The IBM WebSphere Application Server

This chapter describes our experiences using The IBM WebSphere Application Server in the OS/390 R8, R9, and R10 time frames.

There is also a Web-based user interface, the *WebSphere Troubleshooter for OS/390*, which includes the most current debugging and tuning hints for all supported releases of the IBM HTTP Server and the IBM WebSphere Application Server. It also includes hints for ServletExpress and the WebSphere Application Server. Look for this at the following URL:

`http://www.ibm.com/software/webservers/httpservers/troubleshooter.html`

In this chapter, we'll discuss the following topics:

- "Migrating to the IBM WebSphere Application Server V1.1 for OS/390 R8"
  - "Setting Up the IBM WebSphere Application Server to Run in a Read-Only HFS"
- "Migrating to the IBM WebSphere Application Server V1.2 for OS/390 R9" on page 81
- "Using the IBM WebSphere Application Server V1.2 for OS/390 R10" on page 83
- "Migrating to the WebSphere Application Server Version 3.02 Standard Edition for OS/390" on page 83

Migrating to the IBM WebSphere Application Server V1.1 for OS/390 R8

To migrate the IBM WebSphere Application Server V1.1 to the OS/390 R8 level, we used *Application Server Planning, Installing, and Using V1.1 for OS/390*.

The major change we made for the OS/390 R8 level of the IBM WebSphere Application Server was setting up to run the Application Server in a read-only HFS. See "Setting Up the IBM WebSphere Application Server to Run in a Read-Only HFS".

Setting Up the IBM WebSphere Application Server to Run in a Read-Only HFS

Note: If you have applied PTF UQ37258 for Releases 7 or 8, you should not use this procedure for setting up the WebSphere Application Server to run in a read-only HFS. PTF UQ37258 provides a new way of customizing the Application Server. (PQ28467 in PTF UQ37258 introduces configuring the WebSphere Application Server with the updateproperties utility and was.conf instead of the administration GUI, Application Server Manager.)Because our testing was done before PTF UQ37258 was available, we were not able to test our procedure with the new function provided in this PTF.

If you have PTF UQ37258 installed, you should use the latest level of *Application Server Planning, Installing, and Using V1.1*, which talks about using APARs PQ23479 (for releases 5 and 6) and PTF UQ37258 (for releases 7 and 8) in the *About This Book* section.

Our HFS strategy is to run with our serviceable product directories HFS's shared between the systems in our sysplex in read-only mode. (See "Chapter 11. Managing a Hierarchical File System (HFS)" on page 109.) However, the WebSphere Application Server mixes serviceable and customizable files together in product

directories. (The future direction of the WebSphere Application Server is to separate serviceable and customizable files so that serviceable files can be run from a read-only HFS environment.)

Note that this procedure applies **only** to the version 1.1 level of the WebSphere Application Server.

In OS/390 R7, we solved the problem of mixed serviceable and customizable files by moving the WebSphere Application Server into a read-write HFS. (See our December '99 edition). However, that solution doesn't really fit our strategy of sharing serviceable code across the sysplex in read-only mode, and putting the customizable information into system-unique read-write HFSs. It also makes picking up service a very manual process. So, as part of the migration to the OS/390 R8 level of the IBM WebSphere Application Server, we decided to set up the Application Server to run in a read-only HFS environment.

On a high level, we made two changes to run in a read-only HFS:

- Separating our serviceable from our customizable code for the application server. To do this, we left the serviceable code in a read-only HFS and put the customizable data in read-write HFSs. See Figure 13 on page 77 for an overview of that process. Note that the numbers and letters shown in the figure correspond to steps in "Setting up our IBM WebSphere Application Server to Run in a Read-Only HFS" on page 77.
- We moved code that writes to the root HFS from the post-install job supplied with the code (postinstall.sh) to a job that runs within the service process (WAS.sh). We moved this code because in our new read-only HFS environment, the root HFS is no longer writable at post-install time. In OS/390 R7, since we ran the Application Server in a read-write HFS environment, the post-install job could write to the root, but in a read-only root environment, the root HFS is only writable during the service process.

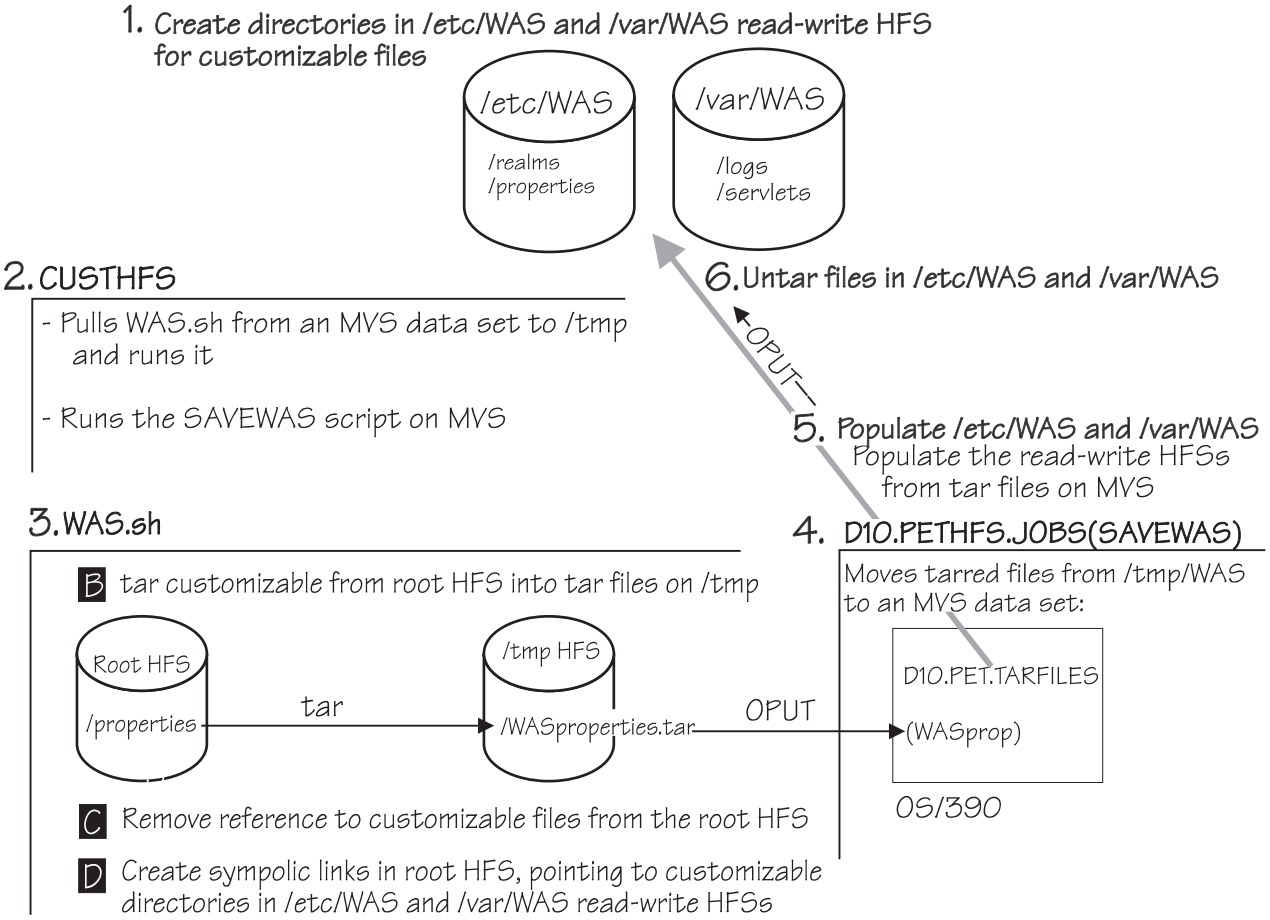


Figure 13. Moving Customizable Application Server Directories From the Root HFS into Read-Write HFSs

Note that the following scripts and programs referenced below are included in the samples on our Web site:

- CUSTHFS
- WAS.sh
- SAVEWAS
- petpostinstall.sh

In detail, our process was as follows:

1. We created directories in /etc/WAS and /var/WAS for our customizable information, to which we'll need write access. The customizable directories include realms, properties, logs, and servlets.

   We created the following directories for realms and properties in /etc/WAS using the following UNIX System Services commands:

   ```
   mkdir realms
   mkdir properties
   ```

   We created the following directories for logs and servlets in /var/WAS using the following UNIX System Services commands:

   ```
   mkdir logs
   mkdir servlets
   ```

Note that logs in /var can tend to eat up a lot of storage. You should monitor the space available in /var when you are running with logging turned on so that you don't fill it. Further, logging should be limited to times when you are diagnosing a specific problem.

We create these /etc and /var directories once per system in our sysplex.

2. We made changes to CUSTHFS, which is our exec that customizes the root read-only HFS. CUSTHFS runs at service time, the only time the root HFS is writable. We added the following to CUSTHFS to call two additional programs, WASSH and SAVEWAS:

```
/* ************************************************ */
/* Updates to allow WAS to run in a R/O root        */
/* ************************************************ */
say "Running WAS R/O fix job"
address tso "oput 'D10.PETHFSN.JOBS(WASSH)' '/tmp/WAS.sh'"
'chmod /tmp/WAS.sh 777'
address tso "oshell /tmp/WAS.sh "$root
address tso "ex 'D10.PETHFS.JOBS(SAVEWAS)'"
/* ************************************************ */
/* End of Updates to allow WAS to run in a R/O root */
/* ************************************************ */
```

   These lines pull the WAS.sh shell script from an MVS data set and put it in the /tmp directory to run it. They also run the SAVEWAS script on MVS.

3. We created a WAS.sh shell script. In a number of steps, this script removes our customizable files from the root HFS and includes the sections of code from our post-install script that require a writable root HFS. The letters, such as **A** correspond to sections of the Figure 14 on page 79 and Figure 13 on page 77.):

   To remove the customizable files from the root HFS, our WAS.sh shell script does the following:

   - **A** — The first section of WAS.sh simply creates variables for our customizable properties, logs, realms and servlets directories. These variables will be used in symbolic links created in section **D**.
   - **B** — This section of WAS.sh issues the UNIX system services tar command for each of the customizable directories (/usr/lpp/WebSphere/AppServer/properties, logs, realms and servlets). The tar command takes the contents of an entire directory and puts it into a single file. In our case, we call the tarred files WAS*xxxx*.tar, where *xxxx* is either properties, logs, realms, or servlets. We place these files into a temporary directory, /tmp. The properties directory, for example, is tarred into /tmp/WASproperties.tar. This is done in preparation for moving them into a read-write HFS data set.
   - **C** — Here, we remove the old customizable directories, /usr/lpp/WebSphere/AppServer/properties, /logs, /realms, and /servlets. We'll be replacing them with symbolic links to our customizable directories in our writable HFSs, /etc/WAS and /var/WAS.
   - **D** — We create the symbolic links for the customizable directories in /etc/WAS and /var/WAS. These create a link from the read-only root to the read-write customizable directories in /etc/WAS and /var/WAS.

   We moved the following code from our post-install script to WAS.sh, because these steps require a writable root HFS (the root will not be writable at post-install time):

   - **E** — This section, which adds symbolic links in the read-only root to the language specific code, was taken from our post-install script, petpostinstall.sh. The postinstall.sh job shipped with the product gives us the option to use either English or Japanese, but we hardcoded English as our language choice in this section. We moved these lines to WAS.sh from

postinstall.sh because at the time that WAS.sh runs in the service process, we can still write to the root HFS to make these updates. After installation, when postinstall.sh runs, we cannot update the root HFS because it is read-only. In the OS/390 R7 time frame, all our IBM WebSphere Application Server directories were in a read-write HFS, so the postinstall.sh could effect changes in the HFS. See our December '99 edition for more information.

- **F** — Finally, we change the owner for all directories in the root to WEBADM.

  After making these changes to our post-install job, we renamed it to petpostinstall.sh.

```
# shell script to allow WAS to work in R/O Root A
echo "Entering WASSH"
echo "Assigning variables"
root=$1
properties=${root}/usr/lpp/WebSphere/AppServer/properties
logs=${root}/usr/lpp/WebSphere/AppServer/logs
realms=${root}/usr/lpp/WebSphere/AppServer/realms
servlets=${root}/usr/lpp/WebSphere/AppServer/servlets
# tar up directories to be saved  B
cd ${root}/usr/lpp/WebSphere/AppServer/properties
tar -cvzfU /tmp/WASproperties.tar -C .
cd ${root}/usr/lpp/WebSphere/AppServer/logs
tar -cvzfU /tmp/WASlogs.tar -C .
cd ${root}/usr/lpp/WebSphere/AppServer/realms
tar -cvzfU /tmp/WASrealms.tar -C .
cd ${root}/usr/lpp/WebSphere/AppServer/servlets
tar -cvzfU /tmp/WASservlets.tar -C .
cd ${root}/usr/lpp/WebSphere/AppServer
# remove directories to allow for creation of sym links  C
rm -rf ${root}/usr/lpp/WebSphere/AppServer/properties
rm -rf ${root}/usr/lpp/WebSphere/AppServer/logs
rm -rf ${root}/usr/lpp/WebSphere/AppServer/realms
rm -rf ${root}/usr/lpp/WebSphere/AppServer/servlets
# create symbolic links  D
ln -s /etc/WAS/properties ${properties}
ln -s /etc/WAS/realms ${realms}
ln -s /var/WAS/logs ${logs}
ln -s /var/WAS/servlets ${servlets}
# adding symbolic links to populate English language specific code
# This step removed from postinstall.sh   E
cd ${root}/usr/lpp/WebSphere/AppServer/doc
ls -l en | awk '$1 ~ /¬-/ {print " ln -s en/" $9 " ."}' | sh -s
cd ${root}/usr/lpp/WebSphere/AppServer/doc/apidocs
ls -l en | awk '$1 ~ /¬-/ {print " ln -s en/" $9 " ."}' | sh -s
cd ${root}/usr/lpp/WebSphere/AppServer/doc/apidocs/images
ls -l en | awk '$1 ~ /¬-/ {print " ln -s en/" $9 " ."}' | sh -s
cd ${root}/usr/lpp/WebSphere/AppServer/doc/guide
ls -l en | awk '$1 ~ /¬-/ {print " ln -s en/" $9 " ."}' | sh -s
cd ${root}/usr/lpp/WebSphere/AppServer/system/admin
ls -l en | awk '$1 ~ /¬-/ {print " ln -s en/" $9 " ."}' | sh -s
cd ${root}/usr/lpp/WebSphere/AppServer/web/admin
ls -l en | awk '$1 ~ /¬-/ {print " ln -s en/" $9 " ."}' | sh -s
# Changing permissions to all files and directories.
# This step removed from our post-install script  F
echo ${root}"/usr/lpp/WebSphere/AppServer to WEBADM"
chown -R WEBADM:IMWEB ${root}/usr/lpp/WebSphere/AppServer
```

Figure 14. Our WAS.sh Shell Script

4. We created the SAVEWAS script, which is stored in our MVS data set D10.PETHFS.JOBS(SAVEWAS). SAVEWAS moves the tar files from /tmp and

stores them in an MVS data set. We put our tar files in an MVS data set so that we can access them easily once we're in a read-only environment. Our SAVEWAS script is as follows:

```
/* rexx */
/* Get the tarred up directories currently kept /tmp and store */
/* them in */ /* an MVS dataset */

"oget '/tmp/WASproperties.tar' 'd10.pet.tarfiles(WASprop)' binary"
"oget '/tmp/WASlogs.tar' 'd10.pet.tarfiles(WASlogs)' binary"
"oget '/tmp/WASrealms.tar' 'd10.pet.tarfiles(WASrealm)' binary"
"oget '/tmp/WASservlets.tar' 'd10.pet.tarfiles(WASservl)' binary"

exit
```

5. If you're setting up the Application Server for the **first** time, it's time to populate the /var/WAS and /etc/WAS directories created to hold customizable code. We take the tarred files from the MVS data set D10.PET.TARFILES and put them into /etc/WAS and /var/WAS directories. The symbolic links in the root HFS link to the customizable files in /etc/WAS and /var/WAS. This step must be done after the service process has ended. We issued the following UNIX system services commands to populate our /etc/WAS and /var/WAS directories:

   ```
   oput 'd10.pet.tarfiles(WASprop)'  '/etc/WAS/properties/WASprop.tar' binary
   oput 'd10.pet.tarfiles(WASrealm)' '/etc/WAS/realms/WASrealm.tar' binary
   oput 'd10.pet.tarfiles(WASlogs)'  '/var/WAS/logs/WASlogs.tar' binary
   oput 'd10.pet.tarfiles(WASservl)' '/var/WAS/servlets/WASservl.tar' binary
   ```

   Note that these commands can also be coded to run in a job.
6. Next, we un-tarred these /etc/WAS and /var/WAS directories, using the following UNIX system services commands (again, note that these commands may be put in a job):
   - From the /etc/WAS/properties directory, we issued the following tar command to split /properties back into it's subdirectories and files:

     ```
     tar -xvzfp WASprop.tar
     ```
   - From the /etc/WAS/realms directory, we issued the following command to split /realms back into it's subdirectories and files:

     ```
     tar -xvzfp WASrealm.tar
     ```
   - From the /var/WAS/logs directory, we issued the following tar command to split /logs back into it's subdirectories and files:

     ```
     tar -xvzfp WASlogs.tar
     ```
   - From the /var/WAS/servlets directory, we issued the following command to split /servlets back into it's subdirectories and files:

     ```
     tar -xvzfp WASservl.tar
     ```
7. Next, we modified our post-install program, creating a new one which we called petpostinstall.sh:
   - We commented out a couple of sections of our post-install program, ″Created Directory and Move Files Appropriately for Specific Languages″ and ″Change File Permissions to all Files and Directories appropriately″. These sections were moved to our WAS.sh script, see sections **E** and **F** of ″Our WAS.sh shell script″ 3 on page 78.
   - We added the following UNIX System Services chown command to give our Web administrator user ID, WEBADM, access to our customizable directories:

     ```
     chown -R WEBADM:IMWEB $SE_ROOT/properties
     chown -R WEBADM:IMWEB $SE_ROOT/logs
     chown -R WEBADM:IMWEB $SE_ROOT/realms
     chown -R WEBADM:IMWEB $SE_ROOT/servlets
     ```
8. After service has been updated, we run our modified post-install program, petpostinstall.sh.

9. Now, we're ready to start the web server in a read-only environment.

Migrating to the IBM WebSphere Application Server V1.2 for OS/390 R9

To migrate the IBM WebSphere Application Server V1.2 for OS/390 R9, we used the *Application Server Planning, Installing, and Using V1.2 for OS/390* book.

To achieve the results we describe in this section for the Application Server, you must install the fix for HIPER APAR PQ30725.The biggest change for the Application Server V1.2 for OS/390 R9 is that we are now running in shared HFS mode. For us, this basically means that our file systems are accessible by users on any system in our sysplex. See "Migrating to a Shared Hierarchical File System (HFS)" on page 109 for information on how we migrated our sysplex to shared HFS mode.

We had the following experiences in migrating to WebSphere Application Server V1.2 for OS/390 R9:

The WebSphere Application Server Separates Serviceable and Customizable Files: For R9, the Application Server V1.2 code is shipped with customizable and serviceable files already separated. (In R8, they were mixed in product directories.) This means that for our R9 Application Server, we no longer have to follow the steps in "Setting Up the IBM WebSphere Application Server to Run in a Read-Only HFS" on page 75 to separate customizable and serviceable code.

Creating server_model_roots For Each System: As part of our migration, we created an HFS for each system, which we mounted at the /WebSphere directory. We then added the /WebSphere directory to the system specific HFS for each system. (See "The System Specific HFSs" on page 114 for information.) This lets us create a customized server_model_root for the instance of the Application Server running on each of our systems. The server_model_root is a directory structure containing the configuration properties for an individual Application Server. Our server_model_root is /WebSphere/AppServer on each system.

High-Level Migration Steps: We used the steps documented in *Application Server Planning, Installing, and Using V1.2 for OS/390* to migrate to WebSphere Application Server 1.2, but we did not have to do all the steps covered because we were migrating rather than installing for the first time. The following list shows the steps we took, at a high level, to migrate to WebSphere Application Server 1.2 for R9 :

- Made sure the IBM HTTP Server was not running.
- Made sure that our JAVA_HOME root directory points to our Java Development kit (JDK) root directory.
- We next ran the makeserver.sh shell script to customize the Application Server. From /usr/lpp/WebSphere/AppServer/config we entered the following:

  ```
  ./makeserver.sh /WebSphere/AppServer /dgw_pet/httpd.conf $JAVA_HOME
  ```
- Then, we checked to make sure that the program control permission bits were correct, giving the Application Server read-write access to the directory structure.
- We updated our httpd.conf file to replace the ServerInit statement for the V1.2 level of Application Server. Our old ServerInit directive read:

  ```
  ServerInit /usr/lpp/WebSphere/AppServer/lib/libadpter.so:AdapterInit
  /usr/lpp/WebSphere/AppServer/properties/server/servlet/servletservice/jvm.properties
  ```

  Note that this directive is actually all on one line in the httpd.conf file.

  The new ServerInit directive for V1.2 reads:

```
ServerInit /usr/lpp/WebSphere/AppServer/lib/libadpter.so:AdapterInit
/WebSphere/AppServer
```

Note that this directive is actually all on one line in the httpd.conf file.

Our existing ServerTerm and Service directives were still valid.

- Finally, we started the Application Server.

Unable to Access Sample Servlets: When we initially tried to access one of the sample servlets, HelloWorldServlet, we got the following error message:

```
404 Not Found
Cannot find class for servlet HelloWorldServlet: HelloWorldServlet
```

This turned out to be a problem with the permission bits for the /WebSphere directory. We solved this problem by changing the permission bits from 700 to 755. We were then able to access sample servlets successfully.

WebSphere Application Server Hardcoded a Version HFS Name in Symbolic Link: When we ran the makeserver.sh script to customize our individual Application Servers on each system, we ran into a problem with some symbolic links it placed in each system's system specific HFS. Because it is a problem involving the shared HFS environment, we request that you read "Migrating to a Shared Hierarchical File System (HFS)" on page 109 for shared HFS concepts and vocabulary.

Following the shared HFS strategy, we put our serviceable, non-customizable code into the version HFS so that the whole sysplex can share it. For example, the shared serviceable code for WebSphere Application Server is in the version HFS in the following directory path:

```
/usr/lpp/WebSphere/AppServer
```

We also ran the makeserver.sh script to customize our individual Application Servers on each system. That is where the problem occurred. The makeserver.sh script placed the following symbolic links in each system's **system specific HFS**:

```
testservlets -> / D83PC1 /usr/lpp/WebSphere/AppServer/testservlets
lib -> / D83PC1 /usr/lpp/WebSphere/AppServer/lib
config -> / D83PC1 /usr/lpp/WebSphere/AppServer/config
```

These links are designed to link a request for /lib, /config, or /testservlets to the shared code in the version HFS where it resides. The first portion of the directory path, **D83PC1** , is the directory where the version HFS is mounted. In our environment, the version HFS always resolves to the SYSRES volume that the system is IPLed from. We set this up using a symbol, $VERSION, so that it always resolves to the version HFS currently in use. We use the symbol because the version HFS in use for a system can change any time we IPL the system, for example, when we perform our service procedure as recommended by UNIX System Services. Note, however, that the links created by makeserver.sh have the version HFS, D83PC1, hard-coded in the directory path. WebSphere Application Server inserted the version HFS in use by the system where we ran makeserver.sh. This works fine until we change the version HFS we're using. When the version HFS in use changes, the symbolic links to /D83PC1/usr/lpp/WebSphere/AppServer/testservlets, /lib, and /config are no longer valid, because the version HFS is no longer mounted at the /D83PC1 directory. After the version HFS in use changes, users invoking the Application Server to access servlets like HelloWorldServlet, for example, get the following browser error:

```
The document contained no data. Try again later, or contact the server's administrator.
```

To get around this problem, we had to remove the version HFS, /D83PC1 in our case, from the symbolic links in our server_model_root, /WebSphere/AppServer. We did this in the following steps:

- From the /WebSphere/AppServer directory, we entered the following to remove the symbolic links to testservlets, lib, and config containing the specific version HFS name:

```
rm -R testservlets
rm -R lib
rm -R config
```

- We added back the symbolic links without the version HFS name:

```
ln -sR /usr/lpp/WebSphere/AppServer/testservlets testservlets
ln -sR /usr/lpp/WebSphere/AppServer/lib lib
ln -sR /usr/lpp/WebSphere/AppServer/config config
```

Now, the directory paths for testservlets, lib, and config in our system specific HFS are as follows:

```
testservlets -> /usr/lpp/WebSphere/AppServer/testservlets
lib -> /usr/lpp/WebSphere/AppServer/lib
config -> /usr/lpp/WebSphere/AppServer/config
```

Now when a user requests /usr/lpp/WebSphere/AppServer/testservlets, for example, it will resolve to $VERSION/usr/lpp/WebSphere/AppServer/testservlets, where $VERSION is the version HFS currently in use. This means that users and the WebSphere Application Server will always be able to access the code in these directory paths.

Using the IBM WebSphere Application Server V1.2 for OS/390 R10

In the OS/390 R10 timeframe, we're running version 1.2 of the IBM WebSphere Application Server along with the IBM HTTP Server version 5.3. At these levels, we recommend the following Java service:

- The fix for APAR OW44577
- The fix for APAR OW44916

These APAR fixes prevent a problem we ran into where the IBM HTTP Server abends when the IBM WebSphere Application Server is enabled in the httpd.conf file with Java at the May 5th, 2000 build date level (PTF UW69732).

Note that in OS/390 R10, the IBM WebSphere Application Server is no longer shipped as an element. You can download the 1.2 version of the IBM WebSphere Application Server from the Web at the following URL:

`http://www.ibm.com/software/webservers/appserver/download_v12.html`

Migrating to the WebSphere Application Server Version 3.02 Standard Edition for OS/390

During our test of OS/390 R10, we installed WebSphere Application Server Version 3.02 Standard Edition for OS/390. We were able to get the version 3.02 of the IBM WebSphere Application Server up and running very quickly with the default configuration file using the following books:

- *WebSphere Application Server for OS/390 Getting Started*
- *WebSphere Application Server Standard Edition Planning, Installing and Using*.

To get the **latest** WebSphere Application Server books, go to

`http://www.ibm.com/software/websphere/appserv/library_390.html`

Using WebSphere Application Server

The installing and customizing section of the *WebSphere Application Server Standard Edition Planning, Installing and Using* book contains a procedure we used to get the IBM WebSphere Application Server up and running.

Note that IBM has made extensive changes to the IBM WebSphere Application Server between the 1.2 version and the 3.02 version, so it's important that you read the *WebSphere Application Server Standard Edition Planning, Installing and Using* book **before** migrating from the 1.2 version of the IBM WebSphere Application Server to the 3.02 version. The changes affect the was.conf file and may also have an impact on the applications you are running using the Application Server. See "Migrating Applications for the Version 3.02 Level of the IBM WebSphere Application Server" on page 85.

Because we were running the Standard Edition of the IBM WebSphere Application server, we installed WebSphere Application Server Standard Edition Version 3.02 and Java Development Kit Version 1.1.8.

One of the biggest changes in the version 3.02 level of the IBM WebSphere Application Server is that there is no makeserver.sh program or updateproperties program, as there was in the version 1.2 level. (makeserver.sh was a program used to define an Application Server model, which is a directory structure, server_model_root, containing the configuration properties of an individual Application Server. The updateproperties program mapped the changes to the properties files.) This means that all the property values are contained in the was.conf file and they do not get propagated to any other file. Any changes you made to the was.conf file at the version 1.2 level must be re-entered manually into the new version 3.02 level was.conf file. See *WebSphere Application Server Standard Edition Planning, Installing and Using* for more information.

After we made the appropriate changes to the httpd.conf file and made sure our $JAVA_HOME variable was pointing to the correct level of Java (as specified in the installation procedure in the *WebSphere Application Server Standard Edition Planning, Installing and Using* book), we were ready to start the Application Server using the default was.conf file shipped with the product. First we started the IBM HTTP Server, which in turn started the Application Server, just as it did in version 1.2 of the Application Server. Once the Application Server was started we verified our installation and setup using the new Installation and Verification Program. This program contains sample JavaServer Pages (JSPs) and a servlet which shows the server configuration.

We have one experience to report:

The IBM WebSphere Application Server Version 3.02 Conflicts with Our Shared HFS Environment: We found a problem while testing the version 3.02 level of the Application Server as a result of our shared HFS environment. While we were testing the version 3.02 level of the Application Server on one test system, we wanted to keep the version 1.2 level available to our other systems. To do this, we mounted the version 3.02 level at an interim mount point, /was302, and kept the version 1.2 level mounted at the production mount point, /usr/lpp/WebSphere.

After we finished testing the version 3.02 level, we were ready to make it available to all the other systems in our sysplex, so we unmounted the version 1.2 level of the Application Server from the production mount point (/usr/lpp/WebSphere) and mounted the new version 3.02 level there instead. Then we verified that the directives in the httpd.conf file were pointing to the correct directory and started the IBM HTTP Server, which in turn started the Application Server. But when we

attempted to access the JSP samples through the Installation Verification Program, we received the following error messages:

```
Error 500
An error has occurred while processing request: http://<IP Address>/webapp/examples/simpleJSP
Message:Server caught unhandled exception from servlet [simpleJSP]
Target Servlet: simpleJSP
```

It turns out during startup, the version 3.02 level of the Application Server Version creates a directory at /tmp/WebSphere in a system specific HFS. This /tmp/WebSphere directory contains files pointing to the sample JSPs in the Application Server code directory structure in the version HFS. While we were in the testing phase, the file in /tmp/WebSphere pointed to /was302.

The problem was that when we stopped the IBM HTTP Server to re-mount the version 3.02 level at the production mount point (/usr/lpp/WebSphere), the Application Server did not remove the /tmp/WebSphere directory. As a result, when we restarted the IBM HTTP Server, which in turn started the version level 3.02 Application Server from the production mount point, the files in /tmp/WebSphere still pointed to /was302. This resulted in the error shown above when we tried to access the JSP samples.

To work around this problem, we stopped the IBM HTTP Server, deleted the /WebSphere directory from /tmp, and then restarted the IBM HTTP Server. The IBM HTTP Server then started the Application Server, which recreates the /WebSphere directory off of /tmp with pointers to the production directory structure at /usr/lpp/WebSphere. IBM has opened APAR PQ40954 against this problem.

Migrating Applications for the Version 3.02 Level of the IBM WebSphere Application Server

The changes made between the Version 1.2 and 3.02 levels of the IBM WebSphere Application Server will affect applications using the Application Server. Some of these changes include:

- **Moving from Java Database Connectivity (JDBC) 1.0 to 2.0:** JDBC is a Java API that lets servlets access relational databases. The 3.02 version of the Application Server supports the JDBC 2.0 Standard extension API (prior to 3.02, we used JDBC 1.0). The JDBC 2.0 Standard extension API lets applications use datasource objects to describe physical databases or data servers accessed at run time. Among other things, this datasource object lets application implement connection pooling during servlet run time. (See *WebSphere Application Server Standard Edition Planning, Installing, and Using for OS/390* for information on connection pooling.) To use JDBC 2.0, we had to migrate connection manager code to use the JDBC Standard 2.0 Extension APIs.

  Migrating to the JDBC Standard 2.0 Extension APIs also entails moving from a JDBC Type-1 driver to a JDBC Type-2 driver.
- **Moving from JSP version .091 to JSP version 1.0:** The 3.02 version of the Application Server supports JSP version 1.0. At the 1.0 level of JSP, a few JSPs are deprecated, so we had to migrate some of our JSPs to conform to the JSP 1.0 Specification.

For complete information on requirements for the version 3.02 level of the Application Server and migrating applications, see *WebSphere Application Server Standard Edition Planning, Installing, and Using for OS/390*.

In the sections below, we will show some of the changes we made to migrate our applications to the version 3.02 level of the IBM WebSphere Application Server. The

examples we show are based on a sample application called INSCO. You can find the full report describing our INSCO application at the following Web address:

```
http://www.s390.ibm.com/os390/support/os390tst/tips.htm#was
```

We made the following kinds of changes to the INSCO application:

Migrating From a Type-1 to a Type-2 JDBC Driver: We migrated to a Type-2 JDBC driver by applying the fix for APAR PQ36011. Then, we migrated our application to use the Type-2 driver. The example below shows the Type–1 methods of submitting a DB2 update or commit commented out, with the corresponding Type–2 methods shown below them. In the following example, the variable query contains an SQL UPDATE statement:

```
//  stmt.executeQuery(query);
    stmt.executeUpdate(query);
//  adataConn.commit();
    conn.commit();
```

New Method of Getting a JDBC Connection: Before migrating to the version 3.02 of the Application Server, we used a JDBC 1.0 method, IBMJdbcConnSpec, to get a JDBC connection. Now we must use the 2.0 JDBC method, getConnection, to connect to DB2. Below is an example of using getConnection to make a JDBC connection. Note that first we set up the servlet with four APIs and then invoke getConnection:

```
owner    = configBundle.getString("JDBCServlet.dbOwner");
Hashtable parms = new Hashtable();
Context jdbcctx = new InitialContext(parms);
ds = (DataSource)jdbcctx.lookup("jdbc/pool1");
Connection conn = ds.getConnection();
```

After we migrated our servlet to run on the version 3.02 level of the Application Server, we tried to use the examples listed in the *WebSphere Application Server Standard Edition Planning, Installing, and Using for OS/390* book to get a JDBC connection with a userid and password. This function was introduced in the version 1.02 level of the Application Server in the fix for APAR PQ34081. The connection with a userid and password was supposed to look as follows:

```
Connection conn = ds.getConnection(user,password);
```

However, the 3.02 examples containing JDBC connections with userid and password connection did not work. Connecting with userid and password worked the first time, but when the servlet switched the userid, we received the following error:

```
SQL error -551
```

For now, we work around this problem by getting connections without a userid and password, as we show above. We have opened APAR PQ41427 against this problem.

New Method of Closing a JDBC Connection: Prior to level 3.02 of the Application Server, we closed our JDBC connection as follows:

```
acmConn.releaseIBMConnection();
```

Now, with JDBC 2.0 at the version 3.02 level of the Application Server, we commented out the old method and replaced it as follows:

```
//  acmConn.releaseIBMConnection();
conn.close();
```

Removing the Path to com.sun.server.http: At the .091 level of JSP, we accessed the classes in our application using the following call:

```
import com.sun.server.http.*;
```

At the 1.0 level of JSP, we replaced this statement with the following statements that give us access to the classes:

```
// import com.sun.server.http.*;
import com.ibm.db2.jdbc.app.stdext.javax.sql.*;  /
import com.ibm.ejs.dbm.jdbcext.*;                /
```

Note that removing the call to the com.sun.server.http has consequences for applications which use Windows based programs, such as VisualAge for Java, to generate servlets. Windows programs use applets, servlets, and Java beans which contain classes accessed through com.sun.server.http. For example, we use VisualAge for Java to generate our servlets, so after generating our servlets, we will have to make manual changes to the servlets to remove references to the HttpServiceRequest and HttpServiceResponse JSPs so that they'll run with our version 3.02 Application Server. See the list item below for how we did that.

Replacing HttpServiceRequest and HttpServiceResponse JSPs: In the 1.0 level of JSP, the HttpServiceRequest and HttpServiceResponse JSPs, which were generated through com.sun.server.http, were deprecated. Below, we show an example of how we replaced the old HttpServiceRequest JSP: :

```
// ((com.sun.server.http.HttpServiceRequest) req).setAttribute("ab", ab);

        req.setAttribute("ab", ab);
        HttpSession session = req.getSession(true);
```

Our next example shows how we replaced the HttpServiceResponse JSP:

```
// ((com.sun.server.http.HttpServiceResponse) ares).callPage("/jsp/inscohomeconnmgr.jsp", areq);
     RequestDispatcher rd = getServletContext().getRequestDispatcher("/jsp/inscohomeconnmgr.jsp");
     rd.forward(areq, ares);
```

The following example shows the messages we received when we invoked our application with the version 3.02 level of the application server without migrating the deprecated JSPs:

```
176:/u/johndoe/INSCO/racf $ javac -deprecation InscoRacf.java

InscoRacf.java:152: Note: The method void setAttribute(java.lang.String,
ng.Object) in interface com.sun.server.http.HttpServiceRequest has been
depricated.
    ((com.sun.server.http.HttpServiceRequest) req).setAttribute("ab", ab

InscoRacf.java:582: Note: The method void callPage(java.lang.String, jav
et.http.HttpServletRequest) in interface com.sun.server.http.HttpService
 has been deprecated.
         ((com.sun.server.http.HttpServiceResponse) ares).callPage("/jsp
meracf.jsp", areq);
```

Note that we invoked the program with – deprecation, which ensures that we see which JSPs have been deprecated in the error message.

The jst.jar and jsdk.jar Files Don't Show Up in 3.02 Application Server: At the version 1.02 level of the Application Server, we were used to seeing the jst.jar and jsdk.jar files in the /usr/lpp/WebSphere/WebSphere/AppServer/lib subdirectory. These files were referenced in our version level 1.02 /lib subdirectory as follows:

- /usr/lpp/WebSphere/WebSphere/AppServer/lib/jst.jar
- /usr/lpp/WebSphere/WebSphere/AppServer/lib/jsdk.jar

Note that you will no longer see these files in the /lib subdirectory of the version 3.02 level of the Application Server — the files are no longer needed. You must omit references to these files in places like your profile or was.conf file.

Chapter 9. Using the IBM HTTP Server

This chapter describes our experiences with the IBM HTTP Server during the OS/390 R9 and R10 time frames as follows:

- "Using the IBM HTTP Server V5.2"
- "Migrating to The IBM HTTP Server Version 5.3" on page 97
- "Using LDAP to Protect IBM HTTP Server Resources" on page 98

For product information about the IBM HTTP Server, see the Web-based user interface, the *WebSphere Troubleshooter for OS/390*, which includes the most current debugging, and tuning hints and tips for all supported releases of the IBM HTTP Server and the IBM WebSphere Application Server. Look for this at the following URL:

```
http://www.ibm.com/software/webservers/httpservers/troubleshooter.html
```

Using the IBM HTTP Server V5.2

This chapter describes our experiences using the IBM HTTP Server V5.2 with OS/390 Release 9. We had already migrated to the V5.2 level of the HTTP Server with OS/390 R8, as documented in our December 1999 edition.

In this section, we will cover the following topics:

- "Using the HTTP Recovery Directive"
- "Using Scalable Server Subsystem Mode"
- "Using Net.Data to Connect the IBM HTTP Server to DB2" on page 90
- "Migrating to Shared HFS for the IBM HTTP Server" on page 91

For a report on an IBM HTTP Server application, see *Porting an e-business Application to OS/390 Using WebSphere* and *Updating an e-business Application for OS/390 V2R9* in the tips section of our Web site.

Using the HTTP Recovery Directive

The IBM HTTP Server 5.2 for OS/390 R9 includes a new configuration directive, Recovery, which is specified in httpd.conf. This new directive is intended to give you the option of letting the Web server control some or all of the recovery when it detects an abend in its address space. The default for this directive is:

```
Recovery Normal
```

Specifying Normal means that the processing of abends in the Web server address space is taken over by the Web server. Since this is a new directive, the idea of having the Web Server take over recovery was something new to us. Because we wanted to keep doing our own recovery processing, we specified the following for the Recovery directive in httpd.conf:

```
Recovery None
```

Specifying None means that the Web server does not perform any abend recovery.

Using Scalable Server Subsystem Mode

We have been using a scalable server subsystem with our Web Server since OS/390 R4. (See our December 1997 edition for how we set up our scalable server subsystem.) In OS/390 R9, however, we noticed a change in the way queue servers are processed in a scalable server subsystem. Prior to OS/390 R9, we usually had one queue server for each of our application environments. Once we

started using the HTTP server in scalable server subsystem mode in R9, however, we found that WLM starts multiple queue servers for a single application environment. When we spoke to the WLM service team about this processing change, they pointed us to APAR OW36367 as a possible cause of the change.

The increase in the number of queue servers also increases the number of files that the HTTP Server writes to the /tmp HFS. In either standalone or scalable server mode, the HTTP server has always placed two kinds of files into the /tmp HFS:

- *IMWSxxxxxxxxx* where the x's are a combination of upper and lower case characters.
- *yyyyyyyy.POyyyyy.Slyyyyyy.Xyyyyyyy* where the y's are numbers.

In scalable server subsystem mode, the HTTP server creates an instance of the *yyyyyyyy.POyyyyy.Slyyyyyy.Xyyyyyyy* file **every time** a queue server is started. Because there are now more queue servers, we also have more of these files in our /tmp HFS. In addition, these files are never deleted. If our /tmp HFS were to fill up as a result of these files, the HTTP Server would not be able to write to the *yyyyyyyy.POyyyyy.Slyyyyyy.Xyyyyyyy* file, and would therefore be unable to start any queue servers. Eventually application environments will start to shut down as a result. WLM will try three times to start a queue server for an application environment, and then it shuts down the application environment to avoid looping. Once the /tmp directory is full, the HTTP server cannot start any additional queue servers until the space problem is resolved and any application environments restarted. APAR PQ37163 has been opened to address this problem by deleting old queue server files.

We also had a problem with **log** files filling the /tmp HFS, preventing the HTTP Server from writing new queue server files to /tmp and starting new queue servers. This happened because we were directing our log files to the /tmp HFS. To reduce the chances of filling the /tmp directory, we mounted a separate HFS at /tmp/wwwlogs and directed log files to this directory instead and restarted the application environments. In general, we recommend keeping the /tmp HFS as clear as possible to help ensure that the queue servers can run smoothly.

Using Net.Data to Connect the IBM HTTP Server to DB2

Net.Data for OS/390 is a DB2 feature that allows you to write applications to access DB2 and other data through the Web. We initially implemented it as an interface between the IBM HTTP Server and our DB2 data during OS/390 R6. (See our December 1999 edition for details on how we set it up.) This topic describes our experiences using Net.Data 2.1 and 2.2:

Net.Data 2.1 Macros Don't Work Well Running as ICAPIs: While we were still using Net.Data at the 2.1 level, we changed Net.Data so that we run macros as ICAPIs rather than CGIs. This resulted in CEEDUMPs where the exceptions referenced the get_next_row entry in the same traceback section that processed the dtw_icapi entry. When we converted Net.Data back to CGI processing, the problem disappeared. Temporarily, we were able to work around this problem by specifying HEAPPOOLS(ON) in our LEParm statement. (HEAPPOOLS is a keyword that controls memory allocation.) This resolved the problem, but is not a good permanent solution for us. You can only specify a certain number of characters on the LEParm statement, and specifying HEAPPOOLS(ON) uses up 13 characters we needed for other things. We resolved the problem by migrating to Net.Data 2.2. If you encounter this problem, either specify HEAPPOOLS(ON) or migrate to Net.Data 2.2.

Net.Data Macros Cause High CPU Consumption: We noticed that while we were running Net.Data macros, our HTTP Server began consuming an abnormally large amount of CPU resources, which can adversely affect other address spaces running on the OS/390 system image. We resolved this problem by installing the fix for APAR PQ32886.

Net.Data and Net.Commerce Files Share the Same Name: Both Net.Data and Net.Commerce (now called WebSphere Commerce Suite) have a file called d2w.cat. If you run with both products, as we do, this duplication can cause confusion, since the two files may contain different information. We ran into this when we migrated to Net.Data 2.2 and encountered a problem using the SHOWSQL function. SHOWSQL, which is used by the custqord.d2w sample macro, is supposed to display the SQL statement currently being processed. However, when we issued the custqord.d2w macro, we received the following output instead of the SQL statement:

```
Net.Data Error:
```

Although the SQL statement was not there, the custqord.d2w macro generated all it's other HTML successfully.

Our problem was caused by a typo we entered when we specified the /netdata directory in the httpd.envvars file. Because we specified the netdata directory incorrectly, the HTTP server was unable to access a file in the netdata directory, d2w.cat. Instead, Net.Data managed to find the d2w.cat file in the /netcommerce directory.

The incorrect /netdata directory specification in the httpd.envvars file was as follows:

```
/usr/lpp/netdata%L/%N.
```

The statement should read:

```
/usr/lpp/netdata/%L/%N.
```

Note the slash between /netdata and %L.

Once we fixed the typo, we had no trouble with custqord.d2w or SHOWSQL.

Migrating to Shared HFS for the IBM HTTP Server

The OS/390 R9 shared HFS function is a great addition to our Web serving environment because it makes it much easier for us to share products and files between systems. (Please read "Chapter 11. Managing a Hierarchical File System (HFS)" on page 109 for a complete picture of our shared HFS environment.)

To understand the changes we made to our IBM HTTP Server environment for shared HFS, you have to understand how it looked before we implemented shared HFS in the OS/390 R9 timeframe. Shared HFS lets us share HFSs across the sysplex in read-write mode, but before that, we could only share HFSs in read-only mode. That meant that before we implemented shared HFS, we shared only our serviceable product files, because systems did not need write access to them. We had to duplicate our customizable data in HFSs on each system so that the system had write access to the data. This set-up had the following drawbacks:

- We had lots of administrative re-work, since updates to data in a system specific HFS often had to be duplicated on each system.
- We were consuming a lot of DASD space with our large system specific HFSs.
- We had to install some of our related Web serving products (such as Java) individually on each system because they are not elements of OS/390 and so are

not included in the OS/390 install image. This meant we had to install and service these products individually on each system.

In the shared HFS environment, our serviceable IBM HTTP Server files are still shared read-only across the sysplex. Now however, we can also share some of our customizable data with read-write access. We separated our customizable data as follows:

- We keep shared customizable data (application data and environmental variables, for example) in an HFS that is shared read-write by all the IBM HTTP Servers in the sysplex. We mount this HFS at mount point /http_shared.
- We keep customizable data that is unique to a system in the system specific HFSs at /dgw_pet on each system. Note that because this is a system specific HFS, we're actually accessing /dgw_pet through the following symbolic link in the sysplex root:

```
dgw_pet -> $SYSNAME/dgw_pet
```

The following picture shows how we split our customizable data between our new shared HFS and our system specific HFSs:

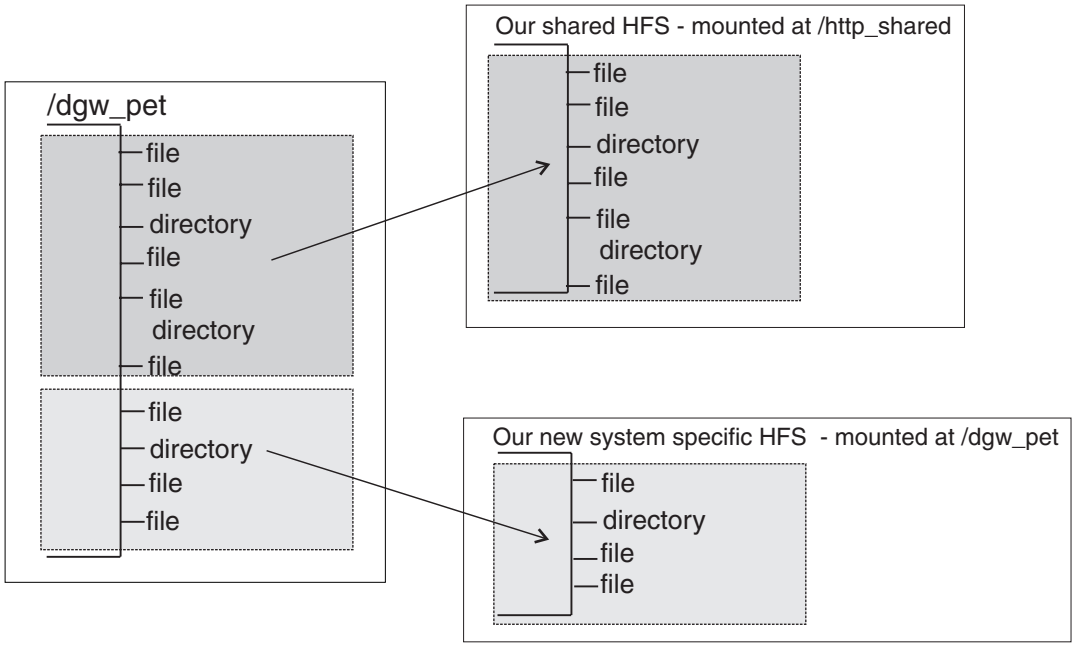


Figure 17. Enabling Shared HFS for the IBM HTTP Server

There are a bunch of advantages for us in using the shared HFS for the IBM HTTP Server:

- We can update shared customizable data just once, for the whole sysplex.
- Because we're consolidating most of the customizable data from our system specific HFSs into our shared HFS, we need much less DASD space for the system specific HFSs
- The shared HFS function lets us install our non-OS/390 Web serving products (such as Java, FastCGI, and Net.Data) just once, into HFSs that are shared across the sysplex. This reduces the amount of maintenance, since service can be performed on the shared HFS rather than each system.

This section covers how we separated and stored our customizable data between our shared read-write HFS mounted at /http_shared and the system specific HFSs.

Setting up Shared HFS for the IBM HTTP Server

In order to migrate our IBM HTTP Server to a shared HFS environment, we did the following:

Consolidating Customizable Data Into Our Shared HFS: We consolidated all of the customizable data that we wanted to share from the system specific HFSs on each system into our shared HFS. We mount our shared HFS at mount point /http_shared and it is shared read-write by all the IBM HTTP Servers in the sysplex. We have the following kinds of files in our shared HFS:

- .gif
- .jpeg
- .cgi
- net.data macros
- servlets
- .jsp
- .jni

We also placed our httpd.envvars file in our shared read-write HFS. This works for us because our httpd.envvars file is fairly simple and can be used by all the OS/390 system images in our sysplex. We also plan to move our httpd.conf file to the shared HFS, after we simplify and standardize it enough to be used by all our systems.

Updating the httpd.conf File for Shared HFS: In order to enable the IBM HTTP Server for shared HFS, we had to update the httpd.conf file for each of our systems as follows:

- We updated the document root pass directive in our httpd.conf files to point to our new shared HFS mounted at /http_shared.
- Added a new pass directive for the IBM HTTP Server .html files. Prior to R9, we had our IBM HTTP Server .html files right in the document root directory, /dgw_pet, with all the other customizable data. While we were separating files for the switch to shared HFS, however, we decided to take the opportunity to group files by type in their own directory. This meant we wanted a directory called /html in the /http_shared directory. There were two ways we could have ensured that users were able to access the .html pages in this new /html directory:
  - We could re-code every URL that requested an .html page in our files to include a direction to the new /html directory.
  - Add a Pass directive that would direct all .html requests to the files in the /html directory.

  We chose the second method because it was faster. Note that the Pass directive for this new .html directory must be the last Pass directive before the document root Pass directive. (The document root Pass directive is the last one). Because the /*.html Pass directive is so general, it has to be next-to-last to allow requests for .html files to resolve in other Pass directives before reverting to ours.

Our updated directives in the http.conf files are as follows:

```
Pass     /*.html                  /http_shared/html
#Pass    /*                       /dgw_pet/*
Pass     /*                       /http_shared/*
```

We have commented out the Pass directive that specifies /dgw_pet as our document root directory in our http.conf file, substituting the new Pass directive for our new shared read-write HFS directory, /http_shared, as our document root directory.

Using the IBM HTTP Server

Since every environment is unique, you'll have to evaluate your own needs when deciding how to update the Pass directives in your own httpd.conf file when you migrate to shared HFS for the IBM HTTP Server.

Adding Our Home Page File to Our Shared HFS: When we decided to move all of our .html files into the /http_shared/html directory, one of those we moved was our home page file, Welcome.html. This was a mistake - the home page file must be in the document root directory (/http_shared) or else users can only access our home page by specifically requesting it, by issuing, for example http://Z0/Welcome.html. After we put the Welcome.html file in our document root directory, users were able to access our home page by default, by specifying http://Z0 for example. Note that we had already defined our default home page in a Welcome directive in our http.conf file.

Updating Our Start-up Procedure: We had to update our IBM HTTP Server start-up procedure, IMWEBSRV, so that it pointed to the httpd.envvars in its new location in the shared HFS.

Service for Migration to Shared HFS: As we ran our workloads against our shared HFS, we found that our HFS became filled to 100% capacity. Our HFS was filling to 100% capacity because the workloads are very read-intensive. Each time data is read from an HFS, a small record is generated, taking up space in the HFS. These records remain until the next time someone writes to the HFS, at which time they are eliminated as part of the write-generated resynch activity. Because our workloads do not write to the shared HFS, it filled to 100% capacity. You might see this problem if you have an HFS that you don't write to very often.

DFSMS APAR OW43771 and UNIX System Services APAR OW44204 have been opened to address this problem.

Retrieving Unneeded Space in System Specific HFSs: Because we were able to consolidate much of the data that had been duplicated on each system in system specific HFSs to one shared HFS for the IBM HTTP Server, we reduced the amount of DASD space needed for system specific HFSs. We wanted to reclaim this space, but since you cannot reduce the size of an HFS dynamically, we did the following to create new, smaller HFSs for our system specific data:

1. Before we did anything else, we archived our original system specific HFSs (mounted at /dgw_pet on each system). To do this we used the UNIX System Services tar command to consolidate all the customizable files from the /dgw_pet HFS from each system into one big file, which we archived in either another HFS or an MVS data set. This step corresponds to **1** in Figure 18 on page 95.
2. Next, we copied the subset of the files that we wanted to share from the /dgw_pet directory into our new shared HFS, which is mounted at /http_shared. This corresponds to **2** in Figure 18 on page 95.

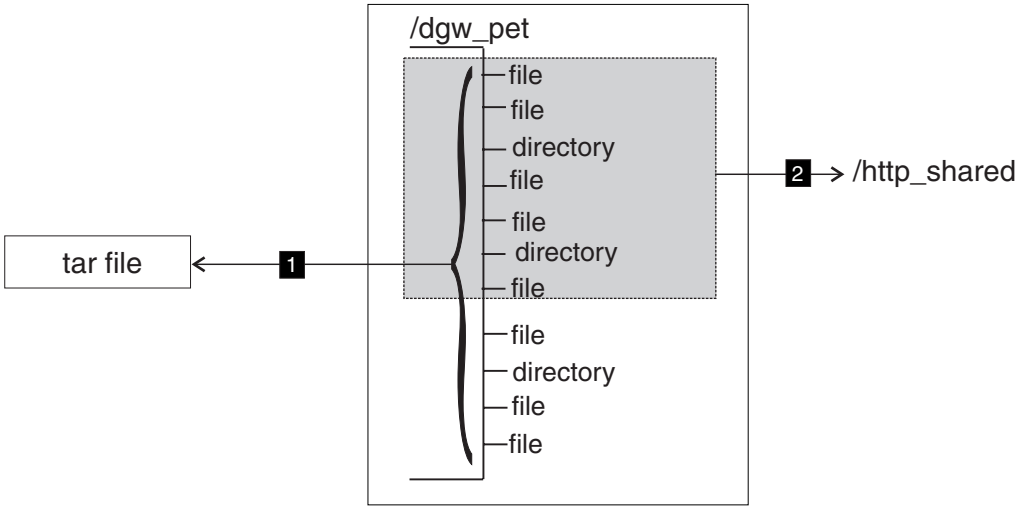


Figure 18. Reclaiming Space from a Too-Large HFS: Steps 1 and 2

3. On each system, we then deleted the files we just copied from the /dgw_pet directory to the shared HFS. This corresponds to **3** in Figure 19 on page 96.

   Once we deleted these files, we had reduced the amount of information in our system specific HFSs mounted at /dgw_pet. But our system specific HFS data sets remained the same size. Because you cannot reduce the size of an HFS data set dynamically, the next set of steps show how we created smaller system specific HFSs and deleted the old, too-large /dgw_pet HFSs.
4. Next we created a new, smaller system specific HFS to hold the system specific data left in /dgw_pet after we took out the data we wanted to share across the sysplex. To size our new system specific HFS, we looked at the percentage of space used in the current HFS, and added a bit more to allow for growth. For example, if your current HFS is 100 cylinders and only 15% is being used, we size it at 15 cylinders plus 5 for growth, for a total of 20 cylinders. We don't add on too much for growth - the system will automatically enlarge the HFS as it fills to capacity, or we can easily enlarge it dynamically.

   We named the new HFS data set hlq.$SYSNAME..tmp_dgw_pet.llq. We created a temporary directory, /tmp/dgw_pet, and mounted our new, trimmed down system specific HFS there. This step corresponds to **4** in Figure 19 on page 96.
5. Next, we copied the files and directories left from the /dgw_pet directory into the new HFS. We mounted the new HFS at /tmp/dgw_pet. This corresponds to **5** in Figure 19 on page 96.

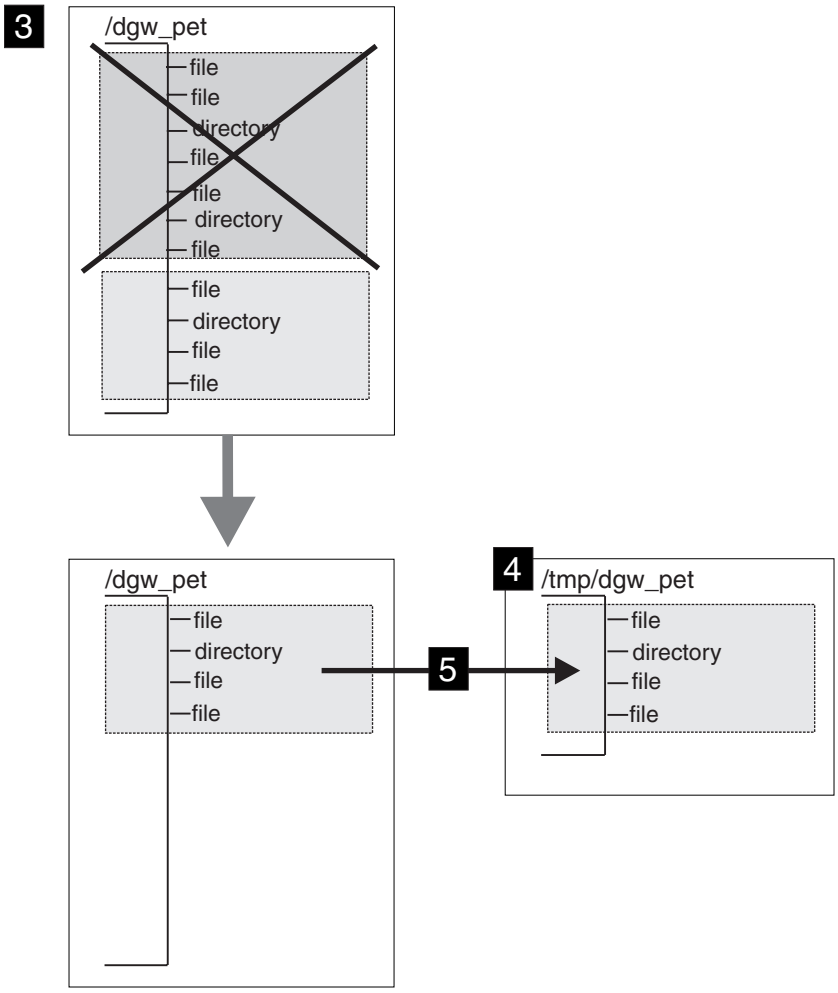


Figure 19. Reclaiming Space from a Too-Large HFS: Steps 3, 4, and 5

6. Next, we wanted to mount our new system specific HFS at the old /dgw_pet mount point. The first step in this process was to unmount the HFSs mounted at /dgw_pet and /tmp/dgw_pet.
7. We then renamed the two HFS data sets as follows:
   - We renamed our original too-large system specific HFS data set from

     `hlq.$SYSNAME..dgw_pet.llq`

     to

     `hlq.$SYSNAME.archive_dgw_pet.llq`
   - We renamed our new smaller system specific HFS data set from

     `hlq.$SYSNAME..tmp_dgw_pet.llq`

     to

     `hlq.$SYSNAME..dgw_pet.llq`

     We gave this HFS the same data set name as our original, large system specific HFS so that we did not have to update the BPXPRMxx parmlib member with a new HFS name in the mount statement.
8. We mounted our new, smaller system specific HFS at /dgw_pet.
9. We then restarted the IBM HTTP Server and verified that the new HFS was working.

10. Finally, when we were sure that everything was working the way it should, we deleted the tar file and the old, too large system specific HFS. Now we have regained much of the DASD space we had been consuming with our old, large, system specific HFSs.

Migrating to The IBM HTTP Server Version 5.3

This section describes our migration from the IBM HTTP Server version 5.2 to version 5.3. We migrated in the OS/390 R10 timeframe using the *HTTP Server Planning, Installing, and Using V5.3 for OS/390* book. For the most recent edition of this book, see the following Web site:

```
http://www.ibm.com/software/webservers/httpservers/doc53.html
```

Migration went smoothly. The major change in version 5.3 of the IBM HTTP Server was migrating to System SSL, see “Migrating to System SSL Support”.

We have the following experience to share:

JESMSGLG Log Now Shows Authorization of IBM HTTP Server Files: In our December ’99 edition, we wrote about how we ensured the correct authorization of files in the IBM HTTP Server directory. Authorization for files in the IBM HTTP Server directories are controlled by program control bits. Having the program control bit set on for an executable program file (load module) is the equivalent of having an OS/390 Security Server (RACF) RDEFINE command issued against the load module to the PROGRAM class. However, it can be difficult to find files that do not have their program control bit set on in order to fix them. (We use the extattr command to restore the correct program control bits.) Now, however, with OS/390 R10, the JES JESMSGLG log shows any files that do not have their program control bits set on. For information on this enhancement, see *OS/390 SecureWay Security Server RACF Migration*.

The following shows an example of the message you’ll see in the JESMSGLG log for a file (libadpter.so) that is missing its program control bit:

```
BPXP015I HFS PROGRAM /usr/lpp/WebSphere/AppServer/lib/libadpter.so IS NOT MARKED PROGRAM CONTROLLED.
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.
```

We use the extattr +p <*filename*> command to turn on the program control bit for a file.

Migrating to System SSL Support

System SSL is part of the Cryptographic Services base element of OS/390. It is a set of SSL interfaces available for use by any product that needs SSL support, which means they no longer need to write or ship their own SSL code. In the past, the IBM HTTP Server (version 5.2 and earlier), wrote and shipped SSL code as part of their product. Now, however, version 5.3 of the IBM HTTP Server uses System SSL for its SSL support.

Our existing SSL environment worked well with System SSL and it took very little work to migrate. (See our December ’98 and December ’99 editions for our existing SSL environment.) We have the following experiences to share:

You Must Add hlq.SGSKLOAD File to the linklist concatenation or STEPLIB: When we tried to make an SSL connection through the IBM HTTP Server we received a message on the browser saying that the server could not be found. We turned on tracing in the IBM HTTP Server procedure and found that the GSKSSL

dynamic link library (DLL) could not be loaded. It turns out that we missed a step documented in the *OS/390 System Secure Sockets Layer Programming Guide and Reference*. In this manual, it states that file *hlq*.SGSKLOAD must be added to either a STEPLIB statement in our IBM HTTP Server procedure or to our linklist concatenation. Initially, we added *hlq*.SGSKLOAD to a STEPLIB statement to see if this would resolve our problem, which it did. Our permanent solution was to add the file to our linklist concatenation.

ikeyman and gskkyman are Very Compatible: Before version 5.3 of the IBM HTTP Server, we used the IBM key management utility (ikeyman) to manage our key databases and act as our own certificate authority (CA). With version 5.3, ikeyman has been replaced by the System SSL gskkyman utility. Since the look and feel of gskkyman is exactly like that of ikeyman, the switch to gskkyman was painless. We found that ikeyman and gskkyman were very compatible; we had no problem manipulating data generated by ikeyman with gskkyman. We used gskkyman to modify and maintain key databases that we had created with ikeyman and we were able to use key databases generated with ikeyman as specified keyfiles in the httpd.conf file in our new System SSL environment.

For examples of using gskkyman, see *HTTP Server Planning, Installing, and Using V5.3 for OS/390* or the following Web site:

`http://www.ibm.com/software/webservers/httpservers/gskkyman.htm`

Update for client Certificate Processing: In our testing of the IBM HTTP Server during the OS/390 R8 and R9 timeframe, we found that we had trouble getting our client certificates to work while the application server was up and running – we would get a CEEDUMP with an exception in Base64Dearmor. To get around this, we used the workaround specified in APAR PQ32514. If you are running on OS/390 R8 or R9, see APAR PQ32514 and make the recommended changes to the was.conf file. Now, however, the 5.3 version of the IBM HTTP Server uses System SSL which resolves this problem; this workaround is no longer required. Client certificates work fine with System SSL.

Using LDAP to Protect IBM HTTP Server Resources

In this section, we'll describe how we tested using Lightweight Directory Access Protocol (LDAP) to protect resources (such as a Web page) within the IBM HTTP Server. We also used the TCP transport protocol to do this instead of the SSL transport protocol. "Chapter 13. Migrating the OS/390 Security Server LDAP Server to OS/390 R10" on page 141 describes how we set up our OS/390 LDAP Server configuration. We used the following examples shipped with the code to test LDAP protection of an IBM HTTP Server resource:

- httpd.conf — The IBM HTTP Server provides a sample configuration file which is shipped in /usr/lpp/internet/samples/config/C/httpd.conf.
- LDAP sample server — We used the data loaded in the LDAP sample server shipped in /usr/lpp/ldap/examples/sample_server/sample.ldif . The LDAP sample server includes one groupOfNames objectclass, ″Bowling team″. In addition, we added our own groupOfNames objectclass called ″Insco″.

In this section, we'll show how we customized the examples for our environment to protect an IBM HTTP Server resource, a Web page, by requiring name and password verification before displaying the resource. When a user requests the protected resource, they will be prompted for their name and password in a pop-up window.

Customizing the IBM HTTP Server Configuration File for LDAP Protection

The example below shows our httpd.conf after it has been customized for LDAP protection. In the sections below the example, we'll explain each of the highlighted directives that we changed:

```
LDAPInfo Acme {
  Host <IP Address>
 Transport TCP
 ClientAuthType Basic
 ServerAuthType Basic
  ServerDN "cn=LDAP Administrator, o=Your Company"
  ServerPasswordStashFile </directory name>/LDAPStash
  UserSearchBase "o=Your Company"
  UserNameFilter "(&(objectclass=organizationalPerson)(cn=%v1 %v2))"
  GroupNameFilter "(objectclass=groupOfNames)"
}

 Protection Acme {
        ServerId        Acme
        UserId          %%SERVER%%
        PasswdFile      "%%LDAP:Acme%%"
        GroupFile       "%%LDAP:Acme%%"
         Mask             "cn=Bowling team, ou=Groups, o=Your Company"
}
#
 Protect  /LDAPsamp.html Acme
```

Changes to the `LDAPInfo` ***directive:*** We made the following changes within the LDAPInfo directive:

- `Host` : We changed the host IP address to specify the LDAP server we want to access.
- `ServerDN` : We changed ServerDN to specify the data in OS/390 LDAP Server slapd.conf file for the adminDN directive.
- `ServerPasswordStashFile` : In the ServerPasswordStashFile, we specified the location and name of the stash file created. We created the stash file using the htadm command - See *HTTP Server Planning, Installing, and Using V5.3 for OS/390* for information on the htadm command.
- `UserSearchBase` : We changed UserSearchBase to the value specified in the OS/390 LDAP Server slapd.conf file for the suffix directive.
- `UserNameFilter` : We added the UserNameFilter directive. This directive specifies a pattern for the way users will enter their user name when they are prompted in order to access the protected resource. The name that the user enters will be used as a search filter to find the LDAP entry for the user. Our UserNameFilter is ″cn=%v1 %v2″, which indicates that the user name entered must contain two words, %v1 and %v2. The important point here is that the UserNameFilter **must match** the way user names have been specified in the LDAP database entries. In our LDAP database, user names contain two words (a first and last name, for example), so we matched it with a two word UserNameFilter of ″cn=%v1 %v2″. If a user named Eddie Catu tries to access the protected resource, he types in ″Eddie Catu″ when prompted for his user name, which matches both the UserNameFilter and an entry in the LDAP database. ″Eddie Catu″ is authenticated and can access the protected resource.

  If you mismatch your UserNameFilter and the way user names are specified in the LDAP database, users will not be able to access the protected resource because they will fail authentication. For example, lets say we use our two word UserNameFilter, ″cn=%v1 %v2″, but have defined a one word user name in the LDAP database, Dave. A user entering ″Dave″ as their user name will fail authentication and cannot access the protected resource, **even though** ″Dave″ is

defined in the database. You'd need a UserNameFilter of ″cn=%v1″ for a one word user name. There are other ways you can mismatch your UserNameFilter and LDAP database, (see *HTTP Server Planning, Installing, and Using V5.3 for OS/390*). But you can easily avoid the problem by picking a UserNameFilter that matches your database.

- `GroupNameFilter` : We added the GroupNameFilter directive.

Changes to the `Protection` ***Directive:*** In the Protection directive, we modified the `Mask` directive for LDAP protection of our IBM HTTP Server. To determine the value or values we wanted to specify for this directive, you can use the ldapsearch command as follows:

```
ldapsearch -h <IP Address>  -b "o=Your Company"  "(objectclass=groupofNames)"
```

Following is the output we got from this search, showing our two groupOfNames objectclasses, Insco and Bowling team:

```
cn=Clients, ou=Insco, o=Your Company
objectclass=groupOfNames
objectclass=TOP
cn=Clients
description=Clients in Insco in ICE
owner=cn=Insco Manager, ou=Insco, o=Your Company
member=cn=c00001, ou=Insco , o=Your Company
member=cn=c00002, ou=Insco , o=Your Company
member=cn=c00003, ou=Insco , o=Your Company
member=cn=c00004, ou=Insco , o=Your Company
member=cn=c00005, ou=Insco , o=Your Company

cn=Bowling team, ou=Groups, o=Your Company
objectclass=top
objectclass=groupOfNames
description=IBM Home Town Bowling Team
cn=Bowling team
owner=cn=Mary Burnnet, ou=Widget Division, ou=Home Town, o=Your Company
member=cn=Mary Burnnet, ou=Widget Division, ou=Home Town, o=Your Company
member=cn=Michael Campbell+postalcode=4609, ou=Widget Division, ou=Home Town, o=Your Company
member=cn=Eddie Catu, ou=In Flight Systems, ou=Home Town, o=Your Company
member=cn=Melinda Charles, ou=In Flight Systems, ou=Home Town, o=Your Company
member=cn=Al Edwards, ou=Widget Division, ou=Home Town, o=Your Company
```

The value we use for the Mask directive is the line starting with **cn=** above the line starting with **objectclass=groupOfNames**. In this case, we want to give our bowling team access to the protected resource, so we use **cn=Bowling team, ou=Groups, o=Your Company** as the value for the Mask directive.

Customizing the LDAP Server for LDAP Protection

To customize the LDAP server for protection of an IBM HTTP Server resource, we customized the data loaded in the LDAP sample server shipped in /usr/lpp/ldap/examples//sample_server. Note that with OS/390 R10, the sample LDAP server is shipped with the new DB2 backend database, TDBM. (See “Chapter 13. Migrating the OS/390 Security Server LDAP Server to OS/390 R10” on page 141 for information about the TDBM backend database.)

We had to update the LDAP sample server to add the userpassword attribute for each person that we wanted to have access to the protected resource in the IBM HTTP Server. For example, in order to add the userpassword attribute for the people on our Bowling team (see the members displayed in the ldapsearch output above) who need access to the protected resource, we did the following **for each user**:

1. Let's say we're giving user Eddie Catu access to the protected resource. We create a file that we call `ldapeddie` containing the following:

```
cn=Eddie Catu, ou=In Flight Systems, ou=Home Town, o=Your Company
+userpassword=Eddie's_password
```

2. We issued the following ldapmodify command to add the ldapeddie file containing the userpassword attribute for Eddie Catu to our LDAP database:

```
ldapmodify -h <IP Address>  -D "cn=LDAP Administrator, o=Your Company" -w secret  -f ldapeddie
```

Verifying LDAP Protection for The IBM HTTP Server

To verify that you've set up LDAP protection for our IBM HTTP Server, simply try to access the protected IBM HTTP Server resource from your browser:

```
http://<your.IP.address> /LDAPsamp.html
```

When you request your protected site, the browser should display a box prompting you for a user id and password. For example, our friend Eddie Catu would enter:

```
User Name:  Eddie Catu
Password:   Eddie's_password
```

Now, the browser should display the protected resource.

Chapter 10. Using OS/390 Infoprint Server

In this chapter, we describe our strategy for network printing on the OS/390 platform. We originally set up our printing environment in OS/390 R6 using OS/390 Print Server. In OS/390 R7, we incorporated PSF for OS/390 into our configuration. Our environment consists of a TCP/IP-connected IBM Network Printer 24 (4324) which is shared by both Infoprint Server and PSF. See our December 1999 edition for details about our initial setup of both of these environments.

In this edition, we discuss the following topics:
- An overview of our Infoprint Server configuration
- Migrating from OS/390 Print Server to Infoprint Server
- Migrating PSF for OS/390 to use Infoprint Server's AOP Printer Inventory
- Using Infoprint Server Transforms

We used the following publications to help us with our setup:
- *OS/390 Infoprint Server Migration*
- *OS/390 Infoprint Server Customization*
- *OS/390 Infoprint Server Operation and Administration*
- *OS/390 Infoprint Server Introduction*
- *OS/390 Infoprint Server User's Guide*

We also found lots of useful information at the following Web site:

`http://www.ibm.com/printers/R5PSC.NSF/Web/ipservhome`

Once at the Web site, check out the printing scenarios. They are great for giving you a high-level perspective on how to accomplish various printing tasks.

Our Infoprint Server Configuration

The following figure illustrates our Print Server configuration. An important point to note is that print requests from workstations (including OS/2, Windows 95 or Windows NT with the port monitor, and AIX) as well as print requests from OS/390 UNIX must go through the Print Interface component of the Print Server to get to the JES2 or JES3 spool. Print requests from MVS batch or TSO go directly to the JES2 or JES3 spool without having to go through the Print Interface.

Infoprint Server

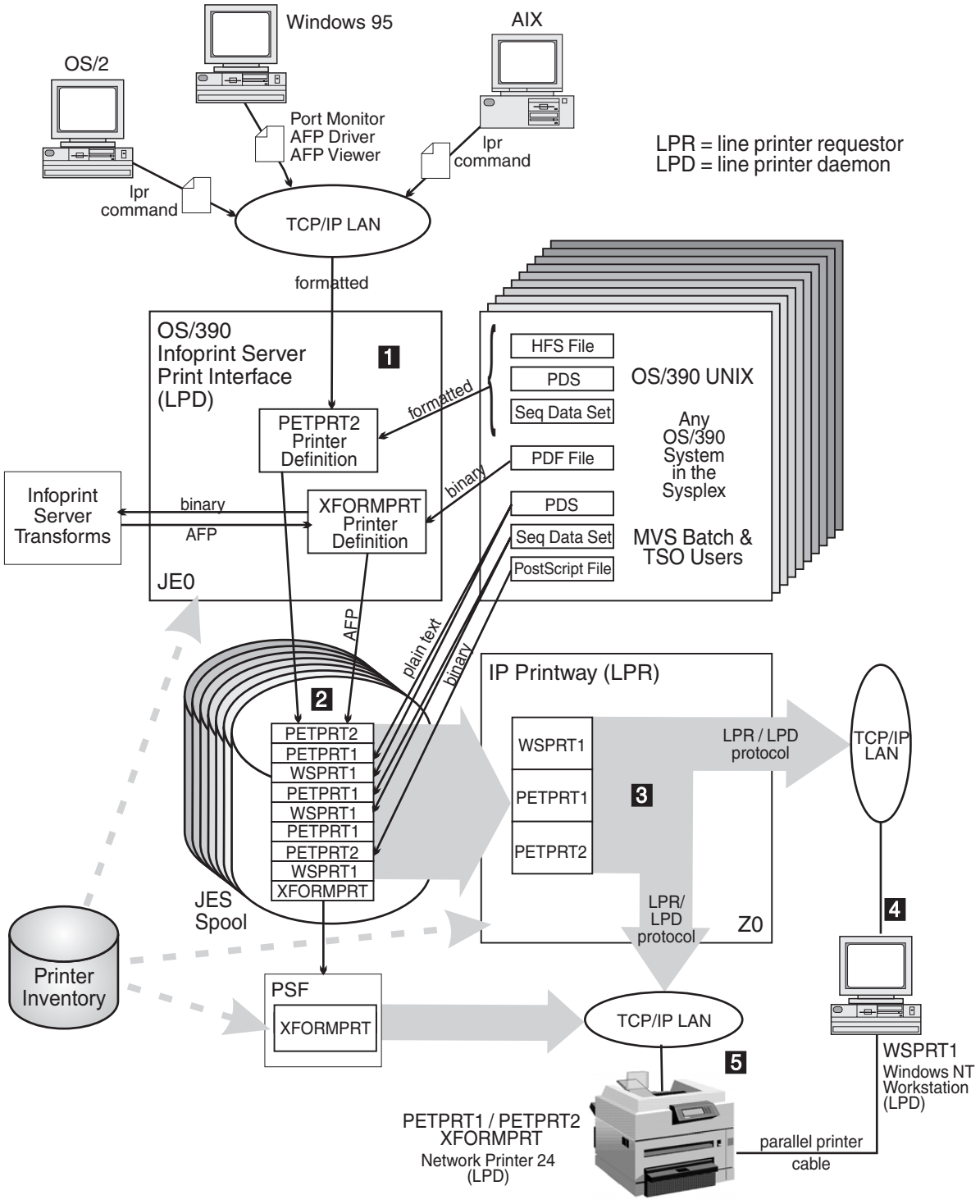


Figure 20. Our Print Server Configuration

Note the following about the above figure:

- Print requests from the following sources come into the Print Interface (1), which is a line printer daemon (LPD) running on OS/390 system JE0. (The print requests are directed to the IP address of the Print Interface):
  - OS/2 workstation (TCP/IP attached)—using the TCP/IP lpr command.

  - Windows 95 workstation (TCP/IP attached)—using standard print submission features available on Windows 95 applications, and using port monitor, Advanced Function Presentation (AFP) driver, and AFP viewer code that is supplied with OS/390 Infoprint Server. Note that the Windows 95 workstation cannot query the status of print requests nor can it cancel print requests.

    The port monitor monitors the LPT port on the workstation. An LPT port would normally have a directly-attached printer. In this case, the port monitor collects the data that is directed to the LPT port and sends it to the network.

    The AFP viewer allows us to view print files of type AFP on the workstation and then send them to the printer.
  - AIX workstation (TCP/IP attached)—using the TCP/IP **lpr** command.
  - OS/390 UNIX (running on any OS/390 system in the sysplex)—using the **lp** command. We can print HFS files, PDSs, and sequential data sets, and can also use the **lpstat** and **cancel** commands to query status and cancel print requests from OS/390 UNIX. See *OS/390 Infoprint Server User's Guide* for information about the **lp**, **lpstat**, and **cancel** commands used by the OS/390 Print Server.

- The Print Interface allocates data sets for the print files on the JES2 or JES3 spool (**2**). MVS batch and TSO users (running on any OS/390 system in the sysplex) send print requests directly to the JES2 or JES3 spool.

  The Print Interface uses a set of files called the Printer Inventory, which contain printer definitions created by a system administrator. Each printer definition contains JES allocation parameters (for example, class, destination, and IP address) that the Print Interface uses when allocating data sets on the JES spool.

- IP PrintWay then takes any print files off the JES2 or JES3 spool that were allocated on the spool with the JES work-selection attributes (for example, CLASS) defined for the IP Printway functional subsystem application (FSA) and distributes them (**3**). IP PrintWay distributes the files either to the IP address of the Windows NT workstation, or the IP address of the 4324 network printer, using the LPR/LPD protocol (a standard of the TCP/IP protocol). Both the printer and the workstation have LPD code running on them to process the print requests.

  Notes:

  1. The IP Printway functional subsystem (FSS) is the Printway address space created by JES; the FSA is an area within the FSS that acquires data sets from the JES spool (based on information that JES gives the FSA) for transmission to the target system.
  2. IP PrintWay no longer uses a routing file; it uses the Printer Inventory. For jobs that come in through Print Interface, IP Printway gets the routing information for the named printer from the printer definition in the Printer Inventory. For jobs that do not come through Print Interface (such as from MVS batch and TSO users) and that do not specify a printer IP address in the JCL, IP Printway uses the jobs' class, destination, and forms attributes to find a matching printer definition in the Printer Inventory and gets the routing information from that printer definition.
  3. When IP PrintWay is on a JES3 system, related JES3 commands must be issued from the JES3 global.
  4. For AFP output (such as files processed with Infoprint Server Transforms), PSF (instead of IP Printway) selects the output data sets from the JES spool and routes them to the printer.

- The print server (that is, LPD on the Windows NT workstation) receives print requests (**4**) and sends them to a printer.

- The printer receives print requests (**5**) either directly from IP PrintWay, from the Windows NT workstation, or from PSF and prints them.

Experiences With Infoprint Server

In this section, we describe some of our experiences with OS/390 Infoprint Server.

Migrating to Infoprint Server

During our OS/390 R8 testing, we had the opportunity to migrate our OS/390 Print Server environment to the new Infoprint Server environment. We followed the information in the program directory, *OS/390 Infoprint Server Migration*, and *OS/390 Infoprint Server Customization*, and our migration went very smoothly. *OS/390 Infoprint Server Migration* directed us to all the sample JCL and UNIX commands we needed to successfully migrate to the new version of the product.

However, there was one thing we needed to know which didn't appear to be mentioned anywhere. In order to successfully issue the UNIX commands (**aopmig** and **pidu**), you need to ensure the following:
- you must be a superuser
- the aopd daemon must be running (by issuing the AOPSTART command)

HFS Customization: In OS/390 R8, we ran with a read-only root HFS (which, in OS/390 R9, is now the read-only version HFS). Because of this, we had to remove the HFS setup steps from the AOPSETUP shell script and put them into our CUSTHFS EXEC so that they run at our service time. For more information about our HFS environment and service strategy, see "Chapter 11. Managing a Hierarchical File System (HFS)" on page 109.

Migrating PSF to Use the Printer Inventory

In OS/390 R8, the option became available to allow PSF for OS/390 to use the Infoprint Server Printer Inventory to control all PSF printers. We chose to implement this option because it is required in order to use Infoprint Server Transforms—which we were very interested in doing.

The migration of PSF to use the AOP Printer Inventory consisted of two parts:

1. We changed the PSF proclib member APSWPROT to include the following statement:

   ```
   //STEP01   EXEC PGM=APSPPIEP,REGION=4096K,PARM=('INV=AOP1')
   ```

   In this statement, *AOP1* is the name of the installation's printer inventory, which is defined in the aopd.conf file. AOP1 is the default name.
2. We defined our PSF-controlled printers in our Infoprint Server Printer Inventory. You can do this either by using the **aopmig** and **pidu** (printer inventory definition utility) commands or by using the Infoprint Server ISPF panels. Both methods are fully described in *PSF for OS/390: Customization* and *OS/390 Infoprint Server Operation and Administration*.

Setting Up and Using Infoprint Server Transforms

Infoprint Server Transforms for OS/390 is an optional, separately orderable IBM program product (5697-F51) that works with Infoprint Server. Its purpose is to convert various printer data stream formats to AFP format for printing on IBM AFP printers. Infoprint Server Transforms can convert the following formats to AFP:
- PCL
- PDF

- PostScript
- SAP ABAP
- SAP OTF

Our concentration was on the PDF to AFP transform that became as new function in APAR OW40194 (PTF UW66117) in December, 1999. Our goal was to install the transforms and print IBM publications in PDF format from OS/390 to OS/390-attached printers.

Installing Infoprint Server Transforms: We used the Infoprint Server Transforms *Program Directory* to install the base program product and then we installed the PTF for the PDF and PostScript transforms. After that, we followed the instructions in the Infoprint Server Transforms chapter in *OS/390 Infoprint Server Customization* to enable and customize our transforms environment. We used all the sample files and all default values for variables to customize our environment.

Adding a Transform Printer: Once we had the transforms installed, customized, and started, we needed to add a transform printer to the Printer Inventory. You can decide how many printer definitions you want to add. We chose to define one printer that could handle all transform types. To do this, we used the Infoprint Server ISPF panels to define a new printer called XFORMPRT.

Defining a new transform printer is very similar to defining other printers. First we defined the new printer, giving it a name and other descriptive information:

```
Edit                    PSF for OS/390 Printer Definition
Command ==>

Printer definition name .  XFORMPRT
Description . AFP Printer for Xforms
Location. . . Networking Lab

                          Component name        Custom values
Section                   (enter to list)       (enter to customize)
Allocation            =>                          => *
Processing            =>                          => *
```

Figure 21. Adding the XFORMPRT Printer Definition

We then customized some of the allocation values for the printer, such as the CLASS, DEST, and FORMS:

```
Printer definition name . XFORMPRT


Spool allocation values:
   CLASS . . . . P               LINECT. . .
   DEST. . . . . PRT999          PRMODE. . .
   JES node. . .                 PRTY. . . .
   FCB . . . . .                 SEGMENT . .
   FLASH count .                 THRESHLD. .
   FLASH name. .                 UCS . . . .
   FORMS . . . . STD             WRITER. . .
   GROUPID . . .
```

Figure 22. XFORMPRT Allocation Values

Finally, we customized the processing options. This is where we specified the filters to invoke the appropriate transforms for the various data types that this printer will support:

```
Printer definition name . XFORMPRT


Document code page . .
Printer code page. . . IBM-1047

Print Interface supported data formats and associated filters:
Data format:  Filter:
/ Line data
/ MO:DCA-P
/ PostScript   ps2afp.dll %filter-options
/ Text         aopfiltr.so
/ PCL          pcl2afp.dll %filter-options
/ PDF          ps2afp.dll %filter-options
```

Figure 23. XFORMPRT Allocation Values

Sample Print JCL: Here is a sample of the JCL statements we use to print a PDF file residing in an OS/390 data set to our transform printer:

```
//USER1    JOB 1,'print'
//PRINT    EXEC AOPPRINT,PRINTER='XFORMPRT'
//SYSIN    DD DSN=MY.PDF.FILE,DISP=SHR
```

Transforming Before Printing: As described in the Infoprint Server publications, Infoprint Server also provides shell commands to allow OS/390 UNIX users to invoke the transforms prior to printing. To test the **pdf2afp** command, which transforms a PDF file to AFP, we issued the following command:

```
pdf2afp -o myfile.afp myfile.pdf
```

This command takes myfile.pdf as input, invokes the appropriate transform, and produces myfile.afp as its output. We then successfully printed the AFP file by issuing the command:

```
lp -d xformprt myfile.afp
```

Improving Print Performance: You can optimize print performance by specifying the proper printer resolution on your transforms. The resolution should match the capability of the printer you intend to use.

For our 4324, the printer resolution is 300 pels. We specify the resolution on the transform command, as follows:

```
pdf2afp -r 300 -o myfile.afp myfile.pdf | lp -d xformprt
```

In the above example, we transform and print by piping the output of the transform command to the **lp** command.

OS/390 Infoprint Server User's Guide explains more about this and provides some guidance for selecting the appropriate resolution for various printers.

Chapter 11. Managing a Hierarchical File System (HFS)

OS/390 UNIX System Services (OS/390 UNIX) files are organized in a hierarchical file system (HFS). Starting with OS/390 V1R5, you are required to run full-function OS/390 UNIX using an HFS if you are using eNetwork Communications Server. In our December 1999 edition, we discussed our strategy as of OS/390 R8. In this chapter, we will cover the following topics:

- "Highlights of Our Strategy"
- "Migrating to a Shared Hierarchical File System (HFS)"
- "Our HFS Service Environment" on page 126
- "HFS Naming Conventions" on page 130
- "Our Post-Install Customization EXEC" on page 131
- "Using the Automount Facility to Manage User Home Directories" on page 133
- "Performance Considerations for Shared HFS" on page 133

Highlights of Our Strategy

Our strategy has the following highlights:

- We use the shared HFS function to share the serviceable portions of the HFS (the version HFS) across systems in the sysplex. The customizable information is kept in system specific HFSs. The sysplex root HFS directs requests to either the version or system specific HFS, where the actual data is. See "The Shared HFS in Our Environment" on page 110.
- Suggested naming conventions for:
  - The build and production version HFSs (associating the version HFS with a specific SYSRES volume).
  - The mount point of the build version HFS.
  - The system specific HFSs for customizable information.

  See "HFS Naming Conventions" on page 130.
- A BPXPRMxx parmlib member shared across systems in the sysplex. We'll call this the shared BPXPRMxx member in the rest of this chapter. In shared HFS mode, the shared BPXPRMxx member contains definitions for the file systems that are shared across the participating group. This is facilitated by the above naming conventions. See "Customizing our Shared BPXPRMxx Member" on page 121.

  Some of our systems also have their own individual BPXPRMxx member for system specific file system definitions. See "Customizing BPXPRMxx Members for Individual Systems" on page 121.
- A post-install customization EXEC that we use after we apply service to the HFS. See "Our Post-Install Customization EXEC" on page 131.
- Use of the automount facility for managing users' home directories. See "Using the Automount Facility to Manage User Home Directories" on page 133.

Migrating to a Shared Hierarchical File System (HFS)

To migrate to the OS/390 R9 shared HFS, we used the *OS/390 UNIX System Services Planning* book. See also the following informational APARs before implementing shared HFS:

- APAR II12212
- APAR II12249

In the topics below, we'll discuss our shared HFS environment and the migration to shared HFS.

The Shared HFS in Our Environment

In our environment, we've shared our serviceable root HFS across the sysplex mounted read-only since OS/390 V1R1. However, users logged on to systems at the R8 level and earlier could only access directories that were mounted on the systems they were logged onto. Now, with OS/390 R9, OMVS users can access read-write directories and files from HFSs mounted on other systems in the sysplex, as long as the following conditions are met:

- The systems are at the R9 level.
- Each system is a member of the SYSBPX XCF group participating in the shared HFS environment. Specifying SYSPLEX(YES) in the BPXPRMxx member for a system makes it a member of the SYSBPX XCF group. (Systems specifying SYSPLEX(NO) cannot use the shared HFS function.) We'll call the SYSBPX XCF group the *participating group* in this chapter.
- There is an OMVS couple data set defined. (See "Creating an OMVS Couple Data Set (CDS)" on page 122.)
- The user's OS/390 UNIX userid has the correct permission bits set to have access to the directory or file.

To create the shared HFS environment, UNIX System Services changed and added additional HFSs. The following table shows a comparison between R8 and R9 HFSs:

OS/390 R8 HFSs	OS/390 R9 HFSs
N/A	**Sysplex Root HFS** - mounted read-write.
Root HFS - mounted read-only.	**Version HFS** - mounted read-only.
N/A	**System Specific HFSs** - mounted read-write.

Note that in shared HFS mode, the root HFS from before R9 is now called the version HFS. The R9 HFSs are described in the topics below:

- "The Sysplex Root HFS" on page 111.
- "The Version HFS" on page 113.
- "The System Specific HFSs" on page 114.

We provide security for our HFS files and directories the same way we did for our R8 HFS environment, using permission bits.

One of the most important points about the R9 shared HFS environment is that though there are multiple HFS file systems which are accessible to all users, the physical location of files and directories are transparent to the user.

The following picture shows a partial picture of our HFS environment, both the logical appearance and physical structure:

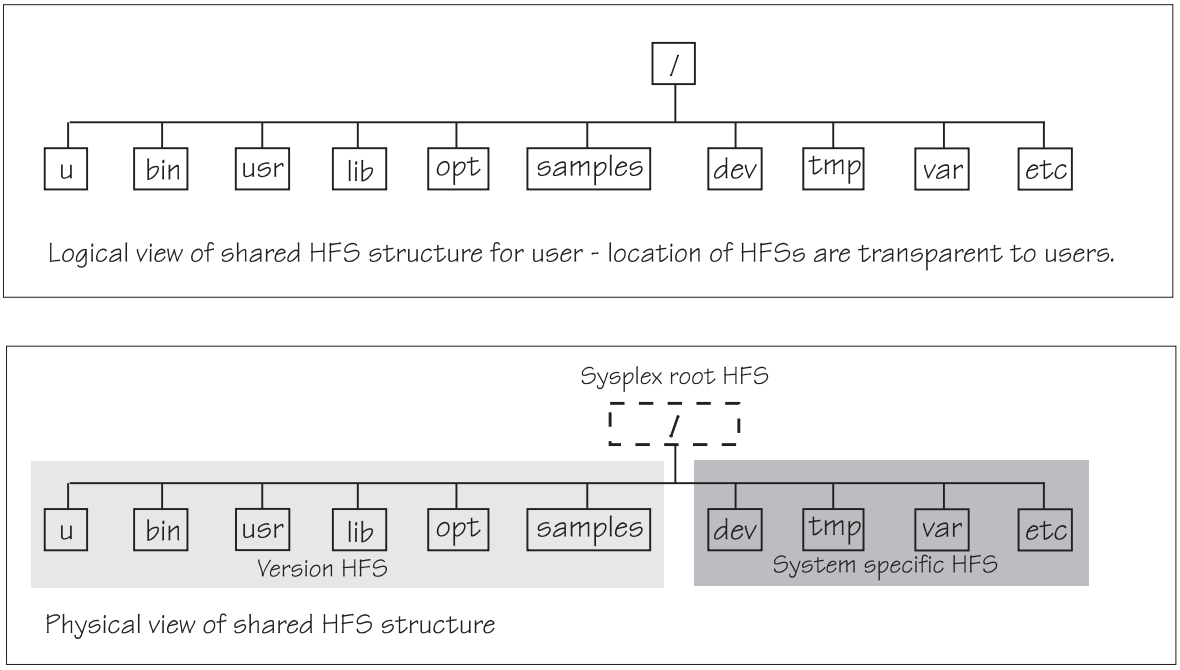


Figure 24. Logical and Physical View of Shared HFS

The Sysplex Root HFS

The sysplex root HFS is used as the sysplex-wide root. It contains symbolic links that redirect requests to the version and system specific HFSs. It also contains directories for HFSs shared by all systems, such as Java.

The sysplex root HFS is very small in size and does not have files containing system or user data. The recommended size is one cylinder, but because of the large number of entries we have in our sysplex root, we've allocated two cylinders for our sysplex root.

For systems in the participating group using shared HFS, the sysplex root is the new root HFS. For R9 systems that are not in the participating group, the sysplex root is not used.

The following figure shows the sysplex root before we customized it for our environment:

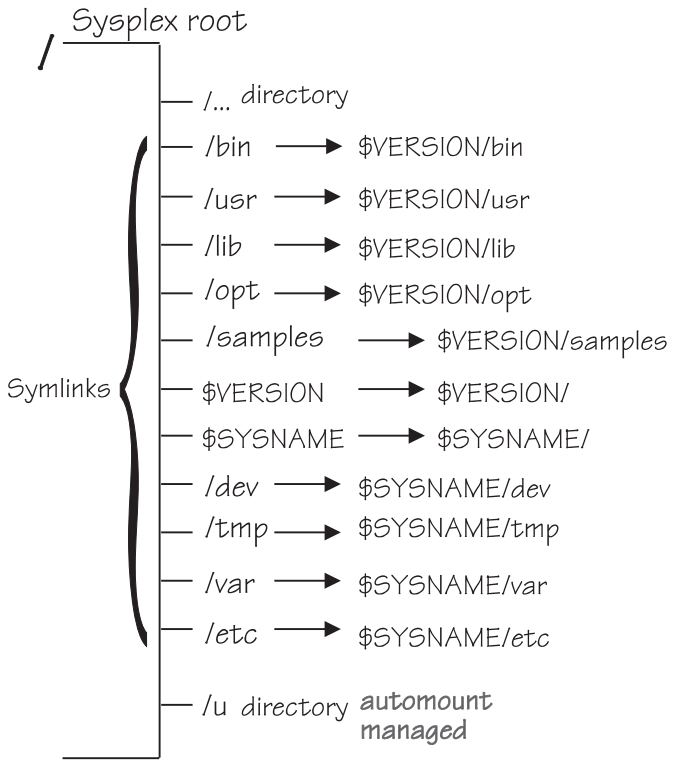


Figure 25. Our Sysplex Root

There are two symbols in our sysplex root, $VERSION and $SYSNAME. $VERSION will resolve to the version HFS in use for a system. The $VERSION symbol is defined in the BPXPRMxx SYS1.PARMLIB member for each system. $SYSNAME will resolve to the system name (J80, for example) for the system making a request. $SYSNAME is defined in the symbols table in the IEASYMxx parmlib member.

Note that in our sysplex root, the $VERSION and $SYSNAME are not preceded by slashes like /bin or /etc. This is because $VERSION and $SYSNAME are symbols.

The sysplex root HFS should be mounted with the AUTOMOVE value specified in the sysplex root file system definition in the shared BPXPRMxx member. Using AUTOMOVE means that the sysplex root HFS will be mounted and available even if the owning system leaves the sysplex.

We created the sysplex root HFS using the BPXISYSR sample job in SYS1.SAMPLIB for OS/390 R9. We did the following before running the BPXISYSR job:

- Named the HFS data set name for our sysplex root OMVSSPN.SYSPLEX.ROOT.FS.
- Customized the job as specified in the comments section in BPXISYSR.

This job created the sysplex root HFS and populated it with the required symbolic links and directories for R9 to give us the sysplex root shown in Figure 25. We mounted the newly created sysplex root HFS at a mount point on our current system so we could create the additional symbolic links and directories needed for our environment. For example, we created the following directories and symbolic links in our sysplex root for mounting HFSs shared by all systems:

- **Directories:**

  - /java
  - /netcommerce
  - /service
- **Symbolic links:**
  - /Notesdata -> $SYSNAME/Notesdata
  - /dgw_pet -> $SYSNAME/dgw_pet

The Version HFS

In shared HFS mode, the version HFS is the R9 name for the root HFS maintained by SMP/E. This HFS contains system code and binaries, including /bin, /usr, /lib, /opt, and /samples directories. One of the fundamental concepts behind the OS/390 R9 shared HFS is to completely separate those portions of the HFS that are subject to service updates from those portions of the HFS that might contain local customization. The version HFS contains only those files (generally executable code) that are SMP/E installable and serviceable, and should not contain any files that contain installation-specific or system-unique information. The object is to be able to apply service to the version HFS without having to save and restore customized files.

This HFS also contains symbolic links that redirect requests for the /etc, /tmp, /var and /dev directories to the $SYSNAME system specific HFSs.

Note that IBM delivers one version HFS; you can define more as you add new system or maintenance levels. We mount this HFS read-only and update it only at service time. See “Our HFS Service Environment” on page 126.

The following diagram shows our version HFS before we customized it:

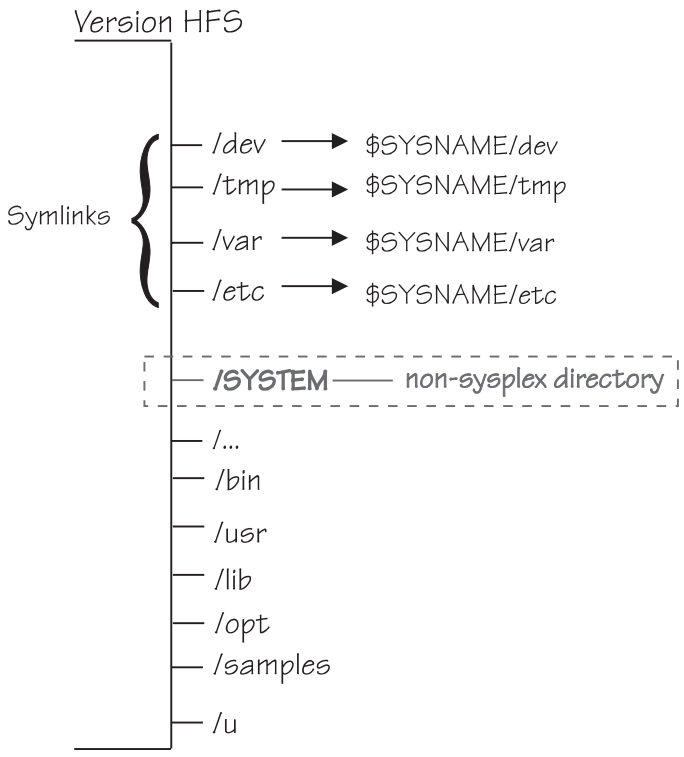


Figure 26. Our Version HFS

As with our pre-R9 root HFS, we had to customize the directories on the version HFS before we could IPL a system with it. We restored and customized the version HFS using a background job that does the following:

- Restores the HFS from the sequential dump copy on the system VOLSER.
- Mounts the restored HFS at a /service mount point on our existing system.
- Runs our post-install customization exec, CUSTHFS, to make the updates. (See "High-Level Steps for Servicing the HFS" on page 128 for details.)

  The CUSTHFS exec that we used to customize our file system for R9 shared HFS is included in the samples on our Web site..

A version HFS should be mounted with the default value AUTOMOVE specified in the shared BPXPRMxx member. Using AUTOMOVE means that ownership of the version HFS will be transferred to another system in the participating group if the owning system leaves the sysplex.

If you need to mount a product HFS file system at a /usr mountpoint, you will need additional symbolic links and directories. See "Mounting File Systems at /usr" on page 122 for details.

Note that in our environment, $VERSION resolves to a symbol, &SYSR1. Our &SYSR1 symbol then resolves to the name of the primary SYSRES volume that the specific system is IPLed from. We define the $VERSION symbol in the shared BPXPRMxx member. This means that the SYSRES VOLSER becomes the mountpoint for our version HFS. This gives us the flexibility to implement new service on a system in the sysplex without affecting the remaining systems — if we IPL on a new SYSRES VOLSER, that VOLSER name then becomes the mountpoint of the new version HFS for that particular system.

The System Specific HFSs

The system specific HFS is new for OS/390 R9 and is used only for systems IPLed in shared HFS mode. (See "How We Perpared Our Environment for R9 Shared HFS" on page 119 for additional details). This HFS contains files that are potentially customizable and are not subject to service updates. In this way, when we apply service, we do not have to be concerned about saving and restoring customized files. Each system has a set of unique HFSs that are mounted.

Each participating system in the sysplex has its own unique HFS which contains:

- Directories /etc, /dev, /tmp, and /var, which are used as mountpoints for customized file systems.
- Symbolic links for /bin, /usr, /opt, /lib, and /samples, which redirect requests for those directories back to the sysplex root, where they then resolve to directories in the $VERSION HFS.
- Additional directories where HFS file systems can be mounted if you need a specific maintenance level of a product for a system.

The following diagram shows how a system specific HFS looks:

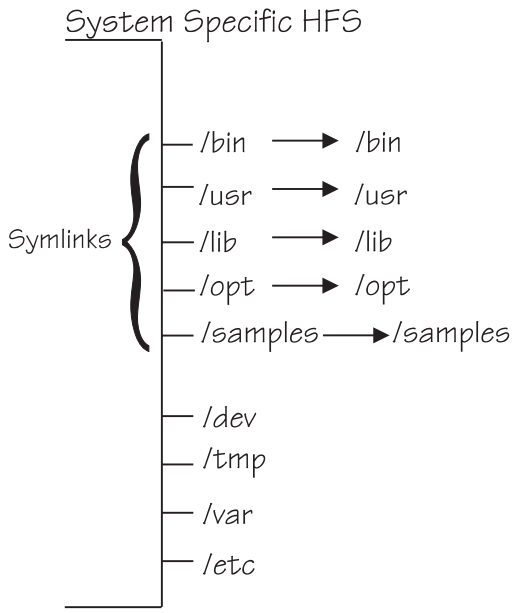


Figure 27. A System Specific HFS

A system specific HFS is mounted read-write and should be mounted NOAUTOMOVE. You can specify the NOAUTOMOVE value in the MOUNT statement for the system specific HFS in the shared BPXPRMxx member. Specifying NOAUTOMOVE means that if a system leaves the sysplex for any reason, ownership of the system unique HFS will not transfer to another system.

To create our initial system specific HFS, we edited and ran the sample BPXISYSS job in SYS1.SAMPLIB. The BPXISYSS job also populates the HFS with the required directories and symbolic links needed for operating in a shared HFS environment, as shown in Figure 27.

The naming convention for our system specific HFSs is as follows:

```
OMVSSPN.sysname.SYSTEM.FS
```

Once we had created the initial system specific HFS, we mounted the HFS at a mount point on our existing system so we could add the unique directories needed for our environment. For example, some of our systems use different product levels than the rest of our sysplex. Below we show a few of the unique directories we added:

- /dgw_pet
- /Notesdata
- /WebSphere

Once we had added the unique directories we needed, we used a DFSMSdss job to dump the HFS to a sequential data set. We then used DFSMSdss to restore and rename the dumped HFS to create a duplicate system unique HFS data set with a new name. We repeated this procedure to create a system specific HFS for all 16 systems in our sysplex.

Before using the system specific HFSs, you may need to initialize some of the other system specific file systems. We had already split out separate HFSs for /etc, /dev, /tmp, and /var, so we did not have to do this for R9. See “Initializing File Systems in the System Specific HFSs” on page 132 for how we did this.

Three Scenarios — How Requests Are Resolved In the Shared HFS Environment

The following topic contains a set of scenarios showing how all these HFSs fit together in the shared HFS configuration and how requests are routed through them. Note that in each of our scenarios, **all requests are routed and resolved through the sysplex root**.

Scenario One: Accessing /etc Directory

In our first scenario, a user on system J80 issues the ls -l /etc/ command to see the contents of the /etc directory on their system. The figure below shows how that request is routed through the sysplex root and resolved:
In the figure above, we show the sequence of how the request is resolved. The

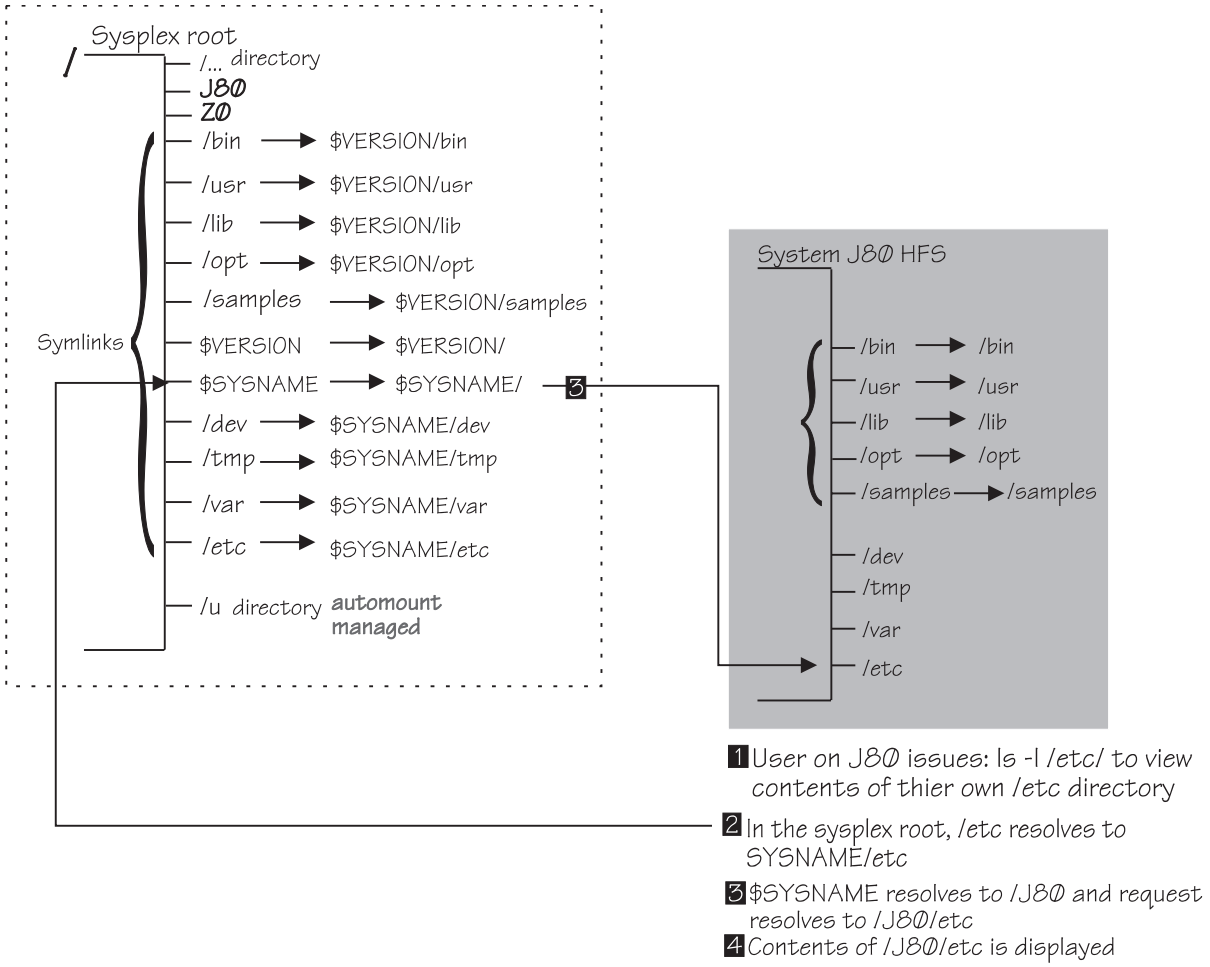


Figure 28. Scenario One: How a Request for /etc Resolves

numbers in the list correspond to steps in the diagram.

- 1 — The user, logged on to system J80 wants to see the contents of their own /etc directory, and so issues the ls -l /etc/ command.
- 2 — The request starts in the sysplex root where /etc resolves to $SYSNAME/etc.
- 3 — $SYSNAME resolves to /J80, because the user is logged onto J80. The request resolves to /J80/etc.
- 4 — The contents of /J80/etc are displayed for the user.

Scenario Two: Accessing Another System's /etc

In our second scenario, a user logged onto J80 issues ls -l /Z0/etc/ to see the contents of system Z0's /etc directory. The figure below shows how that request is routed through the sysplex root and resolved:
The numbers in the list below correspond to steps in the diagram.

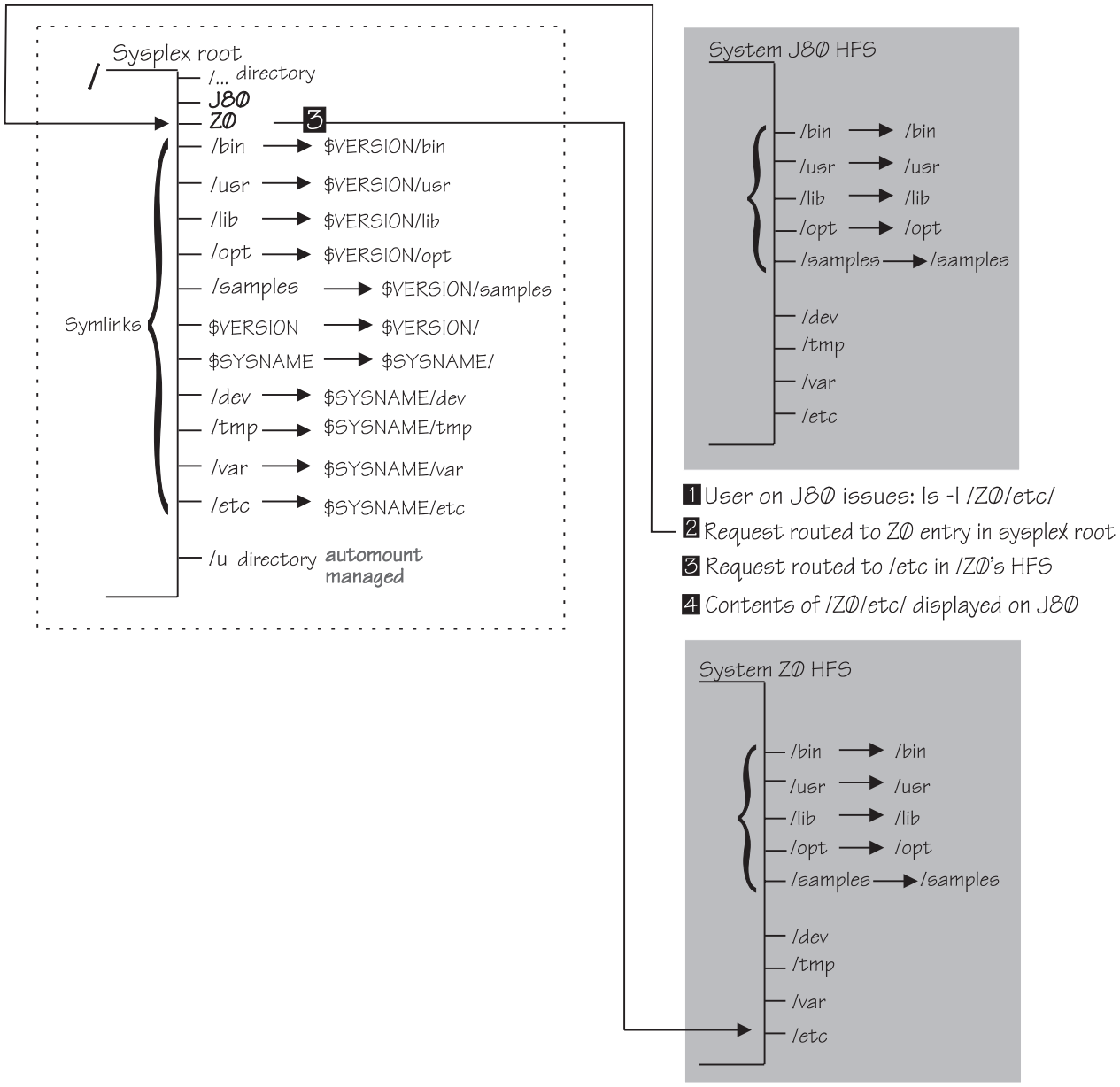


Figure 29. Scenario Two: Accessing Another System's /etc Directory

- **1** — The user, logged on to system J80, issues the ls -l /Z0/etc/ command to see the contents of system Z0's /etc directory.
- **2** — The request for /Z0/etc starts in the sysplex root, where there is an entry for Z0. UNIX System Services dynamically adds an entry for each system in the sysplex that is participating in the shared HFS.
- **3** — The sysplex root Z0 entry directs the request to system Z0, where it further resolves to /etc.
- **4** — The contents of /Z0/etc are displayed for the user on J80.

Scenario Three: Accessing Shared Code in /bin

In our third scenario we show how a user can access shared code, in this case from the /bin directory. The figure below illustrates how a user logged onto system J80 can access the /bin directory:

Note that in this scenario, we've IPLed system J80 on VOLSER D83PB1, which

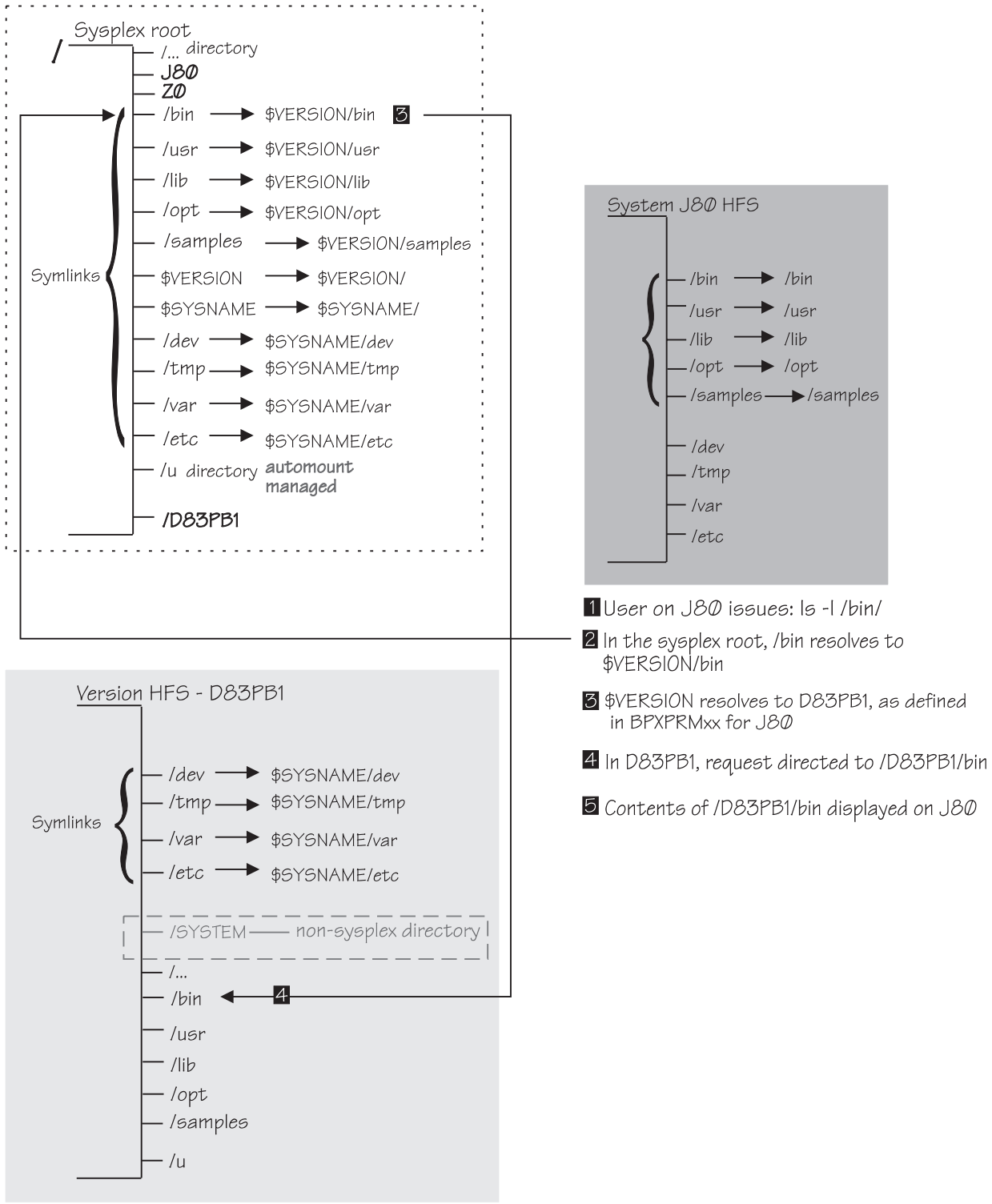


Figure 30. Scenario Three: Accessing the Shared /bin Directory

means that $VERSION will resolve to D83PB1. The numbers in the list below

correspond to steps in the diagram.

- 1 — The user, logged on to system J80, issues the ls -l /bin/ command to see the contents of the shared /bin directory.
- 2 — The request for /bin starts in the sysplex root, where it resolves to $VERSION/bin.
- 3 — $VERSION resolves to D83PB1, as defined in the BPXPRMxx parmlib member for system J80. Note that UNIX System Services dynamically adds an entry for version HFS D83PB1.
- 4 — The contents of /D83PB1/bin are displayed for the user on J80.

Note that UNIX System Services dynamically adds an entry for each version HFS in the sysplex root HFS.

Where Am I? Using the set -o logical Command

In Figure 24 on page 111, we showed that the actual HFS structure is transparent to the user. However, when a user is actually looking at the file systems from OMVS or ISHELL, the actual structure is not necessarily transparent. For example, from OMVS, a user may enter the following command:

```
cd /bin
```

The result of this command will be:

```
/D83PB1/bin $
```

The user has moved to the D83PB1 version HFS being used for this system. This result may come as a surprise to a user who may not be aware of where the /bin directory is or that they were connecting to it through a symbolic link. However, by using the set -o logical command, users can shield themselves from symbolic links and the details of where directories are physically kept. This gives the user a logical view of the HFS structure. For example, the following example shows what the user would see when they cd to /bin after issuing the set -o logical command:

```
:/u/user1 $ set -o logical
:/u/user1 $ cd /bin
:/bin $
```

Note that with set -o logical, all users see is /bin.

In order to shield our users from the confusion that can be created with symbolic links, we added the set -o logical command to /etc/profile.

How We Perpared Our Environment for R9 Shared HFS

On our R8 systems, we prepared our environment for R9 shared HFS using the steps listed below. Note that we were already sharing a read-only root and had separate HFSs for customizable file systems like /etc, /tmp, /var, and /dev when we made our migration.

1. Ran the BPXISCDS job to create and format the UNIX System Services OMVS couple data sets (OCDS). See “Creating an OMVS Couple Data Set (CDS)” on page 122.
2. Updated the COUPLExx member to add the primary and secondary OMVS couple data sets under the DATA TYPE(BPXMCDS) statement.
3. Ran the BPXISYSS job to create the system specific HFS. See “The System Specific HFSs” on page 114.
4. Temporarily mounted the system specific HFS to a directory on our existing system so that the required directories and symbolic links could be created.
5. Used DFDSS to dump/restore/rename the system specific HFS for the remaining systems.

6. Ran the BPXISYSR job to create our sysplex root HFS (you can only have one sysplex root per participating group). See “The Sysplex Root HFS” on page 111.
7. Temporarily mounted the sysplex root HFS to a directory on our existing system so that we could create the directories and symbolic links unique to our environment. For example, the following are a few of the directories and symbolic links we created in our sysplex root:
   - **Directories:**
     - /java
     - /netcommerce
     - /service
   - **Symbolic links:**
     - /Notesdata -> $SYSNAME/Notesdata
     - /dgw_pet -> $SYSNAME/dgw_pet
8. Customized the CUSTHFS exec as needed for R9 changes. The CUSTHFS exec that we used to customize our file system for R9 shared HFS is included in the samples on our Web site.
9. Created a new shared BPXPRMxx member for all our R9 systems in the participating group. See “Customizing our Shared BPXPRMxx Member” on page 121.
10. Updated the individual BPXPRMxx members for each system in the participating group. See “Customizing BPXPRMxx Members for Individual Systems” on page 121.
11. We allocated one HFS per user to be shared throughout the sysplex. Prior to R9, we had one HFS per user **per system**, because before shared HFS, we couldn’t share them between systems. In order to migrate to shared user HFSs, we had to combine each user’s HFSs from all the systems in the sysplex into one that can be shared. We did this on a system-by-system basis, as we migrated each system to R9. We performed the following steps for each system, as follows:
    - First for each user, we dumped all the user’s directories and files from the R8 system into a partitioned data set (PDS). We use a separate PDS for each user. Our naming convention for the PDSs is *userid*.pds(*system_name*). To dump the contents of a user’s HFS (/u/userid) from a specific system into a PDS, we issued the following command:

      ```
      /u/userid $ ==> pax -wf "//'userid.pds(system_name)'" *
      ```

      We use *system_name* as a variable for each member, so that we store data from each system separately.
    - We updated our automount policy for our new naming convention for our new shared user HFS data sets. See “Using the Automount Facility to Manage User Home Directories” on page 133 for how automount automatically mounts these for R9 systems.
    - We then IPLed the system on R9.
    - Then, from our newly migrated R9 system, we restored the files from each user’s PDS into their shared user HFS. To do this, we issued the following command for each user, with the user’s shared HFS as the home directory:

      ```
      /u/userid $ ==> pax -rf "//'userid.pds(system_name)'" *
      ```

    For example, for user A on R8 system SYS1, we dumped all the directories and files from user A’s HFS into PDS USERA.PDS(SYS1). Then, after IPLing SYS1 on R9, we restored them into user A’s one shared user HFS.

    When you restore the files and directories to a shared user HFS, note that the pax command automatically overwrites any files or directories in the shared

user HFS that have the same name as those being restored. If you want to restore identically named files or directories but do not want to overwrite them in a shared user HFS, restore them into a different directory. For example, use the following command to restore files and directories into /u/userid/ **NEWDIR** HFS instead of /u/userid:

```
/u/userid/ NEWDIR  $ ==> pax -rf"//'userid.pds(system_name)'" *
```

After we completed these steps to prepare our environment for shared HFS, IPLing our systems at the R9 level to brought us right into shared HFS mode.

Customizing our Shared BPXPRMxx Member

Because we were already sharing our HFS prior to R9, we had already defined a shared BPXPRMxx member for our file systems shared across the sysplex. For R9 shared HFS function, we customized our shared BPXPRMxx member for file systems shared across the participating group. We're sharing more files now accross the sysplexThe naming convention we use for our version and system specific HFSs makes this easier — we use HFS data set names that contain qualifiers that can be represented with system symbols (see "HFS Naming Conventions" on page 130). For example, we can use the system symbol &SYSR1 in the version FILESYSTEM statement because the version HFS name contains the VOLSER of the associated SYSRES. For example:

```
FILESYSTEM('hlq.&SYSR1..ROOT.llq') TYPE(HFS) MODE(READ)
```

Note that this mounts the build version HFS **read-only**.

We can use the system symbol &SYSNAME in the MOUNT statements because the system specific HFS names contain the name of the associated system. For example:

```
MOUNT FILESYSTEM('hlq.&SYSNAME..TMP.llq') TYPE(HFS)
  MODE(RDWR) MOUNTPOINT('/tmp') NOAUTOMOVE
```

We kept the settings we had in our R8 shared BPXPRMxx member, and added the following statements:

- VERSION('&SYSR1')
- SYSPLEX(YES)
- MOUNT statements for HFS file systems that will be mounted at the sysplex root
- AUTOMOVE for the sysplex root and version HFSs, so that if the owning system leaves the sysplex, ownership will transfer to another system and they'll remain mounted.

A portion of our common BPXPRMxx member is included in the samples on our Web site.

Customizing BPXPRMxx Members for Individual Systems

We have defined nearly all of our HFSs in our shared BPXPRMxx member, as reccommended in the *OS/390 UNIX System Services Planning*. We do, however, define a few HFSs in BPXPRMxx members for individual systems. We do this to define those HFSs that we want mounted **only** on one particular system. For example, lets say we're using two levels of a product, such as the IBM HTTP Server. On most of our systems, we want to use the latest level. However, for one system, system A, we are running an earlier level of the IBM HTTP Server. In the BPXPRMxx member for system A, we'll define the HFS containing that specific level of the IBM HTTP Server. We define these HFSs with the NOAUTOMOVE parameter, so that if the owning system leaves the sysplex, these HFSs will not be

mounted on any other system. We do not want ownership of customized system specific information transferred to another system.

We do not define the HFSs that we want mounted on one particular system in the shared BPXPRMxx member, because when each system IPLs, it will try to mount every HFS defined in the shared BPXPRMxx member.

There is a sample of an indiviual system BPXPRMxx member among the samples on our Web site.

Creating an OMVS Couple Data Set (CDS)

Using the R9 shared HFS requires an OMVS couple data set which contains:

- The sysplex-wide mount table
- Information about all participating systems in the sysplex
- Information about all mounted file systems in the sysplex.
- The current automount policy.

We created our OMVS CDS using the sample BPXISCDS SYS1.SAMPLIB member. We updated the sample with the CDS names and VOLSERs for our environments, and then ran the BPXISCDS to create and format primary and secondary OMVS couple data sets. Next, we updated the COUPLExx parmlib member to add the primary and secondary couple data sets under DATA TYPE(BPXMCDS).

Note that when you IPL an R9 system in a sysplex in shared HFS mode, the cross-system coupling facility (XCF) issues the following message on any pre-R9 level systems:

```
IXC284I UNABLE TO LOAD BPXMCDSF FOR BPXMCDS
```

This message did not impact our R8 systems.

Mounting File Systems at /usr

In our environment we have several special HFS file systems that are mounted at a /usr mount point. These special file systems are either separately serviceable product files **or** spool files. Prior to R9, when the /usr directory was in the pre-R9 root HFS, we just mounted these file systems at a specific directory in the root (such as /usr/lpp/java or /usr/spool). Now however, in the R9 shared HFS environment, all file systems must be mounted at a directory in either the sysplex root or system specific HFS. You **should not** create directories in the version HFS, or mount file systems at these directories because requests for these directories are actually resolved with the $VERSION symbol. Because our $VERSION symbol is based on the SYSRES volume where we IPLed the system, it can change whenever we IPL (to pick up new OS/390 service, for example) to a new SYSRES volume. If you do mount file systems at directories in the version HFS, users may not be able to access these directories after an IPL. The sections below show how we replaced these /usr file systems with symbolic links to the sysplex root or system specific HFS.

Mounting Product File Systems at /usr

We have several product HFS file systems, such as Java, that are serviced separately from our R9 base SMP/E environment. Many of these products require the file system to be mounted at a /usr mount point. As we said above, before R9 we could just mount these HFS file systems at a directory in the root. Now however, in the R9 shared HFS environment, all file systems must be mounted at a directory in either the sysplex root or system specific HFS. In the sections below, we'll show

examples of why creating a directory for a separately serviceable file system in the version HFS will not work, and how putting them into the sysplex root instead **will** work.

Creating a Directory in the Version HFS — At First, This Will Work: Lets say we create a directory and mount a file system at a /usr/lpp/java directory in the current version HFS, VER1. A request for /usr/lpp/java would resolve as /VER1/usr/lpp/java, where the file system is actually mounted. This will work initially, because version VER1 is the version HFS in use, as shown in the figure below: In the figure above, the numbers (like 1) correspond to steps in the diagram:

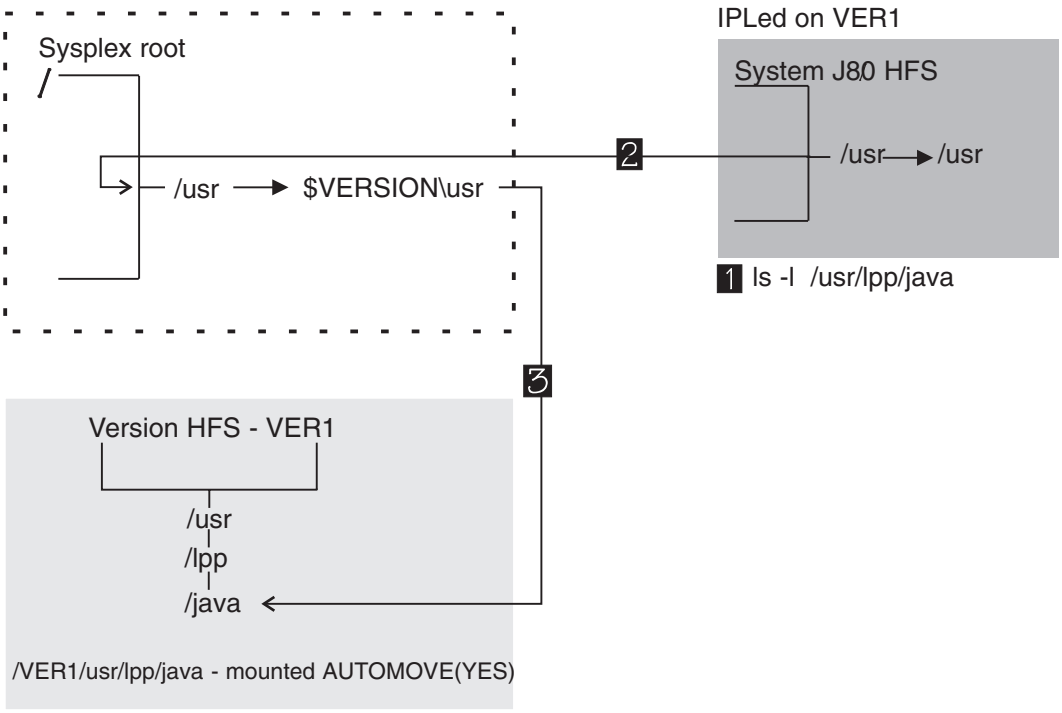


Figure 31. Creating a Directory in the Version HFS: This Will Work Until You IPL the System With a New Version HFS

- 1 — The user, logged onto system J80, issues the ls -l /usr/lpp/java command to see the contents of the /java directory.
- 2 — The request starts at the sysplex root, where /usr resolves to $VERSION/usr.
- 3 — $VERSION resolves to VER1 and the request resolves to /usr/lpp/java in VER1.

Creating a Directory in the Version HFS — This Stops Working When You IPL With A Different Version HFS: Now that we've created a directory and mounted a file system at /usr/lpp/java, let's imagine that we IPL system J80 from a different SYSRES volume and version HFS, VER2. Now a request for /usr/lpp/java will resolve to VER2/usr/lpp/java. But let's also assume that we specified AUTOMOVE(YES) for the file system at /usr/lpp/java, which is recommended for file systems in the version HFS. AUTOMOVE(YES) means that if the owning system leaves the sysplex, ownership transfers to another active system in the participating group. That means that the file system mounted HFS at /VER1/usr/lpp/java will **remain** mounted there. When J80 is IPLed with the new SYSRES volume and version HFS, VER2, the system will try to mount all the mounts defined in the common BPXPRMxx member, resolving them to the new version, including VER2/usr/lpp/java. However, the /java mount will fail because the HFS for /java is still mounted at VER1/usr/lpp/java. It will fail because we specified

AUTOMOVE(YES) for the file at /usr/lpp/java, which means that when system J80 reIPLed ownership for that file transferred to another system where it is still mounted and available. When we tested this, we got the following error message:

```
BPXF002I FILE SYSTEM OMVSSPN.JAVA.FS WAS
NOT MOUNTED.  RETURN CODE = 00000079, REASON CODE = 055B005B
```

The following figure illustrates how this fails:
In the figure above, the numbers (like 1) correspond to steps in the diagram:

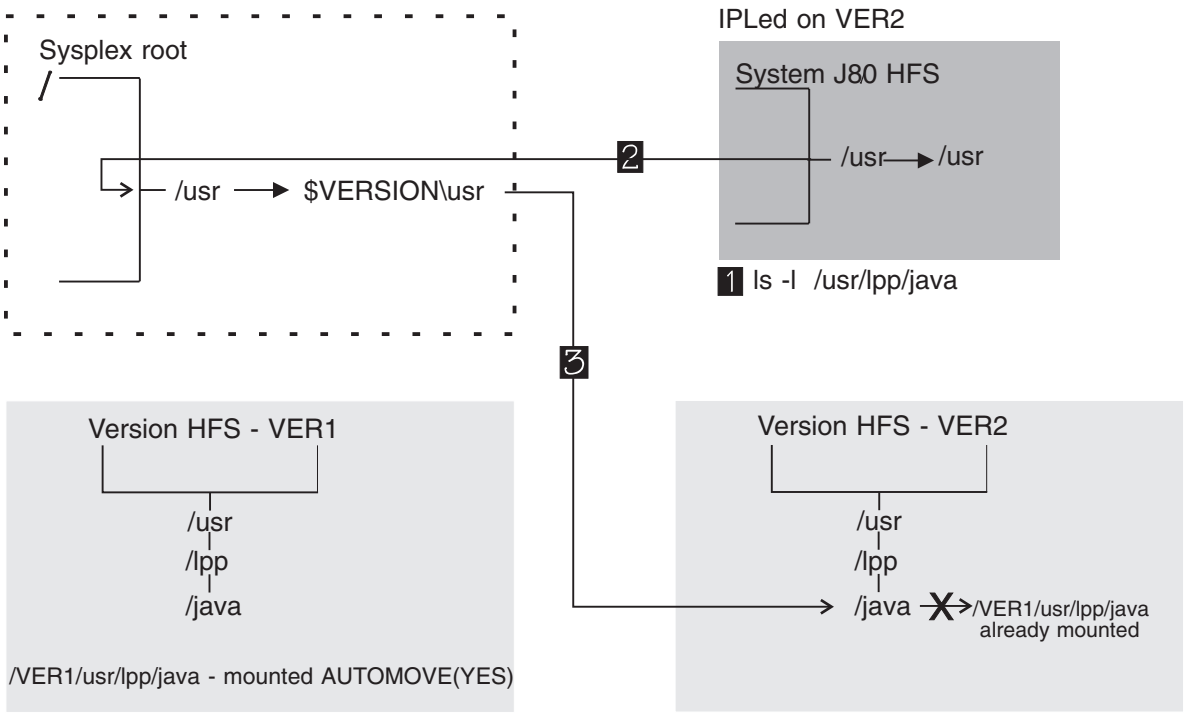


Figure 32. Creating a Directory in the Version HFS: This Stops Working When You IPL the System With a Different Version HFS

- 1 — The user, logged onto system J80 (which has now been IPLed on VER2), issues the ls -l /usr/lpp/java command to see the contents of the /java directory.
- 2 — The request is routed to the sysplex root, where /usr resolves to $VERSION/usr.
- 3 — $VERSION resolves to VER2. The request is routed to /usr/lpp/java in VER2. However, nothing is displayed to user on J80 — The mount for /VER2/usr/lpp/java failed at IPL, because /usr/lpp/java is still mounted at the /VER1/usr/lpp/java mount point.

Creating a Symbolic Link in The Sysplex Root — This Will Work Even When You Switch To a Different Version HFS: To avoid the problems you run into if you create directories and mount file systems in the version HFS, you can create directories and mount file systems in the sysplex root or system specific HFSs. This way, the mount point will not change when you IPL a system on a different SYSRES volume. Because we are creating directories for separately serviceable products that require the file system to be mounted at a /usr mount point HFS, we made /java in the version HFS a symbolic link at customization time (CUSTHFS exec), as follows:

```
java -> /java
```

This symbolic link redirects the request to the sysplex root HFS, where we have created a /java directory. We then mount the file system containing our java code at

/java. Requests for the /usr/lpp/java directory will resolve to the /java directory in the sysplex root and users will get the correct file system, no matter what version HFS is in use. The following figure shows how a request for /usr/lpp/java will be routed to /java in the sysplex root:
In the figure above, the numbers (like 1) correspond to steps in the diagram:

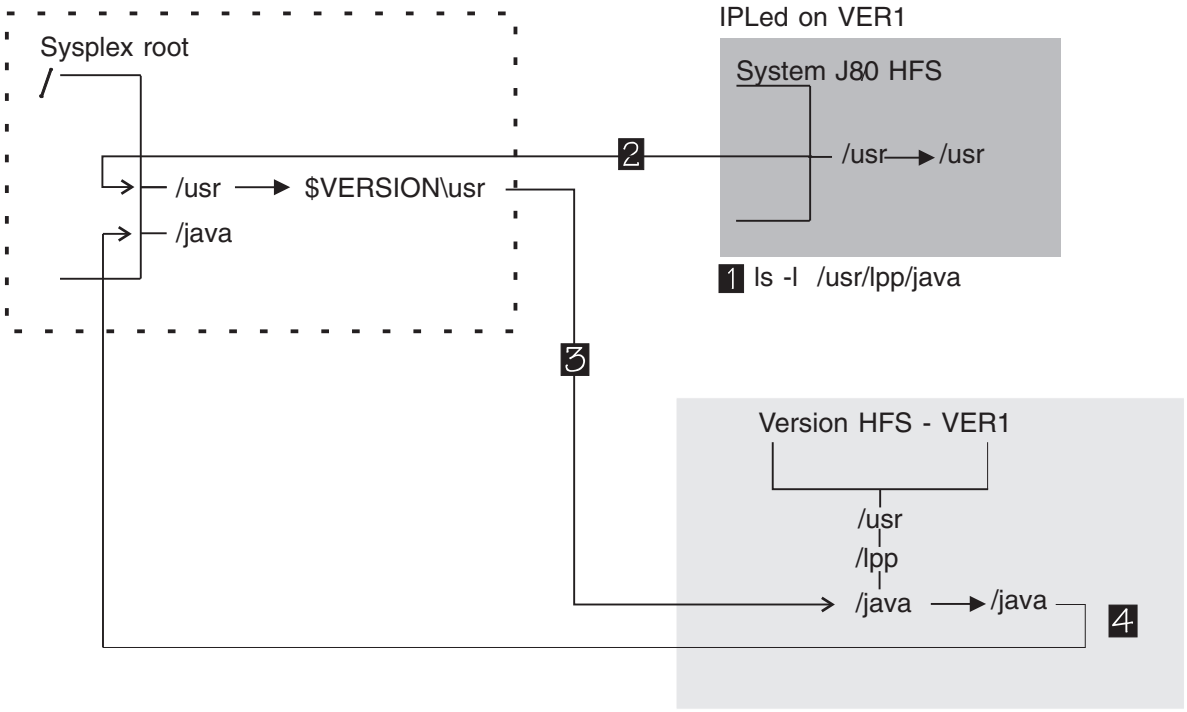


Figure 33. How To Mount Serviceable File Systems at /usr

- 1 — The user, logged onto system J80, issues the ls -l /usr/lpp/java command to see the contents of the /java directory.
- 2 — The request starts at the sysplex root, where it resolves to $VERSION/usr. The $VERSION symbolic link resolves to VER1.
- 3 — In the VER1 version, the request resolves to /usr/lpp/java.
- 4 — /java is a symbolic link that sends the request from VER1 back to /java in sysplex root, where the Java file system is mounted. The contents of /java is displayed for the user on J80.

Because we've coded AUTOMOVE(YES) in the mount statement of the BPXPRMxx parmlib member for the file system mounted at /java, ownership of this file system will transfer to another active system in the participating group if the owning system leaves the sysplex. That means that the file system mounted at /java will remain mounted somewhere in the sysplex, always accessible to users requesting /usr/lpp/java.

Mounting File Systems at /usr/spool

You must be able to mount a unique HFS for each system at /usr/spool for queueing UNIX-to-UNIX Copy Program (UUCP) file requests. The version HFS is delivered with a /usr/spool directory and in a non-shared HFS mode, you can leave this path in the version HFS and mount your unique spool HFS there. This works because in non-shared HFS mode, the version HFS is the root. However, if you leave the /usr/spool path in the version HFS in shared HFS mode, you'll lose access to that HFS every time you IPL with a different version HFS, as we showed above in "Mounting Product File Systems at /usr" on page 122. When we migrated

to shared HFS mode, we had to delete the /usr/spool directory and replace it with a symbolic link redirecting the request to a /spool directory in the system specific HFSs for each system, as shown below:

```
/spool-> $SYSNAME/spool
```

In our environments, these changes are made by the post-install customization EXEC, CUSTHFS, which we run every time we perform our service procedure.

After we create the symbolic link for /spool in the version HFS, we then add a /spool directory in each system specific HFS. Then a request for the spool HFS is resolved and mounted at the /spool directory in the system specific HFS. The example below shows how this looks:
In the figure above, the numbers (like 1) correspond to steps in the diagram:

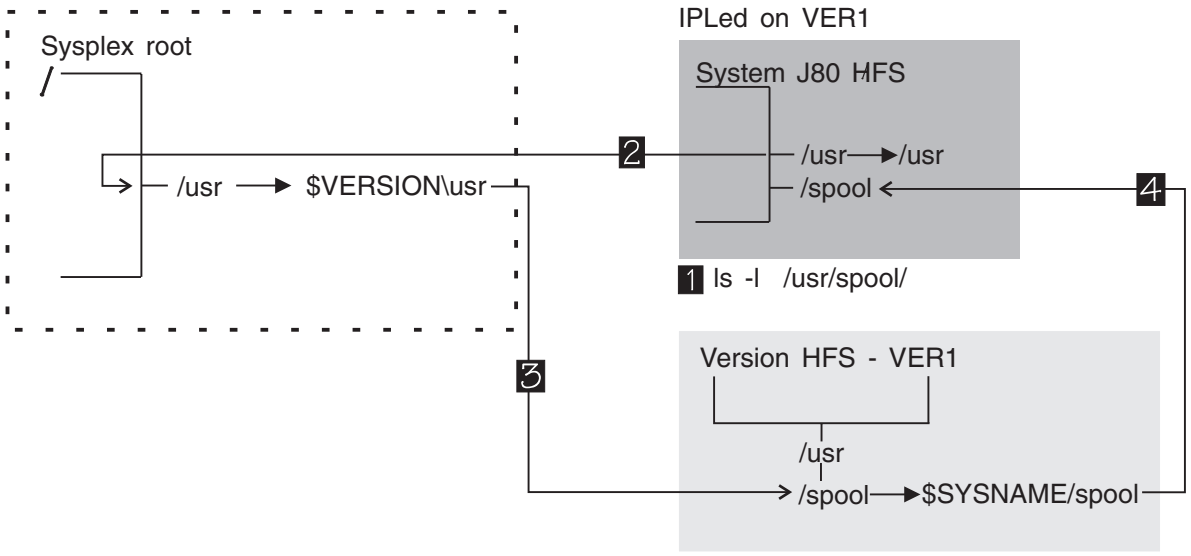


Figure 34. Mounting a File System at /usr/spool

- 1 — The user, logged onto system J80, issues the ls -l /usr/spool/ command to see the contents of the /spool directory.
- 2 — The request starts at the sysplex root, where it resolves to $VERSION/usr. The $VERSION symbolic link resolves to VER1.
- 3 — In the VER1 version, the request resolves to /usr/spool.
- 4 — /spool is a symbolic link that sends the request from VER1 to /spool in the system specific HFS, where the spool file system is mounted. The contents of /J80/spool is displayed for the user on J80.

Our HFS Service Environment

We maintain one build version HFS on a separate system and then copy the version HFS to our production system. We do this because several different test groups are using this same serviced build root HFS. After we've copied the SMP/E serviced root to our production version HFS, we do the customization we need for our environment.

Figure 35 on page 127 shows our current environment — note that the numbers in the drawing (3 for example) refer to steps in our "High-Level Steps for Servicing the HFS" on page 128:

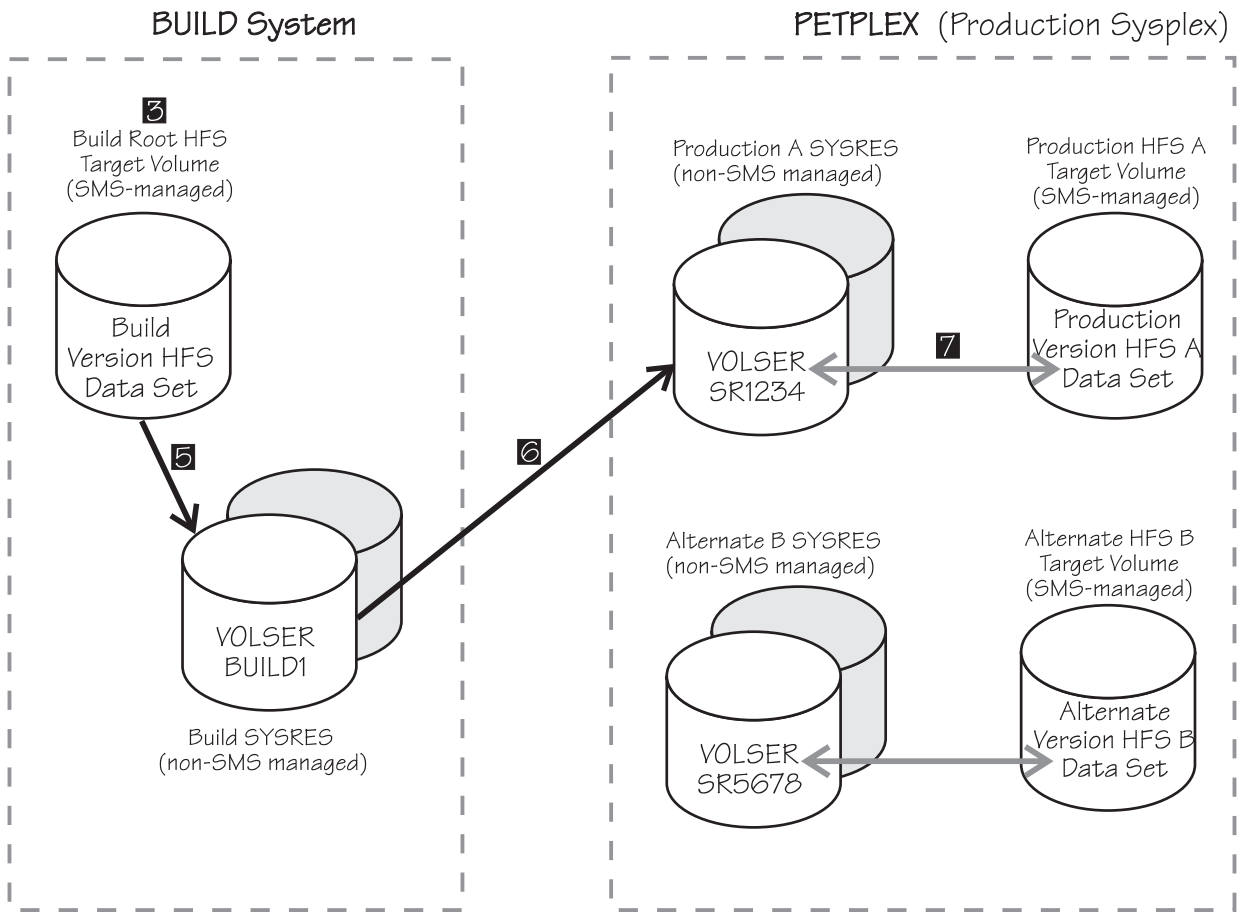


Figure 35. Applying Service in Our HFS Environment:

Our environment has the following features:

- On the build system, we have one build root HFS residing on an SMS-managed volume, referred to as the ***HFS target volume*** in the section on recommended data set placement in *OS/390 Planning for Installation*. We keep our build HFS on a system in a sysplex other than our production sysplex.
- On the PETPLEX production sysplex, we have two root HFSs residing on SMS-managed HFS target volumes. We switch these HFSs between production and alternate — at any given time, one will be in production and the alternate waits for the next service update. When we have a service update, we copy service to the alternate HFS and then IPL it into production.
- Each of our version HFSs is ***associated with*** a SYSRES volume. The HFS does not reside on the SYSRES, but is associated with the SYSRES through a naming convention, which we explain in "HFS Naming Conventions" on page 130. This association allows us to use the correct level of the HFS by IPLing systems with the correct level of the SYSRES. The version HFS resides on a separate SMS-managed volume, referred to as the HFS target volume in the section on recommended data set placement in *OS/390 Planning for Installation*.

  When we speak of a SYSRES volume, it is a logical volume comprised of two physical volumes from which we IPL our systems. At any given time in our environment, we have at least three of these logical SYSRES volumes:
  - SYSRES volume BUILD1 is the build SYSRES
  - SYSRES volume SR1234 is the production SYSRES
  - SYSRES volume SR5678 is the alternate SYSRES

Figure 36 shows how a root HFS is associated with a SYSRES volume.

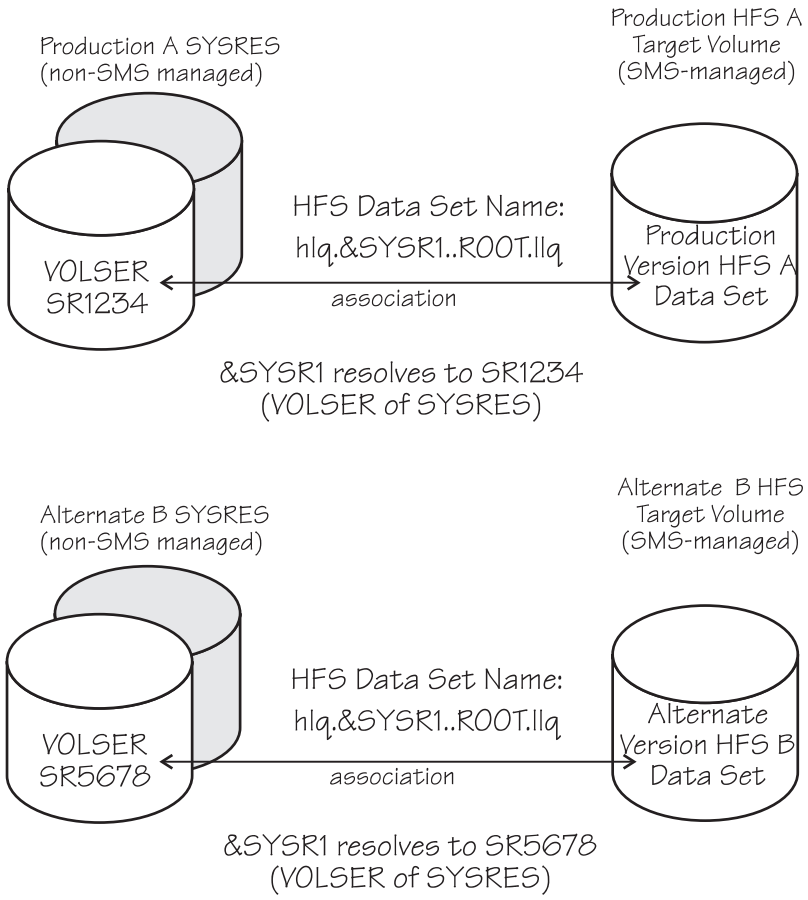


Figure 36. Association of a Root HFS Data Set with a SYSRES Volume

If we had different systems in our sysplex at different service or product levels during a migration window, for example, we might have additional SYSRESs for IPLing systems at different levels.

High-Level Steps for Servicing the HFS

Using our strategy, the steps below show our high-level steps for performing service on the build version HFS and copying it to the alternate HFS. The main change to our service procedure for the OS/390 R9 shared HFS environment is that we service a build version HFS rather than the build root HFS.

We switch the two version HFSs on the PETPLEX production sysplex between production and alternate — at any given time, one will be in production and the alternate waits for the next service update. Refer to Figure 35 on page 127 for an illustration of this process. For this procedure, we're assuming that root HFS A is the current production HFS.

In the steps below, we'll copy the service update to the current alternate HFS, HFS B.

1. Create a directory to be used to contain mount points for the build version HFS (we use /service). Then allocate a small HFS data set and mount it at this directory on your build system, read-write. You need only create the /service

directory and mount this HFS once manually. (For future mounts, the CUSTHFS exec will create the /service directory and you can place the TSO MOUNT command in the BPXPRMxx parmlib member.) Note that we use a system outside our production sysplex for our build system.

Here is an example of the TSO command to create the /service directory:

```
mkdir /service
```

Here is an example of the TSO command to mount the HFS for the service directory on the build system (entered on 1 line):

```
mount filesystem('hlq.BUILD.SERVICE.llq') mountpoint('/service')
   type(hfs) mode(rdwr)
```

When you put this mount command in your BPXPRMxx parmlib member, you must replace BUILD (the name of our build system) with &SYSNAME., the system symbol for the system name. (The &SYSNAME. system symbol is defined in the IEASYMxx member of SYS1.PARMLIB.)

2. Create a directory under /service to be used as the mount point. Mount the build version HFS to be serviced at that mount point, read-write. We created this additional directory under /service because as a test shop, we need to be able to service or customize more than one version HFS at a time. We could also simply mount the build version HFS at the /service mount point.

   Here is an example of the TSO command to create the /service/BUILD1 directory, the mountpoint for the build version HFS:

   ```
   mkdir /service/BUILD1
   ```

   Here is an example of the TSO command to mount the HFS read-write (entered on 1 line):

   ```
   mount filesystem('hlq.BUILD1.ROOT.llq') mountpoint('/service/BUILD1')
      type(hfs) mode(rdwr)
   ```

3. Perform the SMP/E service procedure on the build version HFS mounted at mountpoint /service/BUILD1. This step corresponds to **3** in Figure 35 on page 127.

   A description of the SMP/E service procedure is beyond the scope of our test report. However, you should make sure that you reference the mount point for the build version HFS in your DDDEF path. You can use the SMP/E ZONEEDIT command to change your DDDEF paths to the mount point for the build version HFS as follows:

   ```
   SET BDY(#tzone) .   ZONEEDIT DDDEF.
     CHANGE PATH('/service/xxxxxx/'*,
                 '/service/BUILD1/'*).
     ENDZONEEDIT.
   ```

4. Dump the build version HFS to a sequential data set on the build SYSRES volume (BUILD1). We use DFSMSdss to dump the build version HFS. This step corresponds to **5** in Figure 35 on page 127.
5. Copy the build SYSRES (BUILD1) to the alternate SYSRES volume (SR5678). This step corresponds to **6** in Figure 35 on page 127.
6. Now that the build SYSRES volume has been copied to the alternate SYSRES (SR5678), you must use DFSMSdss to restore the alternate version HFS from the sequential data set on the alternate SYSRES (SR5678). (Note that the HFS target volumes are SMS managed.) This step corresponds to **7** in Figure 35 on page 127.

   We use the RENAMEU parameter to name the version HFS using the convention we suggested. We also specify the SMS storage class with the STORCLAS parameter so that the root HFS will be allocated in the proper pool

of DASD. Our storage class for HFSs is SMSOE. Alternatively, you can set up SMS to automatically select the correct storage class based on the high-level qualifier of your HFS data sets. Sample dump and restore jobs are included in the samples on our Web site.

7. Run the post-install customization EXEC on the build version HFS. (See "Our Post-Install Customization EXEC" on page 131.)
8. IPL with the alternate SYSRES volume containing the new service update, SYSRES volume SR5678. Version HFS B, which is associated with SYSRES volume SR5678, is then mounted read-only on each system that IPLs with SYSRES SR5678.
9. Version HFS B is now the production HFS, and version HFS A is the alternate awaiting the next service update. Next time we apply service to our version HFS, we'll copy the service to version HFS A.

The sections following this one provide the detailed background and setup information you need to successfully perform these steps.

Samples Available on Our Web Site

As you read through this chapter, we'll refer to various samples that we've placed on the samples page of our Web site:

- CUSTHFS—this EXEC will customize the version HFS.
- DGWSS—IBM HTTP Server shell script to run setup.sh.
- BPXPRMxx—partial contents of the BPXPRMxx parmlib member.
- ETCAUTO—contents of /etc/auto.master.
- ETCUMAP—contents of /etc/u.map.
- HFSDUMP—version HFS dump job.
- HFSREST—version HFS restore job.

HFS Naming Conventions

We have suggested naming conventions for the following:

- ***Version HFS to be serviced.***

  We associate a version HFS with a SYSRES by using the VOLSER of the SYSRES as a qualifier in the name of the HFS data set. The VOLSER of the SYSRES can be represented with the system symbol &SYSR1. The name of our version HFS is as follows (where hlq is the high-level qualifier and llq is the low-level qualifier):

  ```
  hlq.&SYSR1..ROOT.llq.
  ```

  The hlq we use is defined as an alias in our master catalog, so any HFS data sets created with this hlq are actually cataloged in a user catalog. Note that we use the ***same*** hlq for all our HFS data sets, including the HFSs used for home directories. As a result, all our HFS data sets are cataloged in the ***same*** user catalog.

  This naming convention, illustrated in Figure 36 on page 128, works well for our enterprise and is intended to be a guide. You should choose a convention that works well in your S/390 enterprise.

  If your enterprise consists of multiple sysplexes that use the same SYSRES VOLSERs, then you could add the sysplex name as a qualifier in the version HFS data set name. The sysplex name can be represented by the system

symbol &SYSPLEX, which is defined in the LOADxx member of the SYS0.IPLPARM data set. The version HFS could be named:

```
hlq.&SYSR1..&SYSPLEX..ROOT.llq.
```

- ***Mount point of the build version HFS to be serviced.***

  The strategy requires a directory that will be used as a mount point for the build version HFS. We have created a directory named /service for this purpose, which is actually a small HFS mounted read-write. We then create a directory in /service that will be used as a mount point for the build version HFS. The name of the directory created in /service is the VOLSER of the SYSRES associated with the build version HFS. For example, the mount point for the build version HFS that is associated with build SYSRES BUILD1 will be named:

  ```
  /service/BUILD1
  ```

  See Figure 36 on page 128 for an illustration.

- ***System specific HFSs.***

  We associate our system specific HFSs with a system by using the system name as a qualifier in the name of the HFS data sets. The system name can be represented by the system symbol &SYSNAME., which is defined in the IEASYMxx member of SYS1.PARMLIB. For example, the HFSs mounted read-write at /etc on our systems are named:

  ```
  hlq.&SYSNAME..ETC.llq.
  ```

  The table below shows the system specific HFSs that are mounted read-write. Note we show the mount points as they are defined in our shared BPXPRMxx member:

Table 10. System Specific HFSs Mounted Read-Write

Mount Point	HFS Data Set Name
/$SYSNAME.	hlq.&SYSNAME..SYSTEM.llq
/$SYSNAME./etc	hlq.&SYSNAME..ETC.llq
/$SYSNAME./dev	hlq.&SYSNAME..DEV.llq
/$SYSNAME./tmp	hlq.&SYSNAME..TMP.llq
/$SYSNAME./var	hlq.&SYSNAME..VAR.llq
/$SYSNAME./spool	hlq.&SYSNAME..SPOOL.llq
/service	hlq.SERVICE.llq
/usr/lpp/java	hlq.JAVA.llq

These directories will ***not*** be updated during service (using SMP/E) or customization (using the CUSTHFS EXEC).

Our Post-Install Customization EXEC

Once the version HFS has been mounted read-write on your build system and the SMP/E service procedure has been performed, you need to do some customization. We do ours with a REXX EXEC named CUSTHFS that is included in the samples on our Web site. This EXEC must be executed from a superuser and be passed a parameter which is the mount point of the root HFS being customized. For example, to customize the build root HFS mounted at /service/BUILD1, enter the following command from TSO:

```
EXEC 'CUSTHFS' '/service/BUILD1'
```

The CUSTHFS EXEC performs various tasks outlined below. The sample /etc/rc that ships with OS/390 contains a default initialization shell script that performs many of these tasks. However, with a read-only root HFS, you cannot run these tasks out of /etc/rc, so we moved them to our CUSTHFS EXEC, which:

- Builds symbolic links to the various directories and symbolic links in the sysplex root HFS.
- Executes the IBM HTTP Server post-install shell script (setup.sh).
- Executes shell scripts that do the following:
  - A shell script which converts /etc to a symbolic link in the version HFS. (Note that our build group had already converted /etc to a directory — our shell script only had to convert /etc to a symbolic link.)
  - Deletes the /usr/spool directory from the version HFS.
  - Builds a symbolic link in the version HFS to the /usr/spool file system in the system specific HFS.
- Provides customization for the OS/390 Firewall Technologies Kit and OS/390 Print Server.

Prior to R9, we used CUSTHFS to create directories in the root for mounting the HFSs for other products. Now in R9, we had to move these directories to the sysplex root HFS or the system specific HFS.

Initializing File Systems in the System Specific HFSs

Before you can use some of the file systems in the system specific HFSs, they must be initialized with the appropriate information as follows:

Initializing the /etc HFS: The /etc HFS is used to hold parameters for individual systems, much like the SYS1.PARMLIB data set on OS/390. The way you get a customized /etc directory onto each system depends on whether you already have an existing HFS for your /etc directory. Note that we only describe how to initialize the /etc HFS **in a ServerPac environment** because we use ServerPac to install OS/390.

- **If you already have an existing HFS for your /etc directory**, see the program directories for your products to find the /etc directory updates needed.
- **If you are creating an HFS for the /etc directory for the first time**, you can do it any number of ways, but we did it as follows:
  1. Restored the ServerPac /etc HFS on our build system as part of ServerPac installation and customized it for our product environment. Then dumped the /etc HFS to a sequential data set using the DFSMSdss dump utility. You'll only need to do this step one time per ServerPac. We used this customized /etc HFS as a base /etc for our system unique versions.
  2. On each system, restored the /etc HFS from the sequential data set using the DFSMSdss restore utility with the RENAMEU option. RENAMEU lets you rename the PDS to correspond to the HFS naming conventions. For naming convention guidelines, see "HFS Naming Conventions" on page 130. You can customize the system-unique /etc HFS on each system, if necessary, according to product documentation.

     Now, as each system is brought up, the system-unique /etc HFS will be mounted read-write, at the /etc mount point.

Initializing the /dev HFS: For systems at the OS/390 R7 level or above , you can simply mount an empty HFS at the system unique /dev mount point. OS/390 will create the character special files as needed. This works in either a ServerPac or CBPDO installation environment.

Initializing the /tmp HFS: HFS contains temporary data used by products and applications. There is no need to initialize /tmp — It will populated by the products and applications that use it.

Initializing the /var HFS: The /var directory contains dynamic data used internally by products and by products, elements, and features of OS/390. You may need to copy your existing /var directory files into the new system specific /var HFSs. Do this as follows for each system:

- First, create the HFS for /var according to our naming standards. (See "HFS Naming Conventions" on page 130.)
- Now mount the new HFS to a temporary mount point.
- Copy your existing files from the old /var directory into the new /var HFS.
- Once the files are copied from the old to the new /var, you can un-mount the new /var HFS from the temporary mount point and remount it at /var.

Using the Automount Facility to Manage User Home Directories

We use the automount facility to manage our user home directories in /u. Before OS/390 R9, we created a small (one track) HFS for each user's home directory on every system in our sysplex. With 16 systems in our sysplex, each user had one HFS per system, 16 HFSs per user, for our 100 OS/390 UNIX users. This gave us a total of about 1600 user HFSs. See our December 1999 edition for how we did this.

Now, in the R9 shared HFS environment we simply create one HFS per user per participating group. This works as it did before — Automount automatically mounts HFSs according to an automount policy you put in place. Note that we no longer use NFS support with automount; we now mount the HFSs directly.

With R9 shared HFS, you can only have one automount policy for the sysplex, so every /etc/u.map file must be identical (there's one for each system). These must match because they contain the naming convention for the user HFSs for the entire sysplex.

Performance Considerations for Shared HFS

During our OS/390 R9 testing, we did some testing of HFS performance. We focused on the performance difference between file systems processed locally and remotely. Local processing means that processing (such as a copy, read, or write) is done on the same system where the file system is mounted. Remote processing means that the processing is done from one system and the file system is mounted at a different one.

We did our testing on system images JG0, JH0, and Z0 on our G6 Enterprise Server. We had four shared ICB channels available from our G6 Enterprise Server to the coupling facility running on a G5 Enterprise Server. See "Our Sysplex Hardware Configuration" on page 3 for details on our configuration.

If you're moving to a shared HFS environment for the first time, our figures and recommendations can help you to understand how your performance might change in a shared HFS environment. Note, however, that because your software, hardware, and workload environment are different than ours, your own performance results will probably not match ours. Our tests were run with a single user performing the tested task and no other activity on the systems. We measured the performance of a large number of read and write operations — up to 1000

operations for small files (less than or equal to 1MB) and up to 25 operations for large files (between 1MB and 20MB). We then averaged the results.

See “Recommendations for Controlling Performance” on page 135 for a couple of recommendations to help control performance.

This topic shows only a portion of the figures and recommendations compiled for HFS performance. For additional HFS performance information, see the UNIX system services Web site at the following address:

`http://www.s390.ibm.com/unix.`

Performance of Read, Write, and Copy of File Systems

We wrote a C program that does reads, writes, and copies of file systems. We compared the performance of read, writes, and copies done for local file systems and for file systems mounted remotely. The bottom line is that copies, reads, and writes take longer for file systems mounted remotely. How much longer? The following tables shows some of the figures.

Note that in all of the following figures, the files that are less than or equal to 1MB reside in HFS cache and have been recently written to or read from.

Performance Comparisons for Reads:

Table 11. Performance Comparison of Reads Between File Systems Mounted Locally and Remotely.. Description

Size of File	Read From Local File Systems	Read From Remote File Systems
Files less than or equal to 1 MB	<.01 second	<.20 second
File greater than 1 MB	varies depending on the size of the file	72% to 200% slower

Performance Comparisons for Writes:

Table 12. Performance Comparison of Writes Between File Systems Mounted Locally and Remotely.. Description

Size of File	Write to Local File Systems	Write to Remote File Systems
Files less than or equal to 1 MB	<.01 second	<.30 second
Files Larger than 1 MB	varies depending on size of file	0% slower

Performance Comparisons for Copies: For our performance testing of copying file systems, we compared the performance for three different ways of copying:

- **Local to local** — Copying a file from a local file system (mounted on the local system) to another local file system.
- **Local to remote** — Copying a file from a local file system to a remote file system (mounted on a remote system).
- **Remote to remote** — From the local system, copying a file from one remote file system to a different remote file system mounted on a third system.

The following table shows our results:

Table 13. Performance Comparison of Copy Between File Systems Mounted Locally and Remotely

Size of File	Local to Local Copy	Local to Remote Copy	Remote to Remote Copy
File less than or equal to 1 MB	<.05 second	<.30 second	<.60 second
Files Larger than 1 MB	varies depending on size of file	5% to 50% slower	40% to 100% slower

Performance of File System Mounts

Mounting file systems across a sysplex takes longer than mounting file systems in a non-sysplex configuration. And the more participating systems there are, the longer a file system mount will take. For example:

- If you have two participating systems, file system mounts take nearly twice as long as they would in a single system sysplex.
- If you have four participating systems, file systems take nearly four times longer to mount than they would in a single system sysplex.

Recommendations for Controlling Performance

There are a couple of things we recommend for controlling performance in a shared HFS environment:

Use Automount to Manage User Home Directories: We use automount to help us manage user home directories because it ensures that a user's file systems are mounted locally (see "Using the Automount Facility to Manage User Home Directories" on page 133). This helps improve performance, since as we've already shown, reads, writes, copies, and mounts of local file systems are faster than remote ones.

Tune the Size of Your OMVS Couple Data Set: Your OMVS CDS should be large enough to accommodate all the file systems you'll have mounted at any given time in the participating group. However, the size of the OMVS CDS does impact performance, and the larger the OMVS CDS, the greater the impact. This means that you should not make your OMVS CDS larger than you currently need for the file systems you'll have mounted concurrently in your environment.

| See the UNIX system services Web site for information on tuning your OMVS CDS.

Chapter 12. Setting Up SAP R/3

In this chapter, we'll discuss how we set up Systems, Applications and Products in Data Processing (SAP) R/3 BB SR1 in our environment. (From here on, we'll refer to it as SAP R/3.) SAP R/3's suite of client/server data processing products are designed to combine different business and technical applications into a single integrated three tier application. The client/server design of SAP R/3 can free you of lots of administration tasks and configuration details concerning individual machines in the network.

Our goal was to install SAP R/3 4.0B SRI to make sure it was stable in our environment. We used the following books to set up SAP R/3 in our environment:

- IBM books:
  - *SAP R/3 Rel 4.0B on DB/2 for OS/390: Planning Guide*
  - *SAP R/3 on DB/2 for OS/390: Connectivity Guide*
- SAP Books:
  - *R/3 Installation on UNIX: DB2 for OS/390*
  - *R/3 Installation on Windows NT: DB2 for OS/390*
  - *BC SAP Database Administration Guide: DB2 for OS/390*
- IBM red books:
  - *SAP R/3 on DB2 for OS/390: Implementing with AIX or Windows NT Applications Servers*
  - *SAP R/3 on DB2 UDB for OS/390: Application Servers on OS/390*

Our SAP R/3 Configuration

The following picture shows our SAP R/3 configuration:

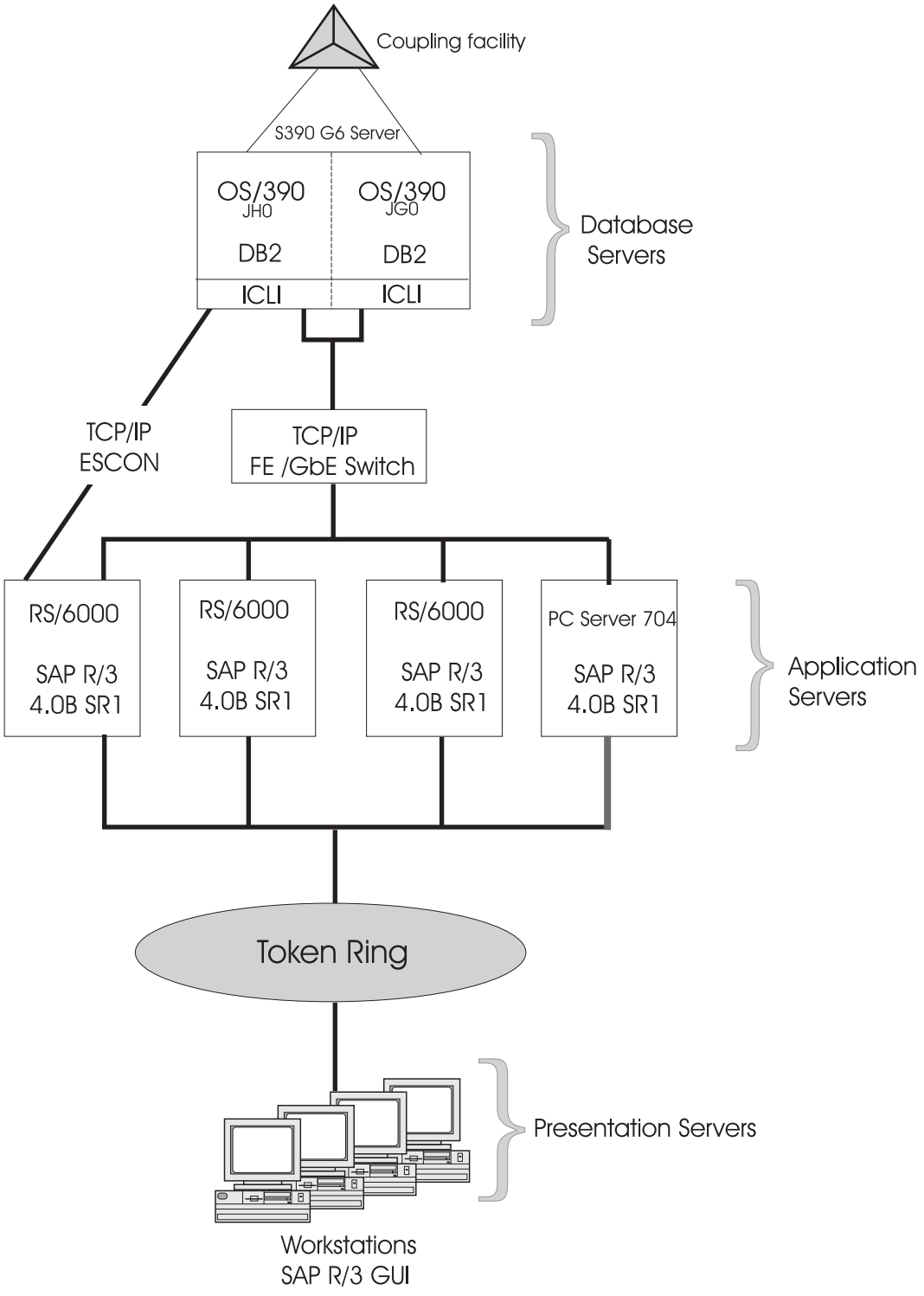


Figure 37. Our SAP Configuration

This picture shows the three tiers of our SAP R/3 configuration:

- **Database servers**: SAP R/3 uses the database server to store the data from the different application servers. We use DB2 for OS/390 as our SAP R/3 database servers. Note that using DB2 for OS/390 as a database server requires UNIX System Services.

- **Application servers**: This is the client side of the SAP R/3 configuration. SAP R/3 application servers run on the client systems where batch and interactive SAP programs run. We run our application servers on AIX (on RS/6000) and Windows NT.
- **Presentation servers**: This is the end user side of the SAP R/3 configuration. A presentation server is a SAP R/3 graphical interface running on an end user system like Windows or MacIntosh. We don't really use presentation servers in our configuration because we're running simulated workloads, which run directly on the application servers.

Note also in our picture the Integrated Call Level Interface (ICLI). ICLI is a component of OS/390 that provides a remote SQL connection between the database server and the application servers. To make sure that you have the correct version of ICLI installed on your database server, see the SAP R/3 installation notes in the *R/3 Installation on UNIX: DB2 for OS/390* and *R/3 Installation on Windows NT: DB2 for OS/390* books.

Installing the Operating System Collector (SAPOSCOL)

When we installed the SAP R/3 central instance on AIX, we also installed the operating system collector (SAPOSCOL) for OS/390. SAPOSCOL is an OS/390 UNIX System Services process that gathers selected performance data from the RMF Monitor snapshots and generates statistics based on that data. The data is stored in the shared memory where it is accessed by another OS/390 UNIX process called RFCOSCOL.

When we installed SAPOSCOL, we misinterpreted the meaning of the GWHOST key in the SAPOS390COL_IND_DB2 installation step. The *SAP R/3 on DB2 for OS/390: Implementing with AIX or Windows NT Applications Servers* book tells us to specify the host where the R/3 gateway is running for the GWHOST key. The gateway is the IP router between the database server and the application server. We wrongly interpreted **host** to mean the database server side of the gateway, where the DB2 server resides, and so we specified the IP address of the OS/390 system. We should have specified the IP address for the application server (AIX or Windows NT) side of the gateway. This error meant that we had the wrong IP address in the /saposcol/saprfc.ini file.

As a result of this error, when we tried to bring up the SAP GUI interface, we received the following message:

```
Error: Unable to bring up the SAPGUI.
```

We also found the following messages in the work directory files:

- The dev_rd work file on the application server contained the following:

```
*  LOCATION    SAP-Gateway on host oeaix15
*  ERROR       timeout during allocate
*
*  TIME        Fri May 19 13:58:11 2000
*  RELEASE     40B
*  COMPONENT   SAP-Gateway
*  VERSION     2
*  RC          242
*  MODULE      gwr3cpic.c
*  LINE        1541
*  DETAIL      no connect of TP rfcoscol from host petjg0
*  COUNTER     4
*
```

- The dev_rfc work file on the OS/390 system contained the following:

```
**** Trace file opened at 20000519 142453 EDT, REL 40A, VER 3
resize I/O buffer to 16000 bytes
Error RFCIO_ERROR_SYSERROR in ./rfccpix.c : 1285
Connect to SAP gateway failed
Connect_PM  TPNAME=COLM, GWHOST=<IP Address>, GWSERV=3301

LOCATION    CPIC (TCP/IP) on local host
ERROR       partner not reached  (host <IP Address>, service 3301)
TIME        Fri May 19 14:24:53 2000
RELEASE     40A
COMPONENT   NI (network interface)
VERSION     30
RC          -10
MODULE      ./niuxi.c
LINE        798
DETAIL      NiPConnect
SYSTEM CALL connect
ERRNO       1128
ERRNO TEXT  EDC8128I Connection refused.
```

- The saprfc.ini work file on the OS/390 system contained the following:

```
/* Type G:  Program  is registered at a SAP gateway
      DEST=COLM
      TYPE=R
      GWHOST=<IP Address>
  TPHOST=petjg0
     TPNAME=/sap/SAP40bSR1/saposcol/rfcoscol
     RFC_TRACE=1
```

Once we corrected the error by specifying the IP address for the application server side of the gateway, we were able to bring up and use the SAP GUI without error.

Chapter 13. Migrating the OS/390 Security Server LDAP Server to OS/390 R10

The OS/390 Lightweight Directory Access Protocol (LDAP) Server is a component of the OS/390 Security Server, which provides client access to LDAP directories. The OS/390 LDAP Server uses the LDAP standard, an open industry protocol for accessing information in a directory. In this chapter, we'll describe how we migrated the OS/390 LDAP Server from OS/390 R8 to OS/390 R10.

We used the following to perform the migration:

- *OS/390 SecureWay Security Server LDAP Server Administration and Usage Guide*
- *OS/390 SecureWay Security Server LDAP Client Application Development Guide and Reference*
- /usr/lpp/ldap/examples/sample_server/README — The README file was essential for setting up the LDAP sample server.

For information on how we originally installed the OS/390 LDAP Server and our LDAP configuration, see our December '99 edition.

Migrating to the R10 level of the O/390 LDAP Server went smoothly using the books mentioned above. We have the following experiences to share:

Customizing Our Automatic IPL Startup for the OS/390 LDAP Server: We customized our automatic startup of the OS/390 LDAP Server using SA for OS/390. SA for OS/390 was used to ensure that RRS was initialized before the OS/390 LDAP Server starts to come up. When we first automated LDAP using the autolog feature in the TCP/IP profile, the OS/390 LDAP Server would often come up before RRS had initialized. We thought that was fine, because we received a message indicating that the OS/390 LDAP Server was up:

```
Slapd is ready for requests
```

However, when we tried to issue LDAP commands, we received an SQLAllocate error. If you encounter this error, you can stop and then restart the OS/390 LDAP Server and then make sure that RRS initializes before the OS/390 LDAP Server in your automatic IPL startup sequence.

Migrating to the New DB2 Backend Database

We migrated to the new DB2 backend database, TDBM. The TDBM database backend, which is new for OS/390 R10, is based on DB2 and is a more scalable database implementation than the RDBM database shipped in prior releases. The OS/390 R10 LDAP server still ships with the RDBM database compatibility, but IBM recommends that you use the TDBM database backend.

We have the following experiences to report about migrating to the new TDBM database backend:

Migrate to the R10 Level of the OS/390 LDAP Server on RDBM Before Moving to TDBM: Before migrating to the TDBM backend on R10, we ran the OS/390 R10 LDAP Server with the RDBM backend. We did this to make sure that the OS/390 R10 LDAP Server worked in our current configuration before we migrated to the

new backend. We verified that the ldapsearch, ldapadd, and ldapdelete commands worked successfully with the RDBM backend on the R10 level of the OS/390 LDAP Server.

Changes to the LDAP Schema: Migrating to the TDBM backend involved changes in the schema for the LDAP database and in the way we handle the schema (see steps 2 through 4 on page 143 in the procedure below). The LDAP schema is like a template for the database — it tells the database how the data will be structured. The schema is stored as an entry in the database directory. See Chapter 15 of the *OS/390 SecureWay Security Server LDAP Server Administration and Usage Guide* for more information.

The R10 level of the OS/390 LDAP Server allows us both to search and udpate the schema in the active directory **for the TDBM backend only** using the following commands:

- ldapsearch - This command lets you search the active database directory, including the schema, and display it's contents. The manuals refer to this search and display function as schema publication. To display our LDAP schema, we entered the ldapsearch command as follows:

  ```
  ldapsearch -h <IP Address> -b "cn=SCHEMA,o=Your Company" -s base -V 3 "objectclass=*"
  ```

  cn=Schema shows the common name of our schema.
- ldapmodify - This command lets you update the active database directory, including the schema.

For information on the ldapsearch and ldapmodify commands, see the *OS/390 SecureWay Security Server LDAP Client Application Development Guide and Reference*

Migrating Data for the TDBM Backend: Migrating our data to the TDBM backend involved both updating the schema for the TDBM database and loading the data into the database itself. We did the following to migrate our data from RDBM format to TDBM format:

1. Our first step in migrating our data to TDBM was to use the db2ldif utility to dump the existing data from our RDBM database into a file in LDAP Data Interchange Format (LDIF). (Note that you cannot use db2ldif to dump data from a TDBM database - you would use tbdm2ldif for that purpose.) Later, in step 5 on page 143, we'll load this data into the LDAP database directory.

   Note that if you already have a backup of your data in LDIF format you do not need to use db2ldif to dump your data.
2. Next we updated the schema for the TDBM database. Using the TDBM backend required that we add the attribute types for the user defined attributes to the schema for the TDBM database and the schema must be complete before we load the data into the database, because the schema acts as a template for the data in the database. To add the attribute types to the schema, we issued the ldapmodify command to update the active directory schema as follows:

   ```
   ldapmodify -v -f /ldapmod -h <IP Address> -D \
   "cn=LDAP Administrator,o=Your Company, c=US" -w secret
   ```

   cn=LDAP Administrator references the adminDN value found in the slapd.conf file . **/ldapmod** is the name of an LDIF format file containing the attributes and looks as follows:

   ```
   cn=SCHEMA,o=Your Company
   +attributetypes=( 9.9.9.1 NAME 'Inscobadlogoncount' SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
   USAGE userApplications )
   +attributetypes=( 9.9.9.2 NAME 'InscoAcctLocked' SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
   SINGLE-VALUE USAGE userApplications )
   ```

Here, cn= shows the common name of the schema, SCHEMA.

3. To verify that the attributes that we just added to the schema are there, we use the ldapsearch command to display the schema:

```
ldapsearch -h <IP Address> -b "cn=SCHEMA,o=Your Company" -s base -V 3 "objectclass=*"
```

4. Next we added an objectclass, InscoPerson, to the schema. Each database entry is associated with an objectclass, which determines the set of attributes in the entry. InscoPerson is the objectclass for all our application users. To add the InscoPerson objectclass to the schema, we used the ldapmodify command as follows:

```
ldapmodify -v -f /ldapaddobj -h <IP Address> -D \
"cn=LDAP Administrator,o=Your Company, c=US" -w secret
```

Our **/ldapaddobj** LDIF format file containing the objectclass looks as follows:

```
cn=SCHEMA,o=Your Company
+objectclasses=( 9.9.9.0 NAME 'InscoPerson' DESC 'Used as auxiliary Object Class to define
additional attributes belonging to the structural class
organizationalPerson' SUP top Auxiliary MAY ( Inscobadlogoncount $ InscoAcctLocked ) )
```

Originally, we tried to add a structural object class for our application users. This didn't work - the object class for our application users must be **Auxilliary** so that we can modify it to add the user defined attributes to it in step 6. For more information on objectclasses, see the chapter on directory schema in *OS/390 SecureWay Security Server LDAP Server Administration and Usage Guide*.

5. Once we had updated the schema in the active directory with the attributes and object class that we want for our application users, we then loaded our existing RDBM data, obtained in step 1 on page 142, into the TDBM database using the following ldapadd command:

```
ldapadd -h <IP address> -D "cn=LDAP Administrator, o=Your Company, c=US" -w secret -f add.file
```

add.file contains the LDIF format data obtained in step 1 on page 142. The data that we're loading includes the entries for each of our application users.

6. Our next step was to update our database to associate an objectclass (InscoPerson) and values for the user defined attributes (Inscobadlogoncount and InscoAcctLocked) with our existing directory entries for each of our application users. For each user, we issued the ldapmodify command as follows:

```
ldapmodify -v -f  /ldapaddobj4 -h <IP Address> -D \
"cn=LDAP Administrator, o=Your Company, c=US" -w secret
```

Our **/ldapaddobj4** file associates the InsoPerson objectclass with user **c00002** and associates values with the user defined attributes as follows:

```
cn= c00002 , ou=Insco, o=Your Company
+objectclass=InscoPerson
+inscobadlogoncount=0
+inscoacctlocked=FALSE
```

7. Now, we'll search for user c00002 in the schema with the following ldapsearch command:

```
ldapsearch -h <IP Address> -b  "ou=Insco, o=Your Company" "cn=c00002"
```

We'll get back the following output:

```
cn=c00002, ou=Insco, o=Your Company
objectclass=InscoPerson
objectclass=organizationalPerson
objectclass=PERSON
objectclass=TOP
cn=c00002
```

```
sn=c00002
ou=Insco
description=Client  for e Connectors
```

Note that the new attribute values we just added for user c00002 do not show up in this output. However, we were able to display the values for these attributes by issuing a more specific command, as follows:

```
ldapsearch -h <IP Adress> -b  "ou=Insco, o=Your Company" "cn=c00002" Inscobadlogoncount
Inscoacctlocked
```

LDAP displayed the values for these attributes:

```
inscobadlogoncount=0
inscoacctlocked=FALSE
```

The fix for APAR OW45412 resolves this problem so that attributes associated with a user are displayed in ldapsearch command output. Note that If you don't install the fix for APAR OW45412, your applications may fail because they are not receiving the output they expect from the ldapsearch command.

Viewing Data In Our TDBM Database Directory: The easiest way we found of looking at the data in our TDBM database was to use the new tdbm2ldif utility to dump the data into an LDIF format file as follows:

```
tdbm2ldif -o tdbmout
```

We use tdbm2ldif rather than ldapsearch to do this because tdbm2ldif puts the data into a file in LDIF format, which is convenient for later reuse in ldapmodify and ldapadd commands issued against the TDBM database.

To view the schema in the directory, we used the ldapsearch utility as in the examples above. Using the following ldapsearch command syntax, you can put your output data into a data set called schemaout:

```
ldapsearch -h <IP Address> -b "cn=SCHEMA, 0=Your Company" \
-s base -V 3 "objectclass=*" >schemaout
```

Part 3. Systems Management

Chapter 14. Parallel Sysplex Automation 147
Our Automation Setup 148
An Overview of Our Automation 149
Running a Dual NetView Environment 151
Using Command Lists (CLISTs) 151
Setting Trigger Conditions for a Subsystem 152
Displaying Status of a Trigger Condition 153
Displaying or Overriding a Service Period 154
Exploring the Sysplex Automation Functions 156
Parallel Sysplex Operation Center 156
DISPXSYS Command 157
DISPCF Command 160
DISPCDS Command 161
DISPCONS Command 162
Coupling Facility Drain 163
Experiences With SA OS/390 Sysplex Automation 165

Chapter 15. Introducing the Tivoli Management Framework for OS/390 167
Tivoli Terminology Transition 167
How OS/390 Fits Into the Tivoli Environment 168
Tivoli Management Framework for OS/390 Server and Gateway 169
Our Hardware Configuration 170
Planning, Prerequisites, and System Configuration 170
Installing Java on OS/390 170
Installing the Tivoli Software 171
Configuring OMVS Environment Variables 173
Configuring OMVS System Variables 173
Installing the TME 10 Desktop for Windows Version 3.6.1 on Windows NT 4.0 173
Installing Java Virtual Machine and X Server on the NT 4.0 Workstation 174
Understanding HFS DASD Space Usage 174
Our Experiences With Tivoli Management Framework for OS/390 Server and Gateway 174
Running the wserver Command 174
Warning About /etc/services Problem 175
Resolving the ++HOLD Action for PTF UW61819 175
Running Tivoli CLI Commands from OMVS 176
Connecting to the OS/390 Tivoli Management Server Through the Tivoli Desktop 176
Problem Starting Tivoli oserv and lcfd Daemons Under OS/390 V2R9 177
Tivoli HFS Service Procedure 177
Software Installation Service 178
User Administration 181
Security Management 182
Software Distribution 184
Distributed Monitoring 185
Inventory 186
Tivoli Management Framework for OS/390 Framework Endpoint 188
Our Hardware Configuration 188
Our Installation Experiences 190
Figuring Out Which Code To Install On Which Machine 190
Choosing Ethernet or Token Ring 191
Setting Up and Running the lcfd Daemon (or OS/390 Endpoint) 191

Confirming Your Installation Using a Web Browser 194
Experiences with TME 10 User Administration for OS/390 195
Planning to Manage RACF User IDs in Tivoli 195
Setting Up Profiles on the TMR Server 196
Populating the Tivoli Database with wpopusers 197
Our Operational Experiences 199
Experiences with TME 10 Security Management for OS/390. 200
Where to Find More Information 203

The above chapters describe the systems management aspects of our computing environment.

Chapter 14. Parallel Sysplex Automation

In this chapter, we describe how we use automation to more efficiently operate our sysplex from a single point of control, and to automate the startup, shutdown, and restart of many of our major subsystems and applications.

We began writing about Parallel Sysplex automation in our 1997 test reports. At that time, we were just beginning to use NetView and System Automation for OS/390 (then called SA/MVS) to more efficiently operate our sysplex. We were running NetView V3R1 and SA/MVS V1R2.

We currently run TME 10 NetView V1R2 and System Automation for OS/390 (referred to as SA OS/390) V1R3. Our SA OS/390 now also includes the sysplex automation enhancements that IBM delivered in October, 1999, as an SPE in APAR OW39485. This chapter focuses on these new functions. See our previous test reports for details about our basic automation setup and migration.

As documented in *System Automation for OS/390 Planning and Installation*, SA OS/390 is a NetView-based product that integrates the functions of three established licensed programs:

- Automated Operations Control/MVS (AOC/MVS)
- Enterprise System Connection (ESCON) Manager (ESCM)
- Target System Control Facility (TSCF).

Corresponding to the above programs, SA OS/390 consists of the following three components:

- System operations (SysOps)—startup, shutdown, and restart of various software resources, such as subsystems, applications, and components. You can also use SysOps to startup the other two components of SA OS/390 (listed below), although currently we're not set up that way.
- I/O operations (I/O Ops)—monitoring and control of I/O devices such as DASD, tape, and switches.
- Processor operations (ProcOps)—remote access to the hardware management console (HMC) and the support element (SE) to control 9672 model processors, and the processor controller element (PCE) to control ES/9000 model processors.

The key point about SA OS/390 V1R3 is that it uses XCF to communicate between systems, so you no longer need to log onto all systems to manage your sysplex. You can control and operate the sysplex from any member system running SA OS/390 V1R3. Each system running SA OS/390 V1R3 in the sysplex knows about all the resources of all the other systems in the sysplex. ***Because SA OS/390 uses XCF, if VTAM is down on system X, but SA OS/390 is up on system X, you can still monitor and control resources on system X from another system in the sysplex.***

We used many of the SA OS/390 and NetView manuals to set up our automation. See "Appendix D. Useful Publications and Web Sites" on page 217 for a complete list. We also found lots of useful information at the following Web site:

`http://www.ibm.com/s390/products/sa`

Our Automation Setup

We run NetView and SA OS/390 on each production system in our sysplex. System J80 serves as the primary focal point and system Z0 serves as our backup focal point. As explained in *System Automation for OS/390 Planning and Installation*, the *focal point* system is the central point for customization, monitoring, and control of a number of *target systems*. In our case, the target systems are all the systems in our sysplex.

We can log on to system J80 using a NetView console and monitor all systems in our sysplex from a single point. We can log on to any system in the sysplex from a single NetView console and issue commands to control resources on any system in the sysplex. In this way, the NetView console allows us to rely less on the MVS system console.

An important point to note is that, even though SA OS/390 V1R3 uses XCF signalling, the primary and backup focal points still depend on VTAM to get status from the target systems. If VTAM is lost on a target system, the focal points no longer know that system's status, but you could still check the target system's status from the focal point, if NetView is up. If NetView is also lost, you can log on to any other system that has NetView up and running and check the status of the target system's other subsystems that way.

High-Level Process to Implement Automation: Our high-level process to implement automation was as follows:

1. Bring up automation on our test system (Z1), followed by the focal point system (J80), followed by a target system.
2. Establish communication between the target system and the focal point system. This involves setting up gateway autotasks, which are explained in *System Automation for OS/390 Planning and Installation*. At this point, it's handy to have your VTAM system programmer around.
3. Bring automation up on the remaining systems, one at a time, establishing communication between each target system and the focal point.
4. Set up the backup focal point system (Z0) and establish communication between each target system and the backup focal point system.

To date, we have focused our efforts on transitional automation using the SysOps portion of SA OS/390 to automate the initialization, startup, restart, shutdown, and recovery of various subsystems, applications, and MVS components. We've been using the SA OS/390 status display facility (SDF) to display the status of our automated resources on dynamic color-coded panels. SDF is the full-screen operator interface for monitoring automated resource status at the NetView 3270 console (see our December 1999 edition for more information about SDF and the DISPSTAT command). Transitional automation requires both the NetView and NetView subsystem interface (SSI) address spaces.

We still have work to do to implement Systemview AOC/MVS automation features such as those for CICS, IMS, and OPC, to fully implement I/O Ops and ProcOps, and to implement steady state automation (routine performance monitoring of resources), which requires the following on the focal point systems:

- NetView's resource object data manager (RODM) address space
- A batch RMF Monitor III data reporter on all monitored systems
- NetView's graphic monitor facility host subsystem (GMFHS)

- A NetView supporting the NetView graphic monitor facility (NGMF). NGMF is a NetView function that consists of a series of windows controlled by the NetView program which allows you to monitor the network interactively and to monitor the SA OS/390 enterprise graphically.

Migrating to SA OS/390 V1R3: To migrate from SA/MVS V1R2 to SA OS/390 V1R3, we followed the migration path in *System Automation for OS/390 Planning and Installation*. Note that if you have gateway operator IDs defined in your automation network, you must add them to the automation policy *after* migrating to the new release.

Sysplex Automation Enhancements: We installed the Parallel Sysplex automation enhancements that were delivered in October, 1999, as an SPE to SA OS/390 V1R3. See the description of APAR OW39485 for details about the SPE. Documentation for this SPE appears under the “Latest News” topic on the System Automation Web site at

`http://www.ibm.com/s390/products/sa`

We describe our initial experiences with these new functions in “Exploring the Sysplex Automation Functions” on page 156

An Overview of Our Automation

Figure 38 on page 150 shows a high-level illustration of our NetView and SA OS/390 automation environment:

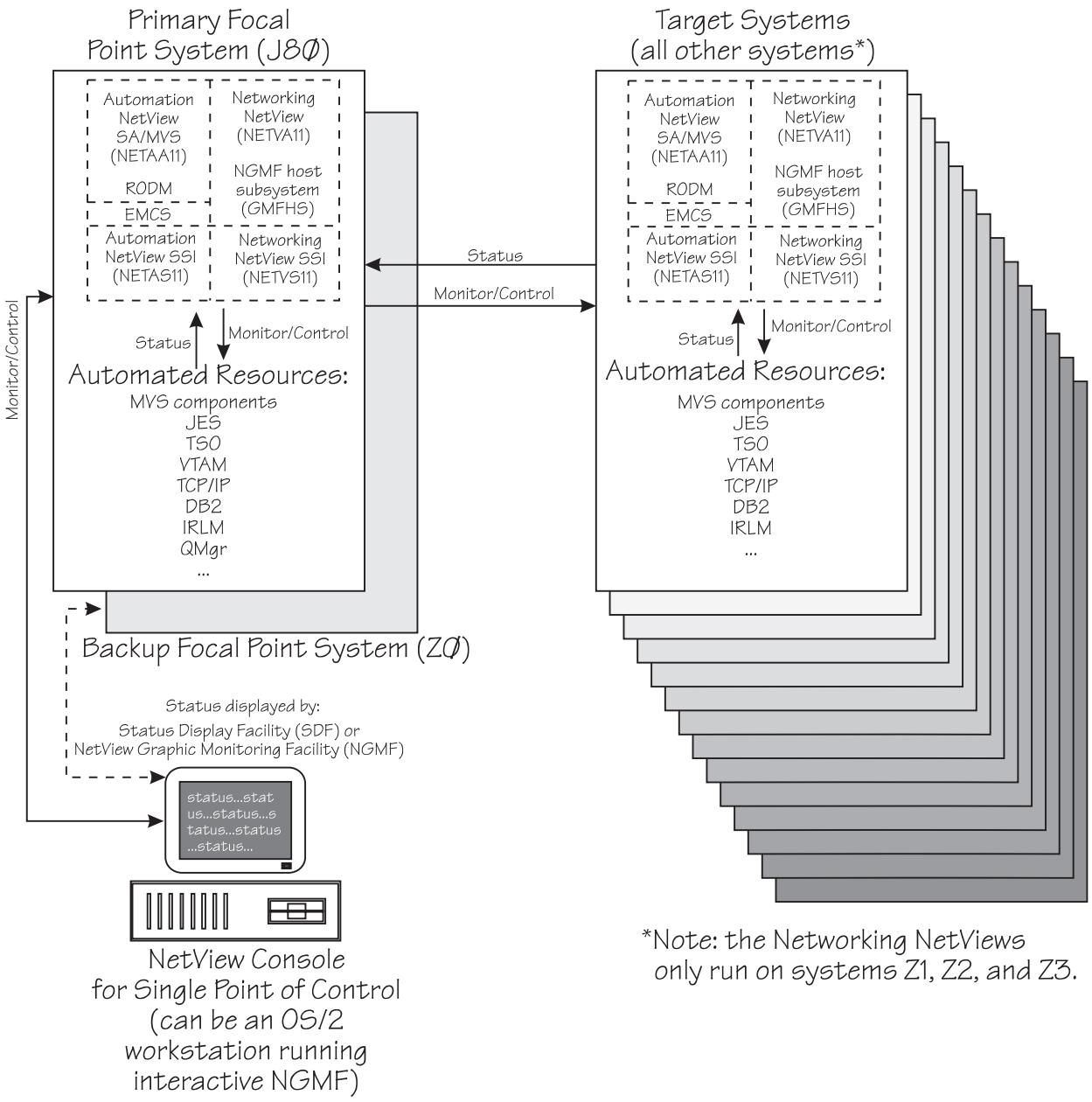


Figure 38. Our Automation Setup

Note the following about Figure 38:

- We have a dual NetView environment on systems J80, Z1, Z2, and Z3. See “Running a Dual NetView Environment” on page 151 for an explanation.
- Each target system monitors and controls its own resources, but focal point system J80 monitors all systems. Each target system reports its status to system J80, which uses NGMF to report status to the graphical workstation. Each target system can see the status of any of the other target systems in the sysplex (using the TARGET parameter available on many commands, such as DISPSTAT), but they do not forward status to one another, nor to the graphical workstation.
- From the NetView console, an operator can log on to any system to correct a problem on any other system.

  An example of a command you can issue to control a resource is:

```
SETSTATE TCPIP,RESTART,START=YES,TARGET=JC0
```

The above command restarts TCP/IP on system JC0. You can issue this command from any system in the sysplex.

- Commands must go through a subsystem interface (SSI)—each NetView has its own SSI. NETAA12 is the job name for the automation NetView, and NETAS12 is its SSI. NETVA12 is the job name for the networking NetView, and NETVS12 is its SSI. (These job names are user-defined.) We use extended MCS consoles (EMCS) for messages.

Running a Dual NetView Environment

We run a dual NetView environment on systems J80, Z1, Z2, and Z3, which consists of the following:

- A networking NetView (which performs network management functions, and where NGMF runs)
- An automation NetView (which manages automation functions, and where SA OS/390 runs)

These two NetViews run in separate address spaces. The separation of the networking functions from the automation functions can improve performance.

In a dual NetView environment, the automation NetView handles all automation-related messages and alerts while the networking NetView handles all VTAM and other networking messages and alerts. Because we have a heavy dependence on VTAM, our team member who is responsible for automation works closely with our VTAM system programmer.

Figure 39 illustrates the message and command flows in a dual NetView environment:

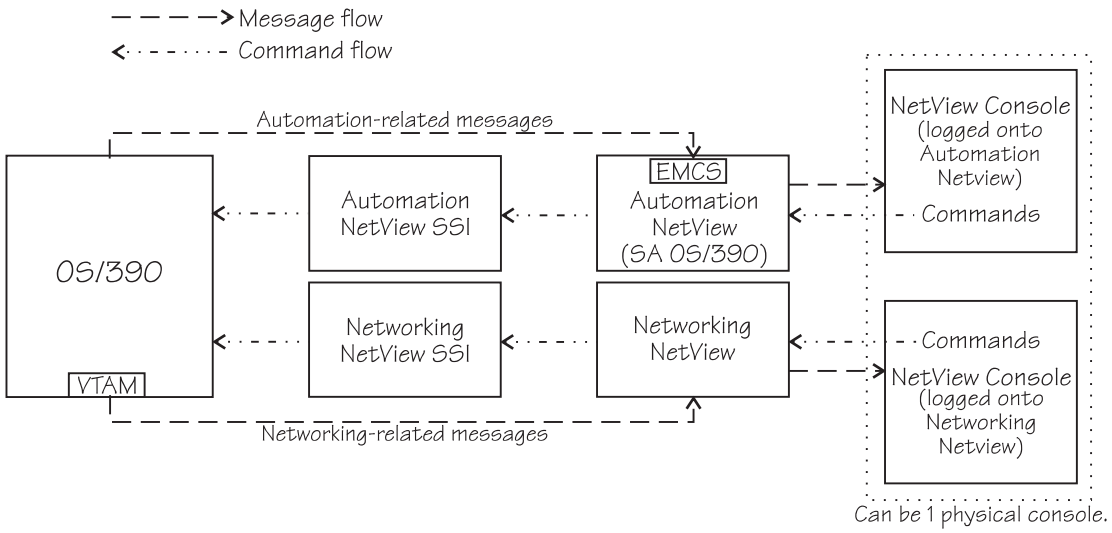


Figure 39. Dual NetView Environment

Using Command Lists (CLISTs)

One of the advantages in a NetView and SA OS/390 environment is the ability to use command lists (CLISTs). You don't have to log on to TSO to execute a CLIST. Instead, you can execute a CLIST from NetView or you can do so through automation.

You can also assign one or more MCS consoles to NetView autotasks, and allow CLISTs and NetView commands to be issued from the console using the NetView designated character.

We have a data set called OPSYS.NV12.SA13.CLIST where we keep a number of CLISTs to execute various functions. That data set is part of the DSICLD concatenation in the NETAA12 procedure, which is the procedure that starts our automation. These CLISTs can be invoked from NetView or can be started by SA OS/390. The following table describes the CLISTs we use. This might give you some ideas for CLISTs you could use in your own environment. The contents of each of these CLISTs is on the ″Samples″ page of our Web site.

Table 14. CLISTs We Use With NetView and SA OS/390.

CLIST Names	Example Invocation	Description
STAT *systemname*	STAT J80	Displays the subsystems for the system we specify.
DB2	DB2	Displays the DB2 subsystems for the systems in the sysplex.
DS *subsystem*	DS VTAM	Displays the particular subsystem for each system on which that subsystem runs.

Setting Trigger Conditions for a Subsystem

SA OS/390 allows you to set trigger conditions to start or stop a subsystem at certain times or when certain events happen. As documented in *System Automation for OS/390 Defining Automation Policy*, you can use customization dialogs to specify events or time intervals (service periods) when you want subsystem startup or shutdown to occur.

We use triggers with a service period condition to automate the startup and shutdown of our DB2 subsystems. We normally run DB2 in a 4-way data sharing group from 6:00 a.m. to 11:59 p.m. everyday except Thursdays. On Thursdays, we run a 12-way data sharing group. We have one trigger that starts and stops DB2 on the four systems where it runs daily, and another trigger that starts and stops DB2 on the other eight systems only on Thursdays. To set up the latter trigger, we performed the following steps:

1. Using the SA Customization Dialog, we defined a new service period policy object and named it DB212WAY. For the SERVICE WINDOWS policy item, we specified a start time of `0600` and a stop time of `2359` for Thursday only, and `DOWN` for all other days of the week.
2. We defined a new trigger policy object and named it the same as our service period policy object (DB212WAY). For the CONDITIONS policy item, we selected the trigger's SERVICE PERIOD condition and specified `DB212WAY` (the name of our new service period policy from step 1).
3. We had our DB2 startup and shutdown processing defined in an application policy named DB2_12WAY. To link our new DB212WAY trigger to our DB2_12WAY application, we did the following:
   - We selected the TRIGGER policy on the **SA OS/390 Policy Selection** panel for our application.
   - On the **SA OS/390 Trigger for Application** panel, we selected the DB212WAY trigger to link it to our application.

We use this scenario as the basis for the examples in the remainder of this section.

Displaying Status of a Trigger Condition

You can display the status of a trigger condition by typing H next to the associated subsystem on the DISPSTAT display. In the following example, typing H beside the first entry for DB2_12WAY (for system JA0) displays a panel showing the defined trigger conditions.

```
AOFKSTA5               SA OS/390 - Command Dialogs      Line  1    of 13
Domain Id   = PETJ8     -------- DISPSTAT ----------     Date = 01/17/00
Operator Id = BOBBYG                                    Time = 11:33:08
 A setauto  B setstate  C shutsys  D thresholds  E explain  F info  G tree
 H trigger  I service  J all children  K children  L all parents  M parents
CMD  SUBSYSTEM    STATUS    SYSTEM   JOB NAME  A I S R D RS TYPE     SHUTDOWN
---  -----------  --------  -------- --------  ------------ -------- --------
 _   DB2_12WAY    AUTODOWN  JA0      DBA1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  JB0      DBB1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  JC0      DBC1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  JD0      DBD1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  JE0      DBE1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2          UP        JF0      DBF1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  JG0      DBG1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2          UP        JH0      DBH1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  JI0      DBI1MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2          UP        J80      DB81MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2PMDCS     UP        J80      DB2PMDCS  Y Y Y Y Y Y  MVS      --none--
 _   DB2_12WAY    AUTODOWN  J90      DB91MSTR  Y Y Y Y Y Y  MVS      --none--
 _   DB2Z0        UP        Z0       DBZ1MSTR  Y Y Y Y Y Y  MVS      --none--

Command ===>
 PF1=Help    PF2=End      PF3=Return               PF5=Bottom     PF6=Roll
             PF8=Forward  PF9=Refresh  PF10=Next   PF11=Filters   PF12=Retrieve
```

Figure 40. SA OS/390 DISPSTAT Panel

The **Trigger Condition List** panel shows that there are two trigger conditions defined for DB2_12WAY on system JA0. Typing S beside a trigger condition displays its status.

```
AOFLT000                SA OS/390 - Command Dialogs         Line  1     of 2
Domain Id   = PETJ8       - Trigger Condition List --          Date = 01/17/00
Operator Id = BOBBYG                                          Time = 13:08:51

Subsystem  . . . ==> DB2_12WAY
System . . . . . ==> JA0         System name, domain ID or sysplex name
Trigger  . . . . . : DB212WAY
Service Period . . : DB212WAY           Status . . . . . : AUTODOWN
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Enter S to select a trigger condition.

  Type      Condition names
_ STARTUP   SERVICE
_ SHUTDOWN  SERVICE



Command ===>
F1=Help      F2=End       F3=Return                            F6=Roll
                          F9=Refresh                          F12=Retrieve
```

Figure 41. SA OS/390 Trigger Condition Selection List

The date and time on the panels shows that we performed this example in the early afternoon on January 17, 2000—which happened to be a Monday. (Recall that we only run 12-way DB2 on Thursdays.) Therefore, if we select the STARTUP

condition, the following panel shows that its status is currently OUTSERVICE. This means that the STARTUP condition has not been met.

```
AOFLT100                 SA OS/390 - Command Dialogs         Line  1     of 1
Domain Id   = PETJ8      - Trigger Condition List --          Date =01/17/00
Operator Id = BOBBYG                                         Time =13:09

Subsystem  . . . . : DB2_12WAY
System . . . . . . : JA0
Trigger  . . . . . : DB212WAY          Condition Type . . : STARTUP
Service Period . . : DB212WAY          Status . . . . . . : AUTODOWN
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Name      Status     Auto   Description
                     unset
SERVICE    OUTSERVICE


Command ===>
F1=Help      F2=End       F3=Return                          F6=Roll
                         F9=Refresh                        F12=Retrieve
```

Figure 42. STARTUP Trigger Condition Status

If we then select the SHUTDOWN condition, we see that its status is INSERVICE. This means that the condition has been met.

```
AOFLT100                 SA OS/390 - Command Dialogs         Line  1     of 1
Domain Id   = PETJ8      - Trigger Condition List --          Date =01/17/00
Operator Id = BOBBYG                                         Time =13:10

Subsystem  . . . . : DB2_12WAY
System . . . . . . : JA0
Trigger  . . . . . : DB212WAY          Condition Type . . : SHUTDOWN
Service Period . . : DB212WAY          Status . . . . . . : AUTODOWN
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Name      Status     Auto   Description
                     unset
SERVICE    INSERVICE


Command ===>
F1=Help      F2=End       F3=Return                          F6=Roll
                         F9=Refresh                        F12=Retrieve
```

Figure 43. SHUTDOWN Trigger Condition Status

On a Thursday afternoon, the status of these two conditions would be the opposite. The STARTUP condition would be INSERVICE and the SHUTDOWN condition would be OUTSERVICE.

Displaying or Overriding a Service Period

You can display or override the details of a service period by typing `I` beside the associated subsystem on the DISPSTAT display. For example, typing `I` beside the first entry for DB2_12WAY (for system JA0) displays the **Service Periods** panel:

```
AOFLP000                 SA OS/390 - Command Dialogs
Domain Id   = PETJ8      ----- Service Periods -----          Date = 01/17/00
Operator Id = BOBBYG                                         Time = 13:10:29

Subsystem  . . . ==> DB2_12WAY
System . . . . . ==> JA0         System name, domain ID or sysplex name
Trigger  . . . . . : DB212WAY
Service Period . . : DB212WAY

Start date . . . ==> 01 / 17 / 00 (MM/DD/YY)

Select an option ==> _

     1. One day schedule

     2. Seven day schedule

     3. Overrides to schedules



Command ===>
F1=Help      F2=End        F3=Return                          F6=Roll
                                                            F12=Retrieve
```

Figure 44. SA OS/390 Service Periods Selection Panel

Selecting option 1 displays the current day's service period schedules, which you can override by typing new start or stop times:

```
AOFLP100                 SA OS/390 - Command Dialogs
Domain Id   = PETJ8       Service Periods Schedules          Date = 01/17/00
Operator Id = BOBBYG                                         Time = 13:10:44

Subsystem  . . . . : DB2_12WAY
System . . . . . . : JA0
Trigger  . . . . . : DB212WAY
Service Period . . : DB212WAY

Overtype to modify or type D to delete the overrides.

              Start/Stop   Start/Stop   Start/Stop   Start/Stop   Start/Stop
Mon 01/17/00   DOWN         DOWN        ____ ____    ____ ____    ____ ____   ____ ____



Command ===>
F1=Help
```

Figure 45. SA OS/390 Display or Override Current Day's Service Period Schedules

Selecting option 2 on the **Service Periods** panel displays a seven-day schedule for the service period. You can override or delete an override for any day in the seven-day period:

```
AOFLP110                 SA OS/390 - Command Dialogs          More : +  -
Domain Id   = PETJ8       Service Periods Schedules           Date = 01/17/00
Operator Id = BOBBYG                                          Time = 13:11:07

Subsystem  . . . . : DB2_12WAY
System . . . . . . : JA0
Trigger  . . . . . : DB212WAY
Service Period . . : DB212WAY
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Overtype to modify or type D to delete the overrides.

                Start/Stop   Start/Stop   Start/Stop   Start/Stop   Start/Stop
  Mon 01/17/00  DOWN  DOWN   ____  ____   ____  ____   ____  ____   ____  ____
  Tue 01/18/00  DOWN  DOWN   ____  ____   ____  ____   ____  ____   ____  ____
  Wed 01/19/00  DOWN  DOWN   ____  ____   ____  ____   ____  ____   ____  ____
  Thu 01/20/00  0600  2359   ____  ____   ____  ____   ____  ____   ____  ____
  Fri 01/21/00  DOWN  DOWN   ____  ____   ____  ____   ____  ____   ____  ____
  Sat 01/22/00  DOWN  DOWN   ____  ____   ____  ____   ____  ____   ____  ____
  Sun 01/23/00  DOWN  DOWN   ____  ____   ____  ____   ____  ____   ____  ____


Command ===>
F1=Help      F2=End       F3=Return                          F6=Roll
F7=Backward  F8=Forward   F9=Refresh                        F12=Retrieve
```

Figure 46. SA OS/390 Display or Override Seven Day Service Period Schedules

Selecting option 3 on the **Service Period** panel displays any overrides that are currently in effect, and lets you modify or delete them.

Exploring the Sysplex Automation Functions

A small programming enhancement (SPE) to SA OS/390 V1R3 extends the existing Parallel Sysplex automation functions within SA OS/390. The SPE ships as the fix for APAR OW39485 (PTF UW99278).

In this section, we introduce some of the new functions provided in the SPE, including:

- Parallel Sysplex operation center
- Coupling facility drain

For complete documentation on all of the new functions, visit the SA OS/390 Web page at:

`http://www.ibm.com/s390/products/sa`

Parallel Sysplex Operation Center

To enhance the single system image and single point of control aspects of a sysplex environment, SA OS/390 provides the Parallel Sysplex operation center. This function groups together the critical sysplex resources and provides an easy interface to work with them. Using simple line commands, you can display detailed information without the need to know the exact—and often complex—operator command syntax. You can also enter commands to rebuild coupling facility structures.

To invoke the Parallel Sysplex operation center, we invoked the DISPPLEX command from NetView. Figure 47 on page 157 shows an example of the operation center's main panel on our sysplex:

```
AOFKX000                SA OS/390 - Command Dialogs
Domain ID   = PETJ8      -------- DISPPLEX --------          Date = 01/10/00
Operator ID = BOBBYG                                        Time = 09:39:32

Sysplex . . . . . : UTCPLXJ8

Select information to be displayed:

1    DISPXSYS   Display systems (including ETR & signalling paths)
2    DISPCF     Display coupling facilities
3    DISPCDS    Display couple data sets
4    DISPCONS   Display consoles



Command ===>
F1=Help      F2=End       F3=Return                            F6=Roll
                                                             F12=Retrieve
```

Figure 47. SA OS/390 DISPPLEX Main Panel

From the main panel, you can select from the following functions:

- DISPXSYS, to display the member systems, including ETR and signalling path information.
- DISPCF, to display the coupling facilities and their structures, and to rebuild structures.
- DISPCDS, to display the sysplex couple data sets.
- DISPCONS, to display information about the sysplex consoles.

We tested each of these functions and we'll show you examples of each one from our own environment.

DISPXSYS Command

The DISPXSYS command displays the target sysplex name, its GRS mode, and its member systems. For each member system, it displays the system name, system status, subsystem unscheduled maintenance (SSUM) action, SSUM interval, SSUM weight, and SFM failure detection interval. From the DISPXSYS panel, you can issue additional line commands for any system to display processor status, timer synchronization mode and ETR ports, IPL information, IOS-related configuration information, central and expanded storage sizes, and XCF signalling path information.

Figure 48 on page 158 shows an example of the DISPXSYS display on our sysplex:

```
AOFKX100                SA OS/390 - Command Dialogs          Line 1    of 16
Domain ID   = PETJ8       -------- DISPXSYS --------          Date = 01/10/00
Operator ID = BOBBYG                                         Time = 09:42:38

Sysplex . . . . . . : UTCPLXJ8
GRS Mode  . . . . . : STAR

 Display more info:  C CPU  E ETR  I IPL  O IOS  S STOR/ESTOR:
  Signalling Path :  D device  T structure
                          Monitor              ----------- SSUM -----------
Cmd  System    Status    Timestamp  INTERVAL  Action      TIME      WEIGHT
---  --------  --------  ---------  --------  ----------  --------  ------
 _   JA0       ACTIVE    09:42:23   25        ISOLATE     60        10
 _   JB0       ACTIVE    09:42:25   25        ISOLATE     60        10
 _   JC0       ACTIVE    09:42:23   85        ISOLATE     60        10
 _   JD0       ACTIVE    09:42:22   25        ISOLATE     60        10
 _   JE0       ACTIVE    09:42:24   85        ISOLATE     60        10
 _   JF0       ACTIVE    09:42:22   85        ISOLATE     60        10
 _   JG0       ACTIVE    09:42:24   85        ISOLATE     60        10
 _   JI0       ACTIVE    09:42:23   85        ISOLATE     60        10
 _   J80       ACTIVE    09:46:35   85        ISOLATE     60        10
 _   J90       ACTIVE    09:46:34   85        ISOLATE     60        10
 _   TPN       ACTIVE    09:46:34   85        ISOLATE     60        10
 _   Z0        ACTIVE    09:46:33   85        ISOLATE     60        10
 _   Z1        ACTIVE    09:46:33   85        ISOLATE     60        10
 _   Z2        ACTIVE    09:46:32   85        ISOLATE     60        10
 _   Z3        ACTIVE    09:46:35   85        ISOLATE     60        10

Command ===>
F1=Help      F2=End       F3=Return                          F6=Roll
                          F9=Refresh                        F12=Retrieve
```

Figure 48. Example of the SA OS/390 DISPXSYS Display

The T subcommand displays detailed signalling path information for all coupling facility structures. Figure 49 on page 159 shows the display that appeared when we issued the T subcommand for system JA0. Note that the panel also shows the actual MVS commands that were executed to generate the information in the display.

```
AOFKX410                SA OS/390 - Command Dialogs          Line 1    of 50
Domain ID   = PETJ8       -------- DISPXSYS --------          Date = 01/10/00
Operator ID = BOBBYG                                         Time = 13:39:09

MVS Command(s). . . : DISPLAY XCF,PATHIN,STRNAME=ALL;
                      DISPLAY XCF,PATHOUT,STRNAME=ALL
Sysplex . . . . . . : UTCPLXJ8          System  . . . . . . :  JA0
------------------------------------------------------------------------------
 STRNAME         REMOTE   PATHIN      UNUSED                LAS
                 SYSTEM   STATUS      PATHS   RETRY  MAXMSG REC
 IXCPLEX_PATH1            WORKING         0     100    1000
                 JB0      WORKING                          455
                 JC0      WORKING                          456
                 JD0      WORKING                          453
                 JE0      WORKING                          603
                 JF0      WORKING                          416
                 JG0      WORKING                          459
                 JH0      WORKING                          452
                 JI0      WORKING                          450
                 J80      WORKING                          626
                 J90      WORKING                          445
                 TPN      WORKING                          412
                 Z0       WORKING                          546
                 Z1       WORKING                          360
                 Z2       WORKING                          452
                 Z3       WORKING                          452
 IXCPLEX_PATH2            WORKING         0     100    1000
                 JB0      WORKING                          557
                 JC0      WORKING                          374
                 JD0      WORKING                          213
                 JE0      WORKING                          401
                 JF0      WORKING                          351
                 JG0      WORKING                          406
                 JH0      WORKING                          295
                 JI0      WORKING                          236
                 J80      WORKING                          907
                 J90      WORKING                          915
                 TPN      WORKING                          500
                 Z0       WORKING                          682
                 Z1       WORKING                          132
                 Z2       WORKING                          164
                 Z3       WORKING                          214
 IXCPLEX_PATH3            WORKING         0     100    1000
                 JB0      WORKING                          305
                 JC0      WORKING                          285
                 JD0      WORKING                          123
                 JE0      WORKING                          315
                 JF0      WORKING                          756
                 JG0      WORKING                          988
                 JH0      WORKING                          856
                 JI0      WORKING                          159
                 J80      WORKING                          794
                 J90      WORKING                          464
                 TPN      WORKING                          249
                 Z0       WORKING                          616
                 Z1       WORKING                           78
                 Z2       WORKING                           97
                 Z3       WORKING                          105

Command ===>
F1=Help      F2=End       F3=Return                          F6=Roll
            PF8=Forward   F9=Refresh  PF10=Next              F12=Retrieve
```

Figure 49. Example Display of the SA OS/390 DISPXSYS T Subcommand

DISPCF Command

The DISPCF command displays the coupling facilities in the sysplex, along with their space utilization and CFLEVEL information. For each coupling facility, you can display its structures and paths or remove it from the sysplex (as long as at least one coupling facility remains active).

Figure 50 shows an example of the DISPCF display on our sysplex:

```
AOFKX200               SA OS/390 - Command Dialogs      Line  1    of 3
Domain ID   = PETJ8     -------  DISPCF   ---------      Date = 01/17/00
Operator ID = BOBBYG                                    Time = 13:13:44

Cmd:  D Drain CF  P Sender paths  S display structures
CMD  CF NAME        CF SPACE UTILIZATION INFORMATION
---  --------   -Total - -Used %- -Free %- -VOLATILE-  -CFLEVEL-
 _   CF1        4019.3 M   32.71    67.13     YES          8
 _   CF2        2641.3 M   60.14    39.63     YES          9
 _   CF3         305.3 M   16.13    81.90     YES          9




Command ===>
 PF1=Help     PF2=End     PF3=Return                          PF6=Roll
                          PF9=Refresh                        PF12=Retrieve
```

Figure 50. Example of the SA OS/390 DISPCF Display

The `D` subcommand invokes the new coupling facility drain function, which we say more about later in “Coupling Facility Drain” on page 163.

The `S` subcommand displays all structures for a coupling facility. Figure 51 on page 161 shows an example of the display that appeared when we issued the `S` subcommand for CF1. Initially, the **Include condition on display** field is set to `NO`. If you type `YES` in this field and then refresh the panel, the display also shows the current condition of each structure. Note that it takes longer to build the panel display when including the structure condition.

```
AOFKX210               SA OS/390 - Command Dialogs      Line  1    of 48
Domain ID   = PETJ8     -------- DISPSTR  ---------      Date = 02/07/00
Operator ID = BOBBYG                                    Time = 10:22:56

 CFname ==>  CF1
 Include condition on display  YES  (YES/NO - Condition retrieval takes longer)

Cmd: F Force  R Rebuild  S Details
CMD  Structure name      Condition
---  -----------------   ------------------------------------------------------
 _   APPCLOG
 _   COUPLE_CKPT1         System-managed rebuild is supported.
 _   CQS_FF_LOGSTR
 _   CQS_FP_LOGSTR
 _   DFHXQLS_G2POOL1     *Rebuild is not supported.
 _   DSNDB1G_GBP0         Duplex rebuild is active.
 _   DSNDB1G_GBP4         Duplex rebuild is active.
 _   DSNDB1G_GBP5         Duplex rebuild is active.
 _   DSNDB1G_GBP6         Duplex rebuild is active.
 _   DSNDB1G_GBP7         Duplex rebuild is active.
 _   DSNDB1G_GBP8         Duplex rebuild is active.
 _   DSNDB1G_LOCK1
 _   DSNDB1G_SCA
 _   FFMSGQ_STR
 _   FFMSGQ_STR_TEST     *No connection exists. (SYSTEM-MANAGED)
 _   FPITMA1C_STR2       *No alternate CF facility defined or available.
 _   FPITMA2C_STR2       *No alternate CF facility defined or available.
 _   FPITMA3C_STR2       *No alternate CF facility defined or available.
 _   FPMSGQ_STR_TEST     *No connection exists. (SYSTEM-MANAGED)
 _   IRLMLOCK1
 _   IRRXCF00_B002
 _   IRRXCF00_P001
 _   IRRXCF00_P003
 _   ISTGENERIC           Structure's initial size is less than actual size.
 _   ISTMNPS
 _   IXCPLEX_PATH2
 _   LG2_DFHJ01
 _   LG2_DFHJ02
 _   LG2_DFHJ03
 _   LG2_DFHJ04
 _   LG2_DFHJ05
 _   LG2_DFHLGLOG
 _   NOSADI01_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSADI02_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSADI03_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSADI04_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSAIT01_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSAWH01_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSAWH02_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSAWH03_VSOSTR1    *No alternate CF facility defined or available.
 _   NOSAWH04_VSOSTR1    *No alternate CF facility defined or available.
 _   RLSCACHE01
 _   RRSLOG_ARCHIVE
 _   RRSLOG_DELAYED
 _   RRSLOG_MAIN
 _   RRSLOG_RESTART
 _   RRSLOG_RMDATA
 _   VSAMCACHE1

Command ===>
 PF1=Help    PF2=End      PF3=Return                          PF6=Roll
 PF7=Back    PF8=Forward  PF9=Refresh                        PF12=Retrieve
```

Figure 51. Example of the SA OS/390 DISPSTR Display

DISPCDS Command

The DISPCDS command displays information about the couple data sets in the sysplex (including the data set name, VOLSER, FORMAT TOD, MAXSYSTEM, MAXGROUP, and MAXMEMBER) for both the primary and the alternate data sets.

Figure 52 shows an example of the DISPCDS display on our sysplex:

```
AOFKX300               SA OS/390 - Command Dialogs      Line  1    of 45
Domain ID   = PETJ8     -------- DISPCDS  -----------    Date = 01/17/00
Operator ID = BOBBYG                                    Time = 13:40:39

  System  : J80         Interval   : 85          OPNotify : 88
  Maxmsg  : 750         Cleanup    : 15          Retry    : 100
  Classlen: 956         Max CFlevel: 9           Couplexx : COUPLE00

                    Primary                      Alternate

  Couple data set : SYSPLEX
  Dataset name    : SYS1.CDS10                   SYS1.CDS11
  VOLSER          : COUPLP                       COUPLB
  FORMAT TOD      : 08/26/1999 13:05:19          08/26/1999 13:05:40
  MAXSYSTEM       : 16                           16
  MAXGROUP(PEAK)  : 120 (80)                     120
  MAXMEMBER(PEAK) : 175 (133)                    175

  Couple data set : ARM
  Dataset name    : SYS1.ARM.CDS10               SYS1.ARM.CDS11
  VOLSER          : COUPLB                       COUPLP
  FORMAT TOD      : 08/26/1999 12:52:32          08/26/1999 12:53:01
  MAXSYSTEM       : 16                           16

  Couple data set : BPXMCDS
  Dataset name    : SYS1.OMVS.CDS10              SYS1.OMVS.CDS11
  VOLSER          : COUPL3                       COUPL4
  FORMAT TOD      : 12/02/1999 07:46:50          12/02/1999 07:46:51
  MAXSYSTEM       : 16                           16

  Couple data set : CFRM
  Dataset name    : SYS1.CFRM.CDS11              SYS1.CFRM.CDS10
  VOLSER          : CFRM03                       CFRM01
  FORMAT TOD      : 08/26/1999 13:13:42          08/26/1999 13:13:33
  MAXSYSTEM       : 16                           16

  Couple data set : LOGR
  Dataset name    : SYS1.LOGR.CDS10              SYS1.LOGR.CDS11
  VOLSER          : COUPL1                       COUPL2
  FORMAT TOD      : 08/26/1999 12:45:27          08/26/1999 12:45:33
  MAXSYSTEM       : 16                           16

  Couple data set : SFM
  Dataset name    : SYS1.SFM.CDS10               SYS1.SFM.CDS11
  VOLSER          : COUPLB                       COUPLP
  FORMAT TOD      : 08/26/1999 12:50:29          08/26/1999 12:50:29
  MAXSYSTEM       : 16                           16

  Couple data set : WLM
  Dataset name    : SYS1.WLM.CDS10               SYS1.WLM.CDS11
  VOLSER          : COUPLP                       COUPLB
  FORMAT TOD      : 08/26/1999 12:48:24          08/26/1999 12:48:28
  MAXSYSTEM       : 16                           16

Command ===>
PF1=Help    PF2=End      PF3=Return                          PF6=Roll
            PF8=Forward  PF9=Refresh                        PF12=Retrieve
```

Figure 52. Example of the SA OS/390 DISPCDS Display

DISPCONS Command

The DISPCONS command displays console information for the sysplex, including the name of the master console, WTO and WTOR buffer utilization, and number of queued messages (replies), mount requests, and operator requests. The command also displays information about each console.

Figure 53 shows an example of the DISPCONS display on our sysplex:

```
AOFKX400                SA OS/390 - Command Dialogs          Line 1    of 14
Domain ID   = PETJ8       -------- DISPCONS --------          Date = 01/17/00
Operator ID = BOBBYG                                         Time = 13:42:26

Sysplex  . . . . . . : UTCPLXJ8         Master Console . . . : J8020
Message Buffer Usage : 0 / 3000         Reply Buffer Usage . : 0 / 9999
Awaiting Replies . . : 31               Eventual Action  . . : 90
Immediate Action . . : 136              Awaiting Mounts  . . : 0
Critical Action  . . : 34               Operator Requests  . : 0

 Display more info:  D Details  R requests
Cmd  Console   Status    AUTH    NBUF  UD  Device  System    ALTGRP
---  --------  --------  ------  ----  --  ------  --------  --------
 _   JB041     INACTIVE  MASTER   N/A  N   -none-  --none--  ALTCONS
 _   JB046     INACTIVE  MASTER   N/A  N   -none-  JB0       ALTCONS
 _   JF0E0     ACTIVE    MASTER     0  N    00E0   JF0       ALTCONS
 _   JF0E1     ACTIVE    MASTER     0  N    00E1   JF0       ALTCONS
 _   J8020     MASTER    MASTER    10  Y    0020   J80       ALTCONS
 _   J8021     ACTIVE    MASTER     4  N    0021   J80       ALTCONS
 _   J8026     ACTIVE    MASTER     4  N    0026   J80       ALTCONS

Command ===>
F1=Help      F2=End       F3=Return                            F6=Roll
             PF8=Forward  F9=Refresh  F10=Next                 F12=Retrieve
```

Figure 53. Example of the SA OS/390 DISPCONS Display

Coupling Facility Drain

Coupling facility drain (CFDRAIN) provides a controlled mechanism to remove structures and connections from a coupling facility, such as when you need to remove a coupling facility for maintenance. This new function helps reduce the amount of manual intervention needed, especially for coupling facilities that contain a large number of structures.

To invoke CFDRAIN, you enter the `D` line command next to one of the coupling facilities on the DISPCF panel. Figure 54 on page 164 shows an example of the initial CFDRAIN display that appears:

```
AOFLX900                SA OS/390 - Command Dialogs          Line 1    of 48
Domain ID   = PETJ8       -------- CFDRAIN  --------          Date = 01/17/00
Operator ID = BOBBYG                                         Time = 14:03:32

Coupling Facility ==> CF1              Status . . . . . . : NORMAL
Sysplex . . . . . ==> UTCPLXJ8
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure         Condition
APPCLOG
COUPLE_CKPT1      System-managed rebuild is supported.
CQS_FF_LOGSTR
CQS_FP_LOGSTR
DFHXQLS_G2POOL1  *Rebuild is not supported.
DSNDB1G_GBP0      Duplex rebuild is active.
DSNDB1G_GBP4      Duplex rebuild is active.
DSNDB1G_GBP5      Duplex rebuild is active.
DSNDB1G_GBP6      Duplex rebuild is active.
DSNDB1G_GBP7      Duplex rebuild is active.
DSNDB1G_GBP8      Duplex rebuild is active.
DSNDB1G_LOCK1
DSNDB1G_SCA
Command ===>
F1=Help      F2=End       F3=Return                            F6=Roll
                          F9=Refresh  F10=Rebuild             F12=Retrieve
```

Figure 54. Example of the SA OS/390 CFDRAIN Initial Status Display

The Status displays NORMAL to indicate that CFDRAIN is not currently in progress. The remainder of the panel lists the structures and their current conditions. (This display is similar to the DISPSTR panel except that, from here, you cannot enter line commands against the structures.) To continue, you must press F10 (Rebuild) to request the structure rebuilds.

When you press F10, the following verification panel appears and you must press F10 (GO) again to confirm your intention and proceed with the rebuilds:
After you press F10 on the verification panel, the following message appears:

```
Domain ID   = PETJ8       -------- CFDRAIN  --------          Date = 01/17/00
Operator ID = BOBBYG                                         Time = 14:05:12

Coupling Facility ==> CF1
Sysplex . . . . . ==> UTCPLXJ8

                       R E B U I L D  Confirmation

The REBUILD process runs asynchronously on the first system in the Sysplex.
Each structure that has no * indicator in front of its status is rebuilt
status accordingly.  The structures are processed in sequence.  Once started
use the refresh PF key for getting the current status of the process.  When
more than one structure is being rebuilt a timeout occured indicating the CF
is very busy. But processing continues.  A display without any structures or
only structures that cannot be rebuilt indicates a successful completion.




Command ===>
             F2=End       F3=Return                            F6=Roll
                                      F10=Go       F11=Cancel   F12=Retrieve
```

Figure 55. Example of the SA OS/390 CFDRAIN Rebuild Confirmation Panel

```
CNM570I STARTING AUTOMATION TASK AUTXCF
```

When CFDRAIN is in progress, the Status on the CFDRAIN panel displays REBUILD, and the structure conditions indicate the rebuild progress of each structure. Pressing F9 refreshes the panel and displays updated status information. Figure 56 shows an example of the CFDRAIN panel with the rebuilds in progress:

```
AOFLX900                SA OS/390 - Command Dialogs          Line 1    of 48
Domain ID   = PETJ8       -------- CFDRAIN  --------          Date = 01/17/00
Operator ID = BOBBYG                                         Time = 14:06:21

Coupling Facility ==> CF1              Status . . . . . . : REBUILD
Sysplex . . . . . ==> UTCPLXJ8
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Structure         Condition
APPCLOG           Structure is being rebuilt.
COUPLE_CKPT1      Structure is being rebuilt.
CQS_FF_LOGSTR     Structure is awaiting rebuild.
CQS_FP_LOGSTR     Structure is awaiting rebuild.
DFHXQLS_G2POOL1  *Rebuild is not supported.
DSNDB1G_GBP0      Duplex rebuild is being stopped.
DSNDB1G_GBP4      Duplex rebuild is being stopped.
DSNDB1G_GBP5      Duplex rebuild is being stopped.
DSNDB1G_GBP6      Duplex rebuild is being stopped.
DSNDB1G_GBP7      Duplex rebuild is being stopped.
DSNDB1G_GBP8      Duplex rebuild is being stopped.
DSNDB1G_LOCK1     Structure is awaiting rebuild.
DSNDB1G_SCA       Structure is awaiting rebuild.
Command ===>
F1=Help      F2=End       F3=Return                          F6=Roll
                          F9=Refresh                        F12=Retrieve
```

Figure 56. Example of the SA OS/390 CFDRAIN Progress Display

See page 165 for more on our experiences with the CFDRAIN function.

Experiences With SA OS/390 Sysplex Automation

The following are some of our experiences to date using the new sysplex automation functions in SA OS/390:

Using the CFDRAIN Function: During our testing, an opportunity arose for us to test the CFDRAIN function in a real operational scenario. We needed to update the coupling facility control code (CFCC) on our CF2 coupling facility to accomodate some other testing we wanted to perform. Generally, this process involves loading the new code and then deactivating and reactivating the coupling facility to pick up the change. That means we had to move all the structures off of CF2 in order to free it up to perform this update.

We decided use the new CFDRAIN function to perform this task. As it turns out, our decision was a good one and we were pleased with the results. Overall, it took less than 4 hours in our environment to complete the update and get the coupling facility back in service.

We took the following steps to perform the CFDRAIN function:

1. From the main DISPPLEX panel, we selected option 2 to invoke the DISPCF panel.
2. On the DISPCF panel, we issued the D line command next to CF2 to invoke the initial CFDRAIN panel.

3. On the CFDRAIN panel, we pressed F10 to request the rebuild of all the structures.
4. We pressed F10 again on the verification panel to confirm our intention and start the rebuilds.

After that, all we had to do was monitor the progress and wait until all the structures were removed from CF2. No other operator intervention was needed.

Once there were no structures left on CF2, we deactivated it and then reactivated it. When CF2 came back up, the DISPCF panel indicated that CF2 was now at the new CFLEVEL. All that remained was to move the structures back.

We used the preference lists to determine which structures belonged back on CF2. To rebuild them back onto CF2, we issued the `R` line command next to those structures on the DISPSTR panel.

Recommended Service: You should install the fixes for the following APARs for SA OS/390 with the sysplex automation SPE installed:

- OW41745—fixes a problem where the S line command issued on the DISPCF panel does not display all structures on the coupling facility (and similarly on the DISPSTR panel).
- OW41930—fixes a problem where display responses can intermittently return incomplete or missing data.
- OW41932—fixes the following problems:
  - Inability to rebuild structures that support system-managed rebuild.
  - Message AOF905I erroneously displays for a successful structure rebuild.
  - DISPSTR shows incorrect status for the JES2 checkpoint structure.
- OW43583—fixes a problem where the DISPCF command displays message AOF760E if there is more than one inactive coupling facility in the sysplex or if you issue the command on a system that has no coupling facilities.

Chapter 15. Introducing the Tivoli Management Framework for OS/390

In previous editions, we tested and reported on the TME 10 Framework for OS/390 Framework Endpoint and the User Administration, and Security Management applications in an environment where the Tivoli Management Region (TMR) server resides on a Windows NT 4.0 workstation and the OS/390 system is an endpoint in the TMR. While we continue to report on this environment, we now also include the Tivoli Management Framework for OS/390 Server and Gateway, in which the TMR server resides on the OS/390 platform.

Note that when we first started reporting on our work with Tivoli products, we were working with the 3.6.0 level. With the Tivoli Management Framework for OS/390 Server and Gateway at the 3.6.1 level, all information concerning the Server and Gateway in this document represents our experience at that level. Also, now that version 3.6.1 of the Tivoli Endpoint for OS/390 is available and integrated into OS/390 V2R9 and V2R10, we include our experiences with the 3.6.1 endpoint.

Because of the change in levels and discrepancies in some of the documentation, we highly recommend that you closely review all Preventive Service Planning (PSP) information for Tivoli Management Framework for OS/390 Server and Gateway and its supported applications and apply all suggested service.

Before we begin, you need to understand some terminology changes that are taking place.

Tivoli Terminology Transition

You might pick up different IBM and Tivoli publications at different times that refer to the same products, or features of products, with different names. We'd like to help you avoid any confusion over these terminology changes. We'll begin with a narrative explanation, and follow with a summary table.

For the past few years Tivoli has had a product line called Tivoli Management Environment 10 (TME 10). The TME 10 product line had several components, the base component being called the TME 10 Framework. Other components were the TME 10 desktop, TME 10 management applications, TME 10 Application Extension Facility (AEF), TME 10 Event Integration Facility (EIF), and TME 10 Application Development Environment (ADE).

The TME 10 Framework is changing its name to the Tivoli Management Framework, and all other TME 10 products are changing their names from TME 10 to just Tivoli or Tivoli Enterprise. But there are still publications that refer to the TME 10 Framework, and you will also continue to see TME 10 as part of the name of some Tivoli products.

The Tivoli Management Framework provides basic system administration capabilities, as well as a set of foundation services for management applications. In general, you install the full set of Framework code on various types of UNIX-based, Windows NT, and now OS/390 systems from which you can then manage many other systems across platforms. The systems doing the managing are Tivoli Management Region (TMR) servers and managed nodes.

Tivoli Management Framework

The systems being managed run Framework endpoint code, and have been called by various terms. At one time they were called lightweight clients or, as part of the lightweight client framework, LCFs. Then they were called endpoints and, after that, Tivoli Management Agents or TMAs. They are currently called endpoints. However, you will continue to see the LCF and TMA acronyms in messages and codes and in publications. Over time, the LCF and TMA terms will be phased out and replaced by endpoint. Endpoints can be on clients running AIX, HP-UX, SunOS, Solaris, Windows 3.x, Windows 95, Windows 98, Windows NT, Netware, OS/2, OS/400, and OS/390.

There are also endpoint gateways (or just gateways) that act as a sort of middleman between the TMR server and the endpoint. The gateways control all communication with and operations on the endpoints.

You can read detailed descriptions of the terms mentioned above in the *TME 10 Framework Planning and Installation Guide* (SC31-8432).

One more terminology change we'll highlight is the change from Application Policy Manager (APM) to Topology Display Server (TDS) or just topology server and, finally, in Tivoli Global Enterprise Manager Version 2.2 and 2.3, to GEM Topology Server. In brief, the topology server gathers information from instrumented applications and uses that information to build views on a topology console where they are graphically displayed. You can read more about this in *Tivoli Global Enterprise Manager Installation User's Guide Version 2.2* (GC31-5152).

The following table summarizes the terminology changes discussed above:

Table 15. Tivoli Terminology Transition Summary

Old Term/Product Name	New Term/Product Name
TME 10 Framework	Tivoli Management Framework
Lightweight client framework (LCF); Tivoli Management Agent (TMA)	endpoint
TME 10 Framework for OS/390 Framework Endpoint	Tivoli Management Framework for OS/390 Framework Endpoint
TME 10 GEM User Administration Service; TME 10 User Administration for OS/390	Tivoli User Administration for OS/390
TME 10 Security Management for OS/390	Tivoli Security Management for OS/390
Application Policy Manager (APM)	Topology Display Server (TDS) or GEM Topology Server or topology server
Tivoli Management Environment (TME)	Tivoli Enterprise

How OS/390 Fits Into the Tivoli Environment

Tivoli now supports OS/390 not only as an endpoint in a TMR, but also as a full TMR server and gateway.

Tivoli Management Framework for OS/390 Server and Gateway: The Tivoli Management Framework for OS/390 Server and Gateway (5697-D10, Version 3.6.1, FMID H25I600), introduced in July, 1999, is now available to make an OS/390 system a full TMR server and gateway. In addition, five Tivoli applications run under the OS/390 server and gateway to manage non-OS/390 endpoints. Two of these applications, User Administration and Security Management, also manage OS/390 endpoints. The Tivoli Management Framework for OS/390 Server and Gateway and

all of the following applications that run under the OS/390 server and gateway are separately orderable and also available through ServerPac. The following are the supported Tivoli applications:

- Applications supporting non-OS/390 endpoints:
  - Tivoli User Administration for OS/390 Server (5697-ADM, Version 3.6.1 FMID H081610)
  - Tivoli Security Management for OS/390 Server (5697-SCM, Version 3.6.1 FMID H085610)
  - Tivoli Inventory for OS/390 (5697-F04, Version 3.6.1 FMID H0AN600)
  - Tivoli Software Distribution for OS/390 (5697-F03, Version 3.6.1 FMID H0AO600)
  - Tivoli Distributed Monitoring for OS/390 (5697-F05, Version 3.6.1 FMID H0AR600)
- Applications supporting OS/390 endpoints:
  - Tivoli User Administration for OS/390 Endpoint (5697-ADM, Version 3.6.1 FMIDs H085605 and J081625)
  - Tivoli Security Management for OS/390 Endpoint (5697-SCM, Version 3.6.1 FMIDs H085605 and J085625)

TME 10 Framework for OS/390 Framework Endpoint, Version 3.6: The TME 10 Framework for OS/390 Framework Endpoint version 3.6 (5697-D10, FMID H25I620) that is integrated into OS/390 V2R7 and V2R8 was introduced with OS/390 V2R6 ServerPac orders in November, 1998, and is orderable for OS/390 V1R3 through V2R6. This is the endpoint portion of the Framework code which you install on an OS/390 system. It allows you to manage an OS/390 system using the two supported Tivoli Framework-based applications (User Administration and Security Management) from a TMR server. The TMR server can be a workstation-based server or OS/390-based TMR server and gateway.

Tivoli Management Framework for OS/390 Framework Endpoint, Version 3.6.1: The Tivoli Management Framework for OS/390 Framework Endpoint version 3.6.1 (5697-D10, FMID H25I625) is orderable for OS/390 V1R3 through V2R8 and is integrated into OS/390 V2R9 and V2R10. This is the upgraded and renamed endpoint portion of the Framework code which you install on an OS/390 system. It allows you to manage an OS/390 system using the two supported Tivoli Framework-based applications (User Administration and Security Management) from a TMR server. The TMR server can be a workstation-based server or OS/390-based TMR server and gateway.

Tivoli Management Framework for OS/390 Server and Gateway

We'd like to introduce the Tivoli Management Framework for OS/390 Server and Gateway, which enables an OS/390 system to act as a full TMR server and gateway, and five supported applications. The product provides the following five applications to manage non-OS/390 endpoints:

- User Administration
- Security Management
- Distributed Monitoring
- Software Distribution
- Inventory

The product also provides the following two applications to manage OS/390 endpoints:

- User Administration
- Security Management

Our Hardware Configuration

Figure 57 illustrates our Tivoli Management Framework for OS/390 Server and Gateway configuration:

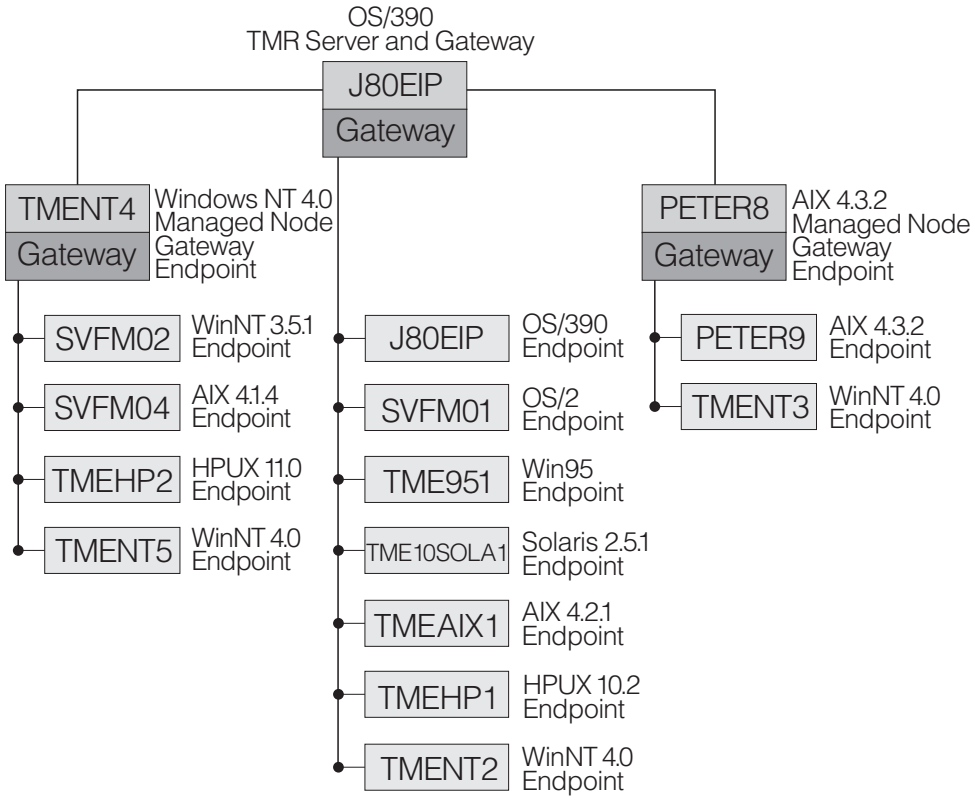


Figure 57. Our Tivoli Management Framework for OS/390 Server and Gateway Configuration

Planning, Prerequisites, and System Configuration

As with the workstation version of the Tivoli Management Framework, the installation process consists of a number of steps. During our planning phase, we used the documentation that Tivoli supplies with the products, including the program directories, release notes, supplements, and standard product documentation. We also checked the PSP buckets for any additional service or other installation requirements.

In this section, we discuss some observations from our planning and pre-installation work.

Installing Java on OS/390

In previous test reports, we reported that Tivoli requires the June 17, 1999, version of Java 1.1.6 for OS/390 and does not support any other version. This is no longer true. Tivoli now also supports Java 1.1.8. However, be aware of the following:

- Anywhere the Tivoli documentation states that you must use Java 1.1.6 for SIS, you may use Java 1.1.8 instead.
- The IBM Java for OS/390 Web site (`http://www.ibm.com/s390/java/`) states that Java 1.1.6 is no longer available as of the end of March, 2000. However, you might still be able to obtain a copy of Java 1.1.6 through Tivoli or IBM support if you prefer to use that version.

- If you currently use Java 1.1.6 and you migrate to OS/390 V2R10 from an earlier release of OS/390, our testing indicates that SIS will no longer work. Tivoli support is investigating this problem. Currently, only Java 1.1.8 functions properly with SIS on OS/390 V2R10.

For our testing, we used Java 1.1.6 on OS/390 V2R8 and V2R9. We installed Java by downloading the compressed tar file containing Java 1.1.6 from the Java for OS/390 Web site. We saved the file locally as /usr/lpp/java/hjva11d.tar.z and then issued the following command to untar the file:

```
tar -xpozf hjva11d.tar.z
```

This creates the directory /usr/lpp/java/J1.1 containing the appropriate Java files.

We mention this here because you must specify the Java directory in the SMP/E installation job FMESISMK. Assuming the Tivoli installation directory is /usr/lpp/Tivoli, the control statement highlighted below must appear in the FMESISMK job. This job sets up a link between Tivoli and the expected Java directory. The SMP/E installation process itself does not use Java.

```
//IKJEFT01 EXEC PGM=IKJEFT01
//SYSEXEC  DD DSN=hlq.H25I600.F1,DISP=SHR
//SYSTSPRT DD SYSOUT=H
//SYSTSIN  DD *
PROF MSGID
FMESMKDR /usr/lpp/Tivoli /usr/lpp/java/J1.1  1.1.6
```

Note: As mentioned above, Tivoli now supports Java 1.1.8. However, this command does not change. Even if the level of Java stored in /usr/lpp/java/J1.1 is version 1.1.8, the last parameter on the command line must still be 1.1.6.

After this job runs, you should be able to view the link by going to directory /usr/lpp/Tivoli/bin/os390/JRE and issuing the `ls -lt` command. The response should look like the following:

```
lrwxrwxrwx   1 setup    BIN         29 Sep 22 17:24 1.1.6 -> /usr/lpp/java/J1.1
```

If the link is missing or incorrect, you can manually create it by issuing the following commands:

```
cd /usr/lpp/Tivoli/bin/os390/JRE
ln -s /usr/lpp/java/J1.1  1.1.6
```

Replace `/usr/lpp/java/J1.1` with the actual directory where you install Java 1.1.6.

Note: As mentioned above, Tivoli now supports Java 1.1.8. However, this command does not change. Even if the level of Java stored in /usr/lpp/java/J1.1 is version 1.1.8, the last parameter on the command line must still be 1.1.6.

Installing the Tivoli Software

After installing Java, you must use SMP/E to install the following software. Note that this list assumes you are installing the Tivoli Framework and the five supported applications.

- For OS/390 UNIX System Services:
  - PTF UW61669 (for OS/390 V2R8; may not be required on other versions of OS/390)
  - PTF UW62102 (for OS/390 V2R8; may not be required on other versions of OS/390)

  - PTF UW67407 (for OS/390 V2R9 only)
  - PTF UW65634 (for OS/390 V2R8; UW65636 is available for OS/390 V2R9; and UW65635 is available for OS/390 V2R7, but it is only required if PTF UW65611 is also installed.)
- Tivoli Management Framework For OS/390 Server and Gateway (5697-D10, Version 3.6.1, FMID H25I600)
- Tivoli User Administration for OS/390 Server (5697-ADM, FMID H081610)
- Tivoli Security Administration for OS/390 Server (5697-SCM, FMID H085610)
- Tivoli Inventory for OS/390 (5697-F04, FMID H0AN600)
- Tivoli Software Distribution for OS/390 (5697-F03, FMID H0AO600)
- Tivoli Distributed Monitoring for OS/390 (5697-F05, FMID H0AR600)
- Tivoli Management Framework for OS/390 Framework Endpoint Version 3.6.1 (5697-D10, FMID H251625)

  Note that you must use SMP/E to install this code only if this version is not already integrated into the version of OS/390 that you are using.
- Tivoli User Administration for OS/390 Endpoint (5697-ADM, FMIDs H085605,J081625)
- Tivoli Security Management for OS/390 Endpoint (5697-SCM, FMIDs H085605, J085625)
- For Tivoli Management Framework (TMF):
  - PTF UW61819
  - PTF UW60543
  - PTF UW61473
  - PTF UW62015
  - PTF UW63798
- For Tivoli User Administration (TUA):
  - PTF UW60915
  - PTF UW61634
  - PTF UW63749 (server)
  - PTF UW63750 (endpoint)

  Notes:

  1. With UW63749 and UW63750 applied, we have observed that populating a user ID containing blank resume or blank revoke date fields results in a resume or revoke date of 01/01/71 in the Tivoli database. Because of the 01/01/71 date, the user ID may fail validation and fail the populate operation. In order to populate such a user ID, you must disable validation for the resume date field in the profile to which you want to populate the user ID. If the user ID then successfully populates to the profile, it will still contain the 01/01/71 date in the resume or revoke fields. Tivoli support assures us that this is not a problem. They are pursuing a change that will not display this date (see APAR OW43442).
  2. Furthermore, these two PTFs are only effective in an environment where you are using an OS/390 TMR server.

     In a non-OS/390 TMR server environment with PTF UW63750 applied to the OS/390 endpoint, you will experience the same date and validation characteristics described above and the problem described in APAR OW40443 will remain unresolved. Tivoli has not provided an equivalent patch to UW63749 for the non-OS/390 TMR servers.
- For Tivoli Security Management (TSM):
  - PTF UW61724 (for Tivoli Security Management (TSM))
  - PTF UW63711

- For Tivoli Distributed Monitoring (TDM):
  - PTF UW60983
  - PTF UW61853
  - PTF UW63823
  - PTF UW65862
  - PTF UW65863
- For Tivoli Software Distribution (TSD):
  - PTF UW63372
  - PTF UW67381
  - PTF UW71278
- For Tivoli Inventory (TIN):
  - PTF UW62995

Note that there are many patches available to resolve problems on non-OS/390 servers which have not been ported to OS/390 and, therefore, are unavailable to resolve problems on that platform. The following patch is of particular interest:

- 3.6.1-TMF-0051: for Tivoli Management Framework to resolve various problems, including removing a 512 MB limit from the Tivoli database, providing performance improvements, and providing additional support to resolve the autopack e-mail function not working. However, a fix to resolve the 512 MB limit on the OS/390 platform is currently in development (see APAR OW41600).

It's important to be aware of these limitations and plan your use of these products accordingly.

Configuring OMVS Environment Variables

Following the instructions in the program directory, we issued the following commands from the OMVS command line to set the environment variables _BPX_SHAREAS and _BPX_SPAWN_SCRIPT to NO:

```
_BPX_SHAREAS=NO
_BPX_SPAWN_SCRIPT=NO
```

For performance reasons, we do not recommend you set these variables to NO on a system-wide basis.

Configuring OMVS System Variables

Following the instructions in the program directory, you must set the OMVS parameters MAXCPUTIME and MAXASSIZE to at least the following minimum values:

```
MAXCPUTIME=1000000
MAXASSIZE=2147483647
```

While the program directory mentions other OMVS parameters, it gives no guidance on setting the correct values. Therefore, those parameters remain unchanged on our system.

Installing the TME 10 Desktop for Windows Version 3.6.1 on Windows NT 4.0

You use the Tivoli Desktop to communicate with the TMR Server and Gateway on OS/390. The Framework documentation indicates support for TME 10 Desktop for Windows version 3.6; however, this is incorrect. The PSP bucket indicates that TME 10 Desktop for Windows version 3.6.1 is required. You can find the TME 10 Desktop for Windows software on the Tivoli Framework 3.6.1 CD-ROM in directory \PC\DESKTOP\DISK1\SETUP.

Installing Java Virtual Machine and X Server on the NT 4.0 Workstation

While the Tivoli Desktop does not require a Java virtual machine (JVM) or an X server, the Software Installation Service (SIS) requires both. This can be confusing because you can start SIS either from the Tivoli Desktop or completely outside of the Tivoli Desktop. In both cases, however, SIS relies on the services of the JVM and the X server in order to run.

We used the Microsoft virtual machine (Microsoft VM) available from http://www.microsoft.com/java. For an X server, we used the Hummingbird Exceed version 6.1 product available from http://www.hummingbird.com.

Understanding HFS DASD Space Usage

The SMP/E installation process creates three HFSs, as follows:

- a software HFS that contains the SMP/E installed software
- an install repository (IR) HFS to contain imported software for use by the Software Installation Service (SIS)
- a database HFS to contain the object data for the TMR

We have the following observations to share:

- The software HFS (*tivoli_hlq*.SFMEHFST) uses approximately 700 cylinders (on a 3390-3 DASD) with the Tivoli Framework server and gateway and all five applications installed.
- The IR HFS (*tivoli_hlq*.SFMEHFSI) uses whatever space you choose to define. The size depends on the number of applications that are imported into the SIS IR. We populated the IR with the software from all of the product CD-ROMs and it amounted to 2000 cylinders. See “Creating Image Copies of Tivoli CD-ROMs for SIS” on page 178 for the products that we imported into this HFS.
- The database HFS (*tivoli_hlq*.SFMEHFSD) also uses whatever space you choose to define. Currently, we have no recommendations for the size of this HFS. Its size depends on the number of endpoints and many other factors. However, while you can allocate any amount of space for this HFS, Tivoli can only use 512 MB of the total allocated space. Once the space usage reaches 512 MB, you may receive unpredictable results when trying to create additional objects in the database. This limitation has been eliminated for the non-OS/390 TMR servers with patch 3.6.1-TMF-0051, but this patch has not been ported to OS/390. However, a fix to resolve the 512 MB limit is currently in development (see APAR OW41600).
- In our environment, we manually created a fourth HFS to contain image copies of all of our Tivoli software CD-ROMs. Those CD-ROMs used 2500 cylinders. See “Creating Image Copies of Tivoli CD-ROMs for SIS” on page 178 for the CD-ROMs that we copied into this HFS.

Our Experiences With Tivoli Management Framework for OS/390 Server and Gateway

After we completed the planning, setup, and installation of prerequisites, we ran the **wserver** command line interface (CLI) command to perform the final Tivoli Management Region installation and configuration.

Running the wserver Command

As stated in the program directory, you must run the **wserver** command under an OMVS user ID with root authority to perform the final installation and configuration of the TMR server. This command uses the file /etc/Tivoli/server.rsp as input, which we created according to the specifications in the program directory. We issued the command as follows:

```
wserver cli /etc/Tivoli/server.rsp
```

Note the following about this command:

- The program directory specifies the cli parameter as part of the command; however, while the function of the cli parameter is undocumented, it is required.
- You will receive errors when this command executes; however, these are normal, as documented in the program directory.
- You might not see the output messages from this command displayed until you press F10 (Refresh) to refresh the OMVS screen. This is a normal OMVS characteristic, not related to Tivoli.

When the **wserver** command completed, it displayed the following messages:

```
Z1EIP: Pushing/Unpacking Files:Install Repository
Z1EIP: Registering product
Z1EIP: Completed with success
Z1EIP: Finished installing 1 product(s)
Z1EIP: Finished at 19990923-1340
    Saving repository nodes.: Finished at: 19990923-1340
```

When the **wserver** command completes, the main Tivoli process, OSERV, is running on OMVS. Remember that, unlike on other TMR server platforms, the **wserver** command does not automatically configure the OSERV to start at IPL. You must specifically include the appropriate statements in the /etc/rc file on OMVS, as described in the program directory.

Note that the **wserver** command also creates an /etc/Tivoli directory and places the file setup_env.sh into this directory. This file needs to be *sourced*. That is, the file must be executed to set up the Tivoli environment variables and allow you to issue Tivoli CLI commands each time you sign on to OMVS. To do this, you can issue the following from the root directory:

```
. /etc/Tivoli/setup_env.sh
```

Warning About /etc/services Problem

In our OS/390 environment, we do not use the file /etc/services to define our TCP/IP services. We use the tcpip.etc.services data set instead. However, when you configure Tivoli with the **wserver** command, it expects to find the /etc/services file. When it did not find this file on our system, it created one containing only the following two entries:

```
objcall   94/tcp     # Tivoli daemon
objcall   94/udp     # Tivoli daemon
```

Because the /etc/services file now existed, at the next system IPL, TCP/IP found /etc/services first and did not look for tcpip.etc.services. As a result, many of our TCP/IP services did not start.

To correct this problem, we copied the two Tivoli entries from /etc/services into tcpip.etc.services and then deleted /etc/services. Tivoli is aware of this unique OS/390 TCP/IP behavior in their installation process. For now, note that Tivoli expects to find /etc/services and will create it if it does not exist. Also note that if you run the wserver command again, it will recreate the /etc/services file again. A fix for the OS/390 TMR server to prevent the creation of the /etc/services file if it does not already exist is currently in development (see APAR OW44079).

Resolving the ++HOLD Action for PTF UW61819

You must complete the ++HOLD action for PTF UW61819 before proceeding any further, or else you will corrupt the Tivoli database. To complete the hold action, we ran the following commands from OMVS. Note that the character surrounding the

data to the right of the equal sign on the second, third, and fourth lines is a backquote, not a single quote. On most keyboards, the backquote is the key to the left of the number 1 key (above the Tab key and below the Esc key). See the hold data in the PTF cover letter for complete instructions.

```
. /etc/Tivoli/setup_env.sh
NRO=`wlookup NameRegistry`
NRCO=`idlattr -t -g $NRO class_objid TMF_SysAdmin::InstanceManager | cut -d# -f1`
NRPO=`objcall $NRCO _get_prototyp`
objcall $NRO addattr lock_timeout
idlattr -t -s $NRO lock_timeout short 60
objcall $NRPO addattr lock_timeout
idlattr -t -s $NRPO lock_timeout short 60
```

A script file named UW61819.sh that contains the above commands is now available. See the PSP bucket for instructions on where to find the script file.

After you complete the hold action and OSERV is running, you can perform any of the following:

- Run Tivoli CLI commands.
- Connect to the OS/390 Tivoli Management Server through the Tivoli Desktop.
- Start the Software Installation Service (SIS) application with the **sisgui** command.

Running Tivoli CLI Commands from OMVS

You can run the standard Tivoli CLI commands from OMVS. The only requirement is that you must first *source*—that is, execute the following script to set up the Tivoli environment variables and paths:

```
. /etc/Tivoli/setup_env.sh
```

Connecting to the OS/390 Tivoli Management Server Through the Tivoli Desktop

In order to sign on to the OS/390 Tivoli Management Server and Gateway through the Tivoli Desktop, you must sign on using the RACF user ID of the default Tivoli administrator. The default user ID defined to the default administrator is `root`. To display the default administrator's characteristics, issue the following commands from OMVS:

```
. /etc/Tivoli/setup_env.sh
wgetadmin
```

On our system, the **wgetadmin** command displayed the following:

```
Administrator: Root_J80EIP-region
logins       : root@J80EIP.xxxxxxxx.IBM.COM
roles        : global super,senior,admin,user,install_client,
               install_product,security_group_any_admin user
               Root_J80EIP-region   admin, user, rconnect
notice groups: TME Administration, TME Authorization,
               TME Diagnostics, TME Scheduler
```

The second line of the display (logins) identifies the default RACF user ID as root. To add other user IDs to the list of logins, issue the following command:

```
wsetadmin -l userid_to_add "Root_J80EIP-region"
```

Replace `userid_to_add` with the actual user ID you want to add.

In our environment, the Tivoli Desktop runs on a Windows NT 4.0 workstation. We started the Tivoli Desktop on the workstation. Then, when prompted, we entered the OS/390 TMR Server IP address and the RACF user ID and password defined to the Tivoli Administrator's login.

Problem Starting Tivoli oserv and lcfd Daemons Under OS/390 V2R9

The Tivoli oserv and lcfd daemons failed to start under OS/390 V2R9 because they could not find various DLLs. PTF UW67407 for OS/390 UNIX System Services is available to resolve this problem. This problem only affects OS/390 V2R9.

Tivoli HFS Service Procedure

As we often describe in our test reports, OS/390 and most program products in our environment are installed on a build system and then copied to our production systems. Specifically, “Our HFS Service Environment” on page 126 describes how we apply service to our root HFS. However, note that the information in that section only applies to the non-Tivoli HFSs.

Tivoli Management Framework for OS/390 Server and Gateway presents a unique service challenge in our environment. As documented in the *Program Directory* and as we observed early in our testing, Tivoli Management Framework for OS/390 Server and Gateway and its associated products are intended to be installed and run on the same system. Therefore, our regular procedure for installing our environment on a build system and then copying it to a production system did not work.

The SMP/E process installs the base Tivoli elements, including several data sets and three HFSs (one for the Tivoli database, one for the SIS IR, and one for the Tivoli executable software). Once copied to the production system, the user runs the Tivoli **wserver** command which then significantly alters the software HFS (tivolihlq.SFMEHFST) containing the Tivoli server and gateway executable software. In addition, SIS makes other modifications to this HFS. Therefore, once we copy the Tivoli product from the build system and customize it on the production system, we can no longer simply apply PTFs on the build system and copy the software HFS to the production system.

To solve this dilemma, we developed the following procedure to service the Tivoli software HFS (tivolihlq.SFMEHFST):

Step 1. Shutdown the Tivoli services such as oserv and lcfd daemon.

Step 2. Dump the customized Tivoli software HFS on the production system.

Step 3. Send the dumped HFS to the build system.

Step 4. Restore the dumped HFS on the build system.

Step 5. Apply the required PTFs on the build system.

Step 6. Dump the customized Tivoli HFS on the build system (now with PTFs applied).

Step 7. Send the dumped HFS to the production system.

Step 8. Restore the dumped HFS with the PTFs applied and replace the old HFS on the production system.

Step 9. Copy all other non-HFS data sets affected by the PTFs from the build system to the production system.

Step 10. Resolve any PTF ++HOLD actions. In several instances, we have found that we had to run SIS to complete the installation of the PTF because the PTF actually put a patch in the SIS IR that needed to be installed with SIS.

Step 11. Restart the Tivoli services that we shut down in step 1.

Tivoli is aware of this service issue and is investigating alternatives to make the process less labor-intensive.

Software Installation Service

We have the following experiences to share about the Software Installation Service (SIS):

- As documented in the SIS Release Notes, SIS cannot install OS/2 endpoints. We also found that it cannot install Windows 95 endpoints. Although we have not tried it, SIS might not install Windows 98 endpoints either. You must locally install these types of endpoints at the endpoint workstations using the CD-ROM.
- SIS cannot be used to install TACF on UNIX endpoints. When you use SIS to install TACF Installation Utilities V3.6.1 SMPE on the TMR server, it creates a TACF-region policy region icon. In this policy region, there is an installation icon that you must then use to install TACF on the endpoints. See our experiences with Security Management later in this section.
- During an installation, SIS displays bar charts to show the installation progress. As documented in the SIS Release Notes, SIS does not display the correct completion percentages. We have found no other way to determine the progress of an installation.

Problem with SIS Hang: When we first attempted to use SIS to configure the SMP/E-installed Tivoli applications, SIS hung on a process called IRPOPULATE. PTFs UW65634 (for OS/390 V2R8), UW65636 (for V2R9), and UW65635 (for V2R7, but only if UW65611 has also been applied) are available to fix this problem. Other versions of OS/390 are not affected.

Creating Image Copies of Tivoli CD-ROMs for SIS: In order to import software from the Tivoli CD-ROMs into the SIS IR, you must make the software available to the OMVS file system. You can do this in a couple of different ways:

Using NFS: You can mount the CD-ROM on a workstation and use NFS to mount the workstation's CD-ROM drive on the OMVS system.

Note that you do not have to load all the CD-ROMs into the IR at the same time. If space is a concern, you can load just the one you need at the moment and then delete it when you're done. We chose to load everything all at once to save time.

Creating an Image Copy: You can create an image copy of the CD-ROM as a workstation file and then transfer the file to OMVS using the following steps:

1. Use the Tivoli CLI command **wcpcdrom** to copy the CD-ROM image to a workstation file. (Note that there are instances in the Tivoli documentation that refer to the **ucpcdrom** command. These should instead refer to the **wcpcdrom** command. See APAR II12376.)

   The **wcpcdrom** command does not copy the entire CD-ROM. It makes an image copy starting at the directory on the CD-ROM that you specify. It copies the Tivoli file packages and CFG files that it finds in that directory. If you want to copy another set of file packages and CFG files starting in another directory, you will need to run this command again and specify a different starting directory. For the complete command syntax, see *TME 10 Framework Reference Manual, V3.6*.
2. Create a tar file from the image copy on the workstation.

   You might wonder why we did not choose to tar the files straight from the CD-ROM instead of from an image copy. On a Windows NT workstation, we found that simply tarring the files right from the CD-ROM caused the CONTENTS.LST file and *.IND files to be untarred in lowercase in the OS/390 HFS. This caused SIS problems when trying to import the product into the IR. We recommend you use **wcpcdrom** to copy the files before tarring them and

sending them to OS/390. Make sure the CONTENTS.LST file and the *.IND files appear in uppercase after untarring them on OS/390.

3. Use ftp to transfer the tar file to OMVS.
4. On OMVS, untar the file into the OMVS file system.

We chose the latter approach and used ftp to transfer a tar file containing an image of the CD-ROM to a separate directory on OS/390 OMVS. We used the following commands to create an image copy of the Framework CD-ROM and tar the file:

```
mkdir /cdroms
cd /cdroms
bash
wcpcdrom -c d:/ g:/cdroms/framework36
tar -cv -f - framework36 > fw36.tar
```

We repeated the above steps to create image copies of the following CD-ROMs and ftp them to OS/390:

- Tivoli Framework Version 3.6
- Tivoli Framework Version 3.6.1 March 3, 1999
- Tivoli Software Installation Service Version 3.6
- Tivoli Software Installation Service Version 3.6.1
- Tivoli Software Installation Service Version 3.6.1 Revision B March 17, 1999

  Note that, initially, we were confused about where to find the endpoint software to install on the different endpoint workstations. We found that the endpoint software is in the /endpoints directory on the Software Installation Service Version 3.6.1 Revision B March 17, 1999 CD-ROM. Like most Tivoli installation directories, this directory contains the usual *.PKT, CONTENTS.LST, and *.IND files. However, it differs in that it does not contain a CFG subdirectory and does contain a group of *.SEED files. There is a seed file for each type of endpoint workstation. SIS needs the seed files in order to install the endpoint software on an endpoint workstation. The requirement for and location of the seed files is not mentioned in the Tivoli documentation (see APAR OW44215).

  Once we knew where to find the endpoint software and that the seed files are required, we attempted to use the **wcpcdrom** CLI command to create an image copy of the /endpoints directory. However, the image copy created did not contain the seed files. To get around this, we mounted the CD-ROM on a workstation and used **ftp** to transfer the seed files from the /endpoints directory on the CD-ROM to the directory on OMVS. Once the seed files were on OMVS, we were able to import the endpoint software, including the seed files, into the SIS IR (see APAR IY10085).
- Tivoli User Administration Version 3.6
- Tivoli User Administration Version 3.6.1 March 3, 1999
- TME 10 User Administration for OS/390 Version 3.6
- Tivoli Security Management Version 3.6
- Tivoli Security Management Version 3.6.1 March 3, 1999

  Note that the Tivoli Access Control Facility (TACF) v3.6.1 for AIX 4.3 and HPUX 11 software is located in the /tacfaihp directory on this CD-ROM. To make an image copy of this software, you must run **wcpcdrom** against the /tacfaihp directory.
- TME 10 Security Management for OS/390 Version 3.6
- Tivoli Security Management for OS/390 Version 3.6.1
- Tivoli Software Distribution Version 3.6
- Tivoli Software Distribution Version 3.6.1 March 3, 1999

- Tivoli Distributed Monitoring Version 3.6
- Tivoli Distributed Monitoring Version 3.6.1 March 3, 1999
- Tivoli Inventory Version 3.6
- Tivoli Inventory Version 3.6.1 March 3, 1999

Starting the Software Installation Service: The Software Installation Service runs on OS/390 under OMVS but displays its dialogs, panels, and other output on an X Window System such as an AIX or Windows NT 4.0 workstation running an X server. The workstation must also have a Java virtual machine (JVM) running. As we mentioned earlier, we used a Windows NT 4.0 workstation with the Microsoft VM and Hummingbird Exceed.

There are two ways to start the SIS:

1. From the Tivoli desktop, select **Software Installation Service** from the **Desktop** menu. For this to work, you must define the Tivoli desktop workstation's IP address in /etc/hosts on the OS/390 system that runs the Tivoli Management Framework for OS/390 Server and Gateway.
2. Issue the SISGUI command from an OMVS session on the OS/390 system that runs the Tivoli Management Framework for OS/390 Server and Gateway. For this to work, you must set and export the DISPLAY environment variable in OMVS to the IP address of the workstation where the SIS dialogs are to display.

To start SISGUI, make sure the X server is active on the workstation and that its security is set up to allow an external system to start an X session on this machine. Then, enter the following from OMVS:

```
export DISPLAY=nt_workstation_ipaddress:0.0
. /etc/Tivoli/setup_env.sh
cd /usr/lpp/Tivoli/bin/generic_unix/SIS
sisgui
```

For `nt_workstation_ipaddress`, specify the IP address of the workstation to display the SISGUI dialogs.

After a few minutes, the first SIS dialog will appear on the workstation. The SIS startup and initialization can take a long time, up to ten minutes.

To enable it to install and configure applications, SIS uses a database called the Install Repository, or IR. The IR must contain all the software that SIS is to install or configure. The software gets into the IR in two ways:

1. The SMP/E process automatically puts the Tivoli applications that it installs into the IR.
2. You must import all other Tivoli applications into the IR from the CD-ROMs.

In general, software that executes on OS/390 is placed into the IR by the SMP/E process. Software that executes on the distributed endpoints and managed nodes (non-OS/390 systems) must be imported into the IR from the CD-ROMs.

Note that once you start SIS, either from the Tivoli desktop or with SISGUI, it may take several minutes before anything appears on the workstation where SIS is to display its dialogs. There have been occasions when SIS was unable to start and we waited for dialogs that never appeared. To verify that SIS is starting, issue the following command from OMVS:

```
ps -ef
```

The following is an example of the processes that will be started when SIS is first initializing. However, they will end within the first few minutes.

```
17:43:51 ? 0:00 /bin/sh /usr/lpp/Tivoli2/bin/os390/../generic_unix/SIS/launch_sis
17:43:57 ? 0:00 idlattr -tg 1294805934.1.342#TMF_SysAdmin::InstanceManager#extension Object
```

The following is an example of the process that appears when SIS is finally running:

```
17:44:15 ? 0:16 /usr/lpp/Tivoli2/bin/os390/JRE/1.1.6/bin/mvs/native_threads/java -classpath .:
```

If you encounter SIS problems, you might find additional information in the file /usr/lpp/tivoli/IR/sis-*xxxxxx*.out, where *xxxxxx* is the TMR name.

Importing Software Into SIS: Once we had the CD-ROM images on OMVS, we used SISGUI to import the images into the IR. The import process worked as described in the Tivoli documentation. However, we did find two instances where SIS prompted us with the following message:

```
Invalid Product Revision dialog. Enter a valid product revision.
```

You can then enter the correct product revision number, assuming you know what it is. We entered the following revision numbers for the indicated products:

- `3.6` — for TME 10 Inventory PC Scanning Program, 3.6
- `3.6.1` — for TME 10 Inventory PC Scanning Program, 3.6.1
- `3.6` — for Tivoli User Administration for OS/390 3.6 - Tivoli Gateway install pack

To find the product revision number of any other product that causes SIS to prompt for a revision number, you must have access to a workstation version of a TMR server where the product is already installed. You can then issue the command:

```
wlookup -ar ProductInformation
```

This displays a list of installed products and their object ID (OID) numbers. To find the revision number, issue the command:

```
idlcall oid _get_revision
```

Replace `oid` with the OID number displayed by the previous command.

If you do not have access to a workstation TMR, then you must call Tivoli support for assistance.

User Administration

Information concerning our experiences with User Administration can be found in our original work with non-OS/390 TMR servers and the OS/390 endpoint in “Experiences with TME 10 User Administration for OS/390” on page 195. Listed below are some additional User Administration observations from our testing of the OS/390 Server and Gateway and Endpoint:

- There is a characteristic exhibited by the User Administration CLI **wpopusrs** that we describe in “Populating the Tivoli Database with wpopusers” on page 197 and “Our Operational Experiences” on page 199 that will populate old revoke and resume dates from the RACF database to the Tivoli User Administration database. When a RACF user ID is given revoke or resume dates, those dates remain in the RACF database even after the dates have passed and RACF no longer displays them. However, during the populate process with **wpopusrs**, those dates are found and are populated to Tivoli. The first time a user ID with one of these old revoke or resume dates is distributed back to RACF, the distribute fails because RACF will not accept revoke or resume dates that are in the past. Only user IDs that have ever had a revoke or resume date are affected. PTFs UW63749 and UW63750 attempt to correct this problem. However, while these PTFs appear to have corrected the problem on the OS/390-based TMR,

they have also introduced another characteristic that puts the date 01/01/71 in the revoke or resume fields for user IDs that have no revoke or resume date. As mentioned in “Installing the Tivoli Software” on page 171, Tivoli assures us that this will cause no problem. Tivoli is working on a solution to hide the 01/01/71 date from view (see APAR OW43442).

- Note also that the fix for the revoke/resume date problem is only available for environments where the TMR server is on OS/390.
- We were able to do some initial testing with the ADMSEC category of User Administration which allows you to connect a user ID to groups directly from the User Administration dialogs, even though the groups are in Security Management. While the function appears to work, we found two characteristics that may cause concern:
  1. If ADMSEC is used to connect a user ID to a group, then *both* the User Administration profile and the Security Management profile must be distributed to RACF. Distributing just the User Administration profile did not cause the Security Management profile to be automatically distributed. Tivoli support is aware of this problem and is working on a solution.
  2. If ADMSEC is used under User Administration to connect a user ID to a group, and then the user ID is disconnected from the group using Security Management, then Security Management does not remove the reference to the group from the user ID under ADMSEC. You must manually delete the group from the user record with ADMSEC (see APAR IY05316).
- If you are running a TMR server on Windows NT 4.0 with Service Pack 4 or 5, you may experience problems with **wcrtusr**. Specifically, we have seen the following error messages:
  - `general oserv errors`
  - `Error Writing log file`
  - `uto_skel1.exe - Application Error. The instruction at 0x77f648a7 reference memory at 0x4e000000. The memory could not be written.`

  We have not experienced these problems on Windows NT 4.0 with Service Pack 3. To resolve this problem with Service Pack 4, you must install patch 3.6.1-ADM-0020. The patch is available on the Web at http://www.support.tivoli.com.

Security Management

Information about Security Management can be found in our original work with non-OS/390 TMR servers and the OS/390 endpoint in “Experiences with TME 10 Security Management for OS/390” on page 200. Listed below are some additional Security Management observations from our testing of the OS/390 Server and Gateway and Endpoint:

- As stated earlier under SIS, you use TACF Installation Utilities to install TACF endpoints, as defined in *Security Management User's Guide Version 3.6*. However, the documentation does not explain that after clicking the TACF-region policy region installation icon, you likely will see an error dialog that says:

  ```
  Something is wrong with the current installation media settings.
  The media host must be connected, and the media directory needs
  to have a file (CONTENTS.LST) which contains a list of patches
  present. Please select a valid media directory. Also, note that
  you can use the "Select Media..." button, on the "Product
  Install" dialog, to change/inspect the current media settings.
  ```

  While this appears to be a program error, it is not. It is a standard message that displays when the Tivoli installation utilities are not pointing to the correct directories. You only need to click **OK** and then enter the correct path to the software to be installed.

Once you click **OK**, a file browse dialog opens. In this dialog, do one of the following:

– If you are installing a product, type `/usr/lpp/Tivoli/IR/Products` in the **Path Name** field to display and install the following TACF products:

```
TACF-3.6-Tivoli_Access_Control_Facility_TACF_3.6
TACF-3.6-Tivoli_Access_Control_Facility_TACF_v3.6.1_for_AIX_4.3_and_HPUX11
```

– If you are installing upgrades or patches, type `/usr/lpp/Tivoli/IR/Patches` in the **Path Name** field to display and install the following TACF patches:

```
TACF-3.6.1-Tivoli_Access_Control_Facility_TACF_Upgrade_to_Version_3.6.1
```

Then, click **Set Media and Close**. This resolves the problem and you can continue with the installation.

- TACF is supported on HPUX 11 and AIX 4.3.1 as defined on page 17 of *TME 10 Security Management Release Notes Version 3.6.1*. Be aware that TACF is *not* supported on AIX 4.3.2 and AIX 4.3.3.
- When populating the data set class to Tivoli, if the RACF database includes profiles for fully-qualified generic data sets (that is, a data set profile with the generic property but with no generic character, such as an asterisk (*), in the profile name), it is possible that ***no*** data set profiles after the fully-qualified generic data set profile will be populated to the Tivoli Security Management database. Currently, there is no circumvention and Tivoli support is investigating the problem (see APAR OW44421).
- When populating any non-data-set class to Tivoli (such as device, tapevol, terminal, or others), if that class only contains generic profiles (and no discrete profiles), then no profiles of that class will be populated to Tivoli (see APAR OW45677). The only circumvention we have found is to add at least one discrete profile of that class to the RACF database. However, due to the problem described in APAR OW45678, this might not always work. We recommend that, after populating general resource profiles from RACF to Tivoli, you verify that all of the expected RACF profiles have been populated into Tivoli records.
- When populating any non-data-set class to Tivoli (such as device, tapevol, terminal, or others), there is a condition that can occur whereby some generic RACF profiles are not populated to Tivoli (see APAR OW45678). There is no way to determine if that condition exists in the RACF database. However, there are two symptoms that we have observed with this problem:

  1. After populating the RACF class, all of the discrete profiles are populated to Tivoli but some of the generic profiles may be missing from the Tivoli database.
  2. We received some ″resource already exists″ messages during the populate operation, even though that RACF class profile had never existed in the Tivoli database before.

  Again, we recommend that after populating general resource profiles from RACF to Tivoli, verify that all of the expected RACF profiles have been populated into Tivoli records.
- When installing Tivoli Security Management for OS/390 Server, the instructions tell you to create the file SecRacfLog with permissions set to 600. We have found that if SecRacfLog becomes full (that is, it reaches 2 MB in size), it may not wrap around and may cause the Security Management operation that is currently in progress to fail with the following messages (see APAR II12045):

  – If a distribute was in progress, the following pop-up message may display on the distribute dialog:

```
The following errors occurred during distribution:
-> Distribute failed for subscriber 'xxxxxx':
-> ipc_timed_send failed: peer=xx.xx.xx.xx+9495, Error=IPC
   shutdown (67)
```

– In the lcfd.log on the OS/390 endpoint:

```
Exception rolling-over logfile: ./SecRacfLog:
Open failed: EDC5111I Permission denied.
```

The following solution resolves the problem and is fully described in the PSP bucket:

1. Create a RACF group, such as TIVADMIN, with any non-zero OMVS gid. We created it with the following:

   ```
   ag TIVADMIN OMVS(GID(123))
   ```

2. Connect the RACF user ID NOBODY and any other user IDs that will appear in the APPLDATA field of a TMEADMIN class profile to group TIVADMIN:

   ```
   connect NOBODY group(TIVADMIN)
   connect OTHERID group(TIVADMIN)
   ⋮
   ```

3. Create the SecRacfLog file:

   ```
   touch SecRacfLog
   ```

4. Change the ownership of the SecRacfLog file to the user ID NOBODY:

   ```
   chown SecRacfLog NOBODY
   ```

5. Change the permissions on the SecRacfLog file to 660:

   ```
   chmod 660 SecRacfLog
   ```

6. Change the group ownership of SecRacfLog to TIVADMIN:

   ```
   chgrp TIVADMIN SecRacfLog
   ```

In order for this to work, the RACF list-of-groups checking must be active or the RACF group that was defined (TIVADMIN) must be the default group of the connected TMEADMIN class APPLDATA user IDs.

Software Distribution

We have the following experiences to share about Software Distribution:

- If you use the Restart Windows or the Reboot Machine option with the BARC program when distributing a file package profile to a Windows NT endpoint or managed node, you must be running Windows NT 4.0 with Service Pack 4 or higher. Be aware of the potential problems that Windows NT 4.0 Service Pack 4 and 5 can cause for User Administration (see the User Administration topic earlier in this section.)

  If your Windows NT 4.0 system is a managed node, you must install patch 3.6.1_TMF_0051 on that system to enable the reboot function that is available through Software Distribution.

- If you use the sample script checkfpout.sh to automatically convert the contents of file package profiles between ASCII and EBCDIC formats at distribution time, as documented in *Tivoli Software Distribution for OS/390 Release Notes* (pages 17-21), the conversion does not work if your file package is distributing a directory rather than individual files. APAR OW43760 is open to address this problem. We plan to test and report on this fix in a future test report.

- If you create file package blocks and define the script checkfpout.sh as the source before program on a file package, the **wcrtfpblock** command fails with a general failure error (see APAR OW45155).

  A circumvention for this and the previous conversion problem is to edit the checkfpout.sh sample script as follows:

  1. Change line 23 from this:

```
if [ $1 = before ] && [ $2 = install ] && [ x$FPNAME != x ]
```

to this:

```
if [ $1 = before ] && ( [ $2 = install ] || [ $2 = fpblock ] ) && [ x$FPNAME != x ]
```

2. Change line 29 from this:

```
elif [ $1 = after ] && [ $2 = install ] && [ x$FPNAME != x ]
```

to this:

```
elif [ $1 = after ] && ( [ $2 = install ] || [ $2 = fpblock ] ) && [ x$FPNAME != x ]
```

- If you define the checkfpout.sh script as the source before program or the source after program (or both) on a file package that is a nested file package of another file package, the nested file package does not distribute with the primary file package. (In a previous test report, we directed you to see APAR OW44161. This APAR has been changed. Please see APAR OW44358 instead.)
- When distributing a file package that includes a nested file package, the contents of the log file for the nested file package are not included in the log file for the primary file package (see APAR OW44566).
- The function UserLink for TME 10 Software Distribution does not work. APAR OW43763 is open to address this problem. We plan to test and report on this fix in a future test report.

  A circumvention for this problem is to edit the /usr/lpp/Tivoli/bin/os390/TAS/HTTPd/cgi-bin/Courier/get_version.cgi file. In the file, line 624 is incorrectly split between lines 624 and 625. Line 624 should look like the following:

  ```
  $t_zero-$t_one));
  ```

  Line 625 should be blank.

Distributed Monitoring

We have the following experiences to share about Distributed Monitoring:

- When installing the Distributed Monitoring 3.6.1 Maintenance Release on a managed node, *Tivoli Distributed Monitoring for OS/390 Release Notes for 3.6.1* is misleading for defect 67349. The release notes indicate to run the following script (this appears on page 14 in our copy):

  ```
  #!/bin/sh
  OSERV='objcall 0.0.0 getattr oserv'
  HOST='hostname'
  SECO='wlookup -r Classes SentryEngine'
  SEOID='objcall $SECO _lookup_object $HOST
  objcall $OSERV boot_method add SentryEngine $SEOID
  run_engine
  exit 0
  ```

  The script should actually look like the following (`run_engine` should really be part of the preceding line):

  ```
  #!/bin/sh
  OSERV='objcall 0.0.0 getattr oserv'
  HOST='hostname'
  SECO='wlookup -r Classes SentryEngine'
  SEOID='objcall $SECO _lookup_object $HOST
  objcall $OSERV boot_method add SentryEngine $SEOID run_engine
  exit 0
  ```

- When using the Run Program option and selecting On Monitored Host, you must use a forward slash (/) in the path name, even for Windows NT systems (see APAR IY09198).

- Distributed Monitoring Webinspector does not work. Tivoli support is investigating the problem, but an APAR has not yet been opened. If you attempt to use Webinspector, you might see the following error message displayed at the bottom of the TME 10 Web Access page:

  ```
  Applet SentryData class SentryData could not be loaded
  ```

  You might also see messages similar to the following in the $DBDIR/.HTTPd/httpserv.log file:

  ```
  <APPLET CODE="SentryData.class" CODEBASE="http://xxxxx:5000/
  classes" WIDTH=0 HEIGHT=0> <PARAM NAME=Authorization
  VALUE="Basic cm9vdDpzccHIxbmc=">
  <P><STRONG>WARNING: JAVA DISABLED OR
  UNSUPPORTED</STRONG>
  </APPLET><BR></BODY></HTML>
  ```

- Distributed Monitoring PTF UW63823 now has a script available to execute all the commands required by the ++HOLD action. The name of the script is UW63823.sh and the PSP bucket provides instructions on where to find it.
- When creating a monitor, if you specify Set Distribution Actions to distribute a program or script along with the monitor, the distribute fails for this monitor and displays the following error message:

  ```
  A failure was detected by the oserv daemon: general failure.
  "general failure" frequently means that a method
  finished abnormally due to a serious error condition.
  ```

  Tivoli support is investigating this problem.

Inventory

Before we describe our experiences with using Tivoli Inventory, we first want to share some of our experiences with installing the application.

Our attempts to connect Tivoli Inventory to DB2 caused some difficulty during our initial installation. While we used the Tivoli documentation *TME 10 Tivoli Inventory Supplement for OS/390 Version 3.6.1*, the RDBMS Interface Module (RIM) configuration information was not always clear. After configuring the RIM, we ran the following command from OMVS:

```
wrimtest -l inventory
```

This command checks the connection from Tivoli to DB2. When we ran it, we got the following error messages:

```
Opening Regular Session...Mon Dec 20 15:14:07 1999 (18):
 : 'PA_RESULT' failed with code '-1':''
```

Unfortunately, this is a generic error message that can be caused by many different errors and there may not be a log that identifies what caused the error. If you get the above error message, the following is a checklist of items to verify:

1. DB2 must be up and DDF must be installed and functional.
2. RIM Object parameter values must be correct, as described in Table 16:

Table 16. RDBMS Interface Module (RIM) configuration values for Tivoli Inventory

RIM Object Parameter	Parameter Value	Comments
RDBMS Vendor	DB2	This field is set during installation by selecting DB2 from the pulldown list.
RIM HOST	ALI_host	This is the default and is the value that we used.

Table 16. RDBMS Interface Module (RIM) configuration values for Tivoli Inventory (continued)

RIM Object Parameter	Parameter Value	Comments
DATABASE ID	INV36MVS	This is the name of the database as defined in the SQL scripts AUTH1, SCHEMA, and AUTH2 provided by Tivoli. INV36MVS is the default and is the value that we used. The information in this field is not case sensitive.
DATABASE HOME	Any existing OMVS directory	This is a required RIM field; however, it is not used for DB2. Therefore, you can specify any existing directory path. The information in this field is case sensitive.
DATABASE INSTANCE	Any existing OMVS directory	This is a required RIM field; however, it is not used for DB2. Therefore, you can specify any existing directory path. The information in this field is case sensitive.

SERVER ID	DDF location parameter	DB2 displays the DDF location at startup. When DB2 completes its startup, you should see messages similar to the following in the OS/390 system console log:
		`DSNL004I @DB1Z DDF START COMPLETE`
		`LOCATION: USIBMDB2`
		`LU : USIBMT6.DB2DB1Z`
		`GENERICLU -NONE`
		`DOMAIN -NONE`
		`TCPPORT 446`
		`RESPORT 6033`
		`DSN9022I @DB1Z DSNYASCP 'START DB2' NORMAL COMPLETION`
		
		In the above example, the location USIBMDB2 is the value to use in the SERVER ID field. The information in this field is not case sensitive.

USER NAME	NOBODY	NOBODY is the default value as delivered from Tivoli. Make sure it is entered into this field in uppercase. If you choose to use a different user ID, make sure that you do the following: • Create the user ID in RACF. If you use RACF authentication with DB2, make sure that this user ID is connected to the appropriate RACF secondary authentication group for DB2. • Create the DB2 database with this user ID as the owner. This user ID must be the owner of the database. Setting the authority and creating the database are done using SQL scripts AUTH1, SCHEMA, and AUTH2. • Enter the user ID into the RIM field in uppercase. A lowercase user ID will not work.

3. DB2 module DSNHDECP must be available to Tivoli either through a STEPLIB in OS/390 UNIX System Services or through an entry in LNKLST. In our case, this module was found in the *db2hlq*.SDSNEXIT data set. We accessed it through STEPLIB in OMVS by issuing the following command under OMVS:

   ```
   export STEPLIB=db2hlq.SDSNEXIT
   ```

4. The DDF Location parameter must be bound to the DB2 plan. In our case, it needed to be bound to DSNOCLI in SYSIBM.SYSPLAN. Note that you may need to rerun the bind after applying service to DB2.

Tivoli Management Framework for OS/390 Server and Gateway

We have the following additional experiences to share about Tivoli Inventory:

- We found that some software scans did not populate the DB2 database with all of the software found on the endpoint. This was because certain fields in DB2 were not long enough to hold the full name of some of the software. As a result, long names were truncated, which made some names appear to be the same as some other software package. The result was that the MIF file was created with all of the correct software entries for the scanned endpoint, but not all were stored in DB2 because they appeared, after truncation, as duplicate names (see APAR IY09138).
- We created a User Data Template to collect user information and store it in a useradd.mif file. The data collection process worked correctly, but the useradd.mif file was created and stored on the AIX endpoint in EBCDIC format, instead of ASCII. Therefore, the READ function in Inventory was unable to read the file and failed with the following message in the Inventory Notice Group (see APAR OW44310):

```
** Exception caught in run_impl: MIF parse error:
: useradd.mif: line 1: Syntax error: "aa" unexpected
```

Tivoli Management Framework for OS/390 Framework Endpoint

This section contains the original information from our previous test reports describing the environment where the TMR server is on a Windows NT workstation and the OS/390 system is just an endpoint in the TMR. See “Tivoli Management Framework for OS/390 Server and Gateway” on page 169 for our discussion of the environment where the TMR server is on OS/390.

Up until now, managing OS/390 with Tivoli has been somewhat limited by the number of applications available and the extent to which a Tivoli server could manage OS/390 systems. This has evolved to the product we tested that is based on TME 10 Management Framework Version 3.6, and that allows an OS/390 system to be managed (still with a limited number of applications) from a server just like any other currently supported platform.

Note that for the TMR server, Tivoli Management Framework 3.6.1, Tivoli Security Management Server and Gateway 3.6.1, Tivoli Security Management for OS/390 Server and Gateway 3.6.1, as well as the 3.6.1 version of the OS/390 Endpoint, User Administration, and Security Management are now available. In the sections that follow, we have included some updates based on our experiences with version 3.6.1. We are now exclusively testing in the 3.6.1 environment.

Our Hardware Configuration

Figure 58 on page 189 illustrates our current Tivoli Management Framework for OS/390 Framework Endpoint configuration (we’ve been making changes to this configuration as our testing progresses):

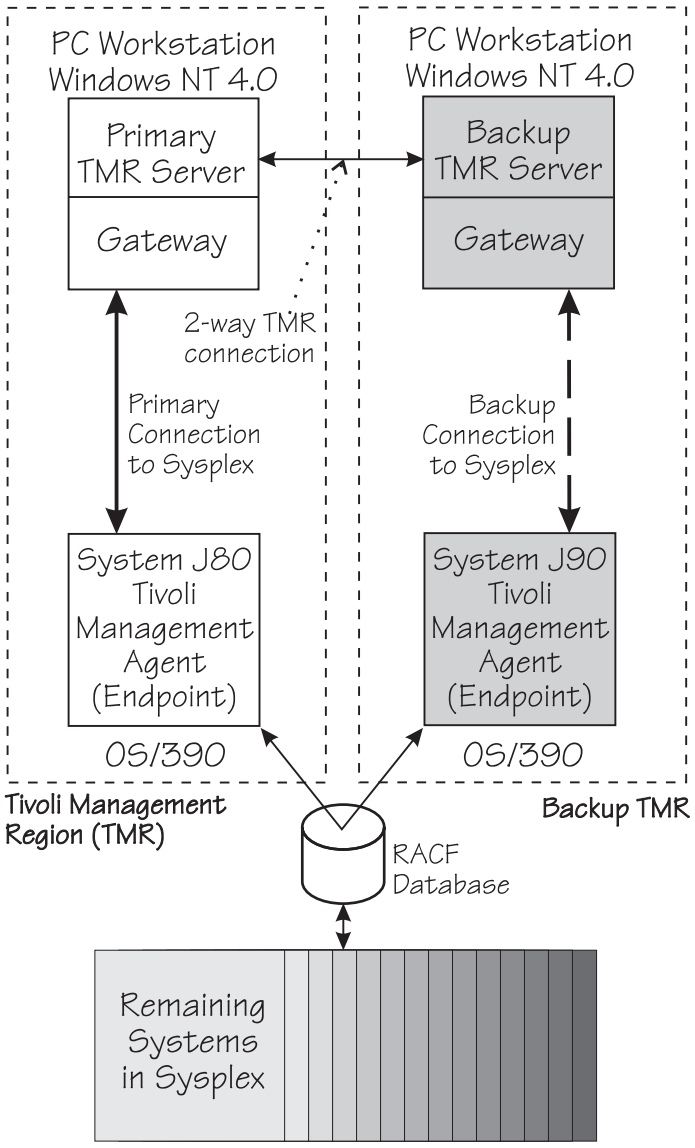


Figure 58. Our Tivoli Management Framework for OS/390 Framework Endpoint Configuration

Note the following about the above figure:

- We can sit at either of the workstations shown in the figure to manage our OS/390 systems. In fact, with the Tivoli Desktop for Windows, we could access the TMR servers from any Windows workstation that has IP connectivity to any managed node in the TMR. However, we use the primary TMR server as our primary interface.
- We have our gateways on the same workstations as our TMR servers. However, you could have your gateway on a different workstation than your TMR server, and you can have multiple gateways being managed by a single TMR server.
- We have OS/390 systems as our endpoints. However, Windows NT and various types of UNIX workstations (HP-UX, Solaris, SunOS) can also be endpoints.
- We have two separate Tivoli Management Regions (TMRs) in our configuration. We have a TMR consisting of just a TMR server and gateway and system J80 (as the endpoint providing the primary connection to our sysplex). Our second

TMR acts as a backup to our primary configuration, providing an alternate path to our sysplex through system J90. Thus our complete configuration has total redundancy for maximum availability.

- Because the OS/390 Security Server (RACF) database is shared amongst all systems in our sysplex, we can manage security for all systems through a single system (either system J80 or J90). If each system in our sysplex had its own RACF database, then each system would have to be an endpoint.
- We have a 2-way TMR connection between our primary and backup TMR servers. That means we could be sitting at the primary TMR server workstation and work on the backup TMR's desktop, or be sitting at the backup TMR server workstation and work on the primary TMR's desktop.

An important point to note is that our testing will be cross-platform. With Tivoli support, you can ***manage multiple platforms*** (AIX, HP-UX, SunOS, Solaris, Windows 3.x, Windows 95, Windows 98, Windows NT, Netware, OS/2, OS/400, and now also OS/390) ***from a single desktop*** (AIX, HP-UX, SunOS, Solaris, and Windows NT). In our environment, we'll manage OS/390, AIX, and Windows NT systems from a desktop on a Windows NT server.

Our Installation Experiences

Our installation of TME 10 Framework for OS/390 Framework Endpoint went fairly smoothly, using the *Program Directory for TME 10 Framework for OS/390 Framework Endpoint*. We do, however, have a few experiences to point out.

Figuring Out Which Code To Install On Which Machine

When you install the TME 10 Framework, the code for the OS/390 endpoint, and the Tivoli framework-based applications, there are many pieces to install, particularly if you are installing multiple applications on multiple workstations. You need to install certain code on OS/390, certain code on the TMR server workstation, and certain code on the gateway workstation. The following information, which we hope you will find helpful, describes what we installed and where we installed it:

- Code we installed on OS/390 (this list might vary depending on the level of OS/390 used):
  - TME 10 Framework for OS/390 Framework Endpoint(5697-D10)
  - TME 10 User Administration for OS/390 (5697-ADM)
  - PTF UW51533 (for TME 10 User Administration)
  - PTF UW54513 (for TME 10 User Administration)
  - PTF UW55092 (for TME 10 User Administration)
  - TME 10 Security Management for OS/390 (5697-SCM)
  - PTF UW55460 (for TME 10 Security Management)
  - PTF UW55461 (for TME 10 Security Management)
- Code we installed on the TMR server:
  - TME 10 Framework Version 3.6
  - TME 10 Framework patch 3.6-TMF-0005
  - TME 10 User Administration Version 3.6
  - TME 10 User Administration for OS/390 Version 3.6-Tivoli Server install pack
  - TME 10 Security Management Version 3.6
  - TME 10 Security Management patch 3.6-SEC-0003
  - TME 10 Security Management for OS/390 Version 3.6-Tivoli Server install pack
  - TME 10 Security Management patch 3.6-SEC-0005
- Code we installed on the gateway:
  - TME 10 User Administration Gateway Package Version 3.6

- TME 10 User Administration for OS/390 Version 3.6-Tivoli Gateway install pack
- TME 10 Security Management for Gateways Version 3.6
- TME 10 Security Management for OS/390 Version 3.6-Tivoli Gateway install pack

Choosing Ethernet or Token Ring

When setting up the Tivoli Management Region (TMR) and installing TME 10 Framework, if you have both an Ethernet and a token-ring connection to the network on the server or gateway systems, Tivoli will most likely choose to use the Ethernet connection. We are working with Tivoli support to understand this preference. We found that to get Tivoli to choose the token-ring adapter, you may have to disable the Ethernet adapter.

Setting Up and Running the lcfd Daemon (or OS/390 Endpoint)

In operation, the OS/390 endpoint is a single process running under OMVS, called the *lcfd daemon*. It is started with the **lcfd.sh** command. In the remainder of this chapter, we will refer to the endpoint as the lcfd daemon.

Installing the lcfd Daemon: In this section, we discuss our experiences with installing and running the lcfd daemon, including how to recognize some common problems you might encounter and how to avoid them.

Where to Find the Startup Documentation: The TME 10 Framework for OS/390 Framework Endpoint product comes with only one publication: *Program Directory for TME 10 Framework for OS/390 Framework Endpoint*. The program directory provides most installation and configuration documentation; however, it does not explain the start procedure or the available start options. To understand how to start the lcfd daemon, see *TME 10 Framework Reference Manual, V3.6* that comes with TME 10 Framework for the workstation-based TMR server product. The reference manual explains the lcfd.sh startup command and all of the available options.

Creating the Run-Time Directory /etc/Tivoli/lcf/dat: The program directory specifies the directory structure for the Run-time Data Location as /etc/Tivoli/lcf/dat, which is the default specified in the lcfd.sh shell script. This directory must be writable because the program creates files and directories while it executes.

Note that you can select any directory you want for the run-time files. However, if you choose a directory other than /etc/Tivoli/lcf/dat, you should copy lcfd.sh to that directory and edit it to change LCF_DATDIR to the desired directory structure.

From our experience, it may be helpful to know the following:

- We used the /etc/Tivoli/lcf/dat directory in our installation.
- The /etc/Tivoli/lcf/dat directory must be manually created (it is not automatically created during the SMP/E installation process). Use the mkdir and chmod commands to create the /etc/Tivoli/lcf/dat directory and assign it the following permissions:

Table 17. Access Permissions for the /etc/Tivoli/lcf/dat Directory Structure

Directory Level	Access Permissions
/etc	755
Tivoli	777
lcf	777
dat	777

- The /etc/Tivoli directory may already exist if you installed and ran the Event Automation Service in TME 10 Global Enterprise Manager (GEM) V1R1. This service provided the capability to forward NetView alerts and messages to the Tivoli Enterprise Console (TEC). The started task IHSAEVNT creates directory /etc/Tivoli the first time it is run. If this directory has already been created, verify that the permissions are set correctly before running the lcfd daemon.

Starting the lcfd Daemon: In this section, we explain the differences between first-time startup and subsequent startups of the lcfd daemon, as well as describe some start options you may find useful.

Starting the lcfd Daemon For the First Time: The first time the lcfd daemon is started, you must use the `install` option. This option causes several directories, files, and logs to be created. It also makes an initial connection to the TMR gateway and registers the OS/390 endpoint. We typically use the following command:

```
lcfd.sh install -d 2 -g xx.xx.xx.xx
```

The `-d 2` option sets logging to level 2 and `-g xx.xx.xx.xx` attempts to connect to the gateway IP address specified by `xx.xx.xx.xx`. The `-g` option is equivalent to the `-Dlcs.login_interfaces=` option. Therefore, the above command is equivalent to:

```
lcfd.sh install -d 2 -Dlcs.login_interfaces=xx.xx.xx.xx
```

Starting the lcfd Daemon After the First Time:

- **Using the Command Line:** After the lcfd daemon is successfully started the first time with the `install` option, all subsequent starts can be performed by issuing the following command from the OMVS command line:

  ```
  lcfd.sh start
  ```

  This type of start uses the configuration information stored in file /etc/Tivoli/lcf/dat/last.cfg by previous runs of the lcfd daemon.
- **Using the System Console:** Three sample procedures are provided in hlq.TIVOLI.SMFESMP1 to start, stop, and install the lcfd daemon from the system console: FMEESTRT, FMEESTOP, and FMEEINST, respectively. FMEESTRT and FMEESTOP will run as delivered in the sample library with no modifications. However, FMEEINST must be modified to include the gateway IP address. Here is an example of the FMEEINST procedure statements showing the modifications:

  ```
  //FMEEINST PROC
  //EXECLCFD EXEC PGM=BPXBATCH,
  //*       PARM='SH nohup /etc/Tivoli/lcf/dat/lcfd.sh install',
  //*       TIME=NOLIMIT
  //*
  //*               .------- change the above PARM= to -------.
  //*               |                    |                    |
  //*               V                    V                    V
  //        PARM='SH nohup /etc/Tivoli/lcf/dat/lcfd.sh install -d 2 -g xx
  //             .xx.xx.xx',
  //        TIME=NOLIMIT
  ```

  It is important that the PARM statement end in column 71 and continue on the next line starting in column 16.

  Note that if the _BPX_SHAREAS and _BPX_SPAWN_SCRIPT environment variables are set to anything other than NO, you can start the lcfd daemon with FMEESTRT; however, the Security Management application may not function properly. APAR OW44995 is open to investigate this problem.

Specifying Other lcfd Daemon Start Options: As we mention above, see *TME 10 Framework Reference Manual, V3.6* to understand how to use lcfd.sh to start the lcfd daemon and to understand all of its options. Below we highlight some of the options you might find helpful.

- **Timeout Values for the lcfd Daemon:** At startup, if the gateway is not available, the lcfd daemon repeatedly attempts to connect to it. With the default startup values, the lcfd daemon tries to make the connection up to six times at five-minute intervals before it attempts to use another gateway address. This means that if the gateway is down, it takes 30 minutes before attempting another gateway. You may want to change these default timeout values. Here is an example of how to do this (enter this as a single command all on one line):

  ```
  lcfd.sh start -Dlogin_interval=300,start_timeout=60,udp_attempts=3,
       udp_interval=20
  ```

  In the above example,

  – `login_interval` is the number of seconds before the OS/390 endpoint tries to login to the gateway again after a failure.

  – `start_timeout` is the number of seconds before a timeout occurs while trying to login to a gateway.

  – `udp_attempts` is the number of times an OS/390 endpoint will transmit an initial login request before trying another gateway.

  – `udp_interval` is the number of seconds between initial login request attempts.

- **Logging Levels for the lcfd Daemon:** You can choose from four levels of logging when you start the lcfd daemon with the `-d` option. Log level `-d 0` turns off logging completely; there will be no entries written to any log or trace file. Log level `-d 4` provides very detailed logging and creates extremely large log files. Log level `-d 4` may cause log files to fill up quickly and wrap, effectively wiping out the log data you're trying to collect. Therefore, you should not run with log level `-d 4` for long periods of time. We recommend running with log level `-d 2`. For example:

  ```
  lcfd.sh start -d 2
  ```

Stopping the lcfd Daemon: When the lcfd daemon starts normally, it records its process ID number in the file /etc/Tivoli/lcf/dat/lcfd.pid. To stop the lcfd daemon, issue the command:

```
lcfd.sh stop
```

The stop command uses the process ID in the lcfd.pid file to stop the lcfd daemon. If the command completes successfully, the lcfd daemon stops and displays no messages. If the command is unsuccessful, it displays the message:

```
Unable to get process id of the Tivoli LCF daemon (lcfd.pid)
```

This usually means that the lcfd daemon did not start successfully or that the lcfd.pid file has been deleted. To stop the lcfd daemon, get the process ID number by issuing the following command from the OMVS command line:

```
ps -ef | grep lcfd
```

The command response resembles the following message:

```
setup 587202585 1 - 20:56:48 ttyp0000 0:00 /usr/lpp/Tivoli/lcf/bin/
     os390/mrt/lcfd
```

In the above sample response, `587202585` is the process ID (which could be any number) and `setup` could be any user ID. Once you have the process ID number, you can stop the lcfd daemon by issuing the command:

```
kill -9 587202585
```

Avoiding Some Common lcfd Daemon Problems: We'd like to tell you about a couple of problems we experienced and tell you how to recognize and avoid them.

lcfd Daemon Starts Then Immediately Stops: We have found that if you attempt to start the lcfd daemon and it starts and then immediately stops, it might be due to a space shortage in the file system caused by some very large .log or .trc files or CEEDUMPS. While we are unable to recommend an exact minimum space requirement, we can tell you that our /etc HFS is currently allocated with 39 cylinders on a 3390 DASD. We suggest you delete the .log and .trc files and all CEEDUMPS from the /etc/Tivoli/lcf/dat directory and then try to start the lcfd daemon again.

lcfd Daemon Starts Normally but Applications Fail to Run: We have seen instances where the lcfd daemon starts normally and connects to the gateway, but when we attempt a User Administration or Security Management operation, the operation fails. When this occurs, the following message appears on the TMR server:

```
net_recv: connection reset: peer=xx.xx.xx.xx+9494
```

In addition, the following message appears in the lcfd.log file:

```
EX: LogMsg: /etc/Tivoli/lcf/dat/lcfd.log: fstat failed: EDC5113I Bad file
     descriptor
```

| APAR OW44995 is open to resolve this problem. We suspect, however, that it might be due to a permissions problem. Here are some potential problem areas to check:

- Specifically, make sure that the RACF definitions outlined in the program directory are correctly entered.
- Make sure the Tivoli run-time directory structure /etc/Tivoli/lcf/dat has the correct permissions defined, as described in Table 17 on page 191.
- Make sure _BPX_SHAREAS and _BPX_SPAWN_SCRIPT environment variables are not set or are set to NO in the environment where the lcfd daemon is started. Usually, these environment variables are set to values other than NO for performance reasons. If this is true on your system, issue one of the following before starting the lcfd daemon:

  ```
  export _BPX_SHAREAS="NO"
  export _BPX_SPAWN_SCRIPTS="NO"
  ```

  or

  ```
  su
  ```

  While, for OMVS users, substituting `su` for the `export` statements may not make sense, it does work and support is investigating this to understand the relationship.

Confirming Your Installation Using a Web Browser

We want to highlight that the Tivoli Framework provides the capability for you to monitor and maintain your endpoints using a web browser. We really like this feature because it makes it possible to check on the status of your endpoint from any place where you have web access to your installation's internal network. We recommend you explore and use this feature.

Experiences with TME 10 User Administration for OS/390

During our testing, we used Tivoli User Administration to manage our RACF user IDs. We'd like to tell you how we set this up and our experiences with using User Administration.

However, before we discuss how we set up our environment, we must clarify some terminology—specifically, *profiles.* In RACF, when a user ID is added to the RACF database, it is referred to as a user ID *profile.* In Tivoli, when a user ID is added to the Tivoli database, it is called a user ID *record.* The term *profile* in Tivoli refers to an entity that contains a collection of user ID records. That is, many user ID records are stored in a profile.

Planning to Manage RACF User IDs in Tivoli

You can simplify management of RACF user IDs in Tivoli by analyzing your RACF database to determine an appropriate scheme to map your user IDs to Tivoli profiles.

Relating Performance To Profile Size: The TME 10 User Administration for OS/390 documentation describes how to set up profiles and assign user records to those profiles, and gives examples of the numbers of records that can be stored in a profile. The number of records you choose to assign to a particular profile will affect performance. This number depends on the speed of the server (for our tests we are using a 350 MHz TMR server with 320 MB of RAM, running Windows NT Version 4.0) and on the number of RACF segments or fields that are filled in on each RACF profile. While we do not have a recommended formula for calculating this number, we suggest that you keep the number small, approximately 500 records per profile, and adjust this number up or down depending on how well your user IDs perform with TME 10 User Administration.

In our installation, our goal was 500 user IDs per profile, although we allowed some profiles to contain as many as 2000 user IDs.

Deciding What Profiles To Create: We began by listing all of the user IDs in our RACF database. We then looked for some pattern to the user IDs and then tried to group them into a logical set of profiles. We decided on the following rules for our profile layout:

- One profile for each letter of the alphabet. When the number of user IDs that begin with a particular letter is greater than 500, we create a second profile for that letter, then a third, and so on. For example: A1_Profile, A2_Profile, A3_Profile, B_Profile, C_Profile . . . .
- For user IDs that belong to a series, such as UID0001, UID0002, UID0003, and so on, we create a profile for the series. For example, UID_Profile.
- All user IDs that begin with a number are stored in Number_Profile.
- All user IDs that begin with the special characters @, #, or $ are stored in Special_Profile.

The following figure illustrates the profiles we created:

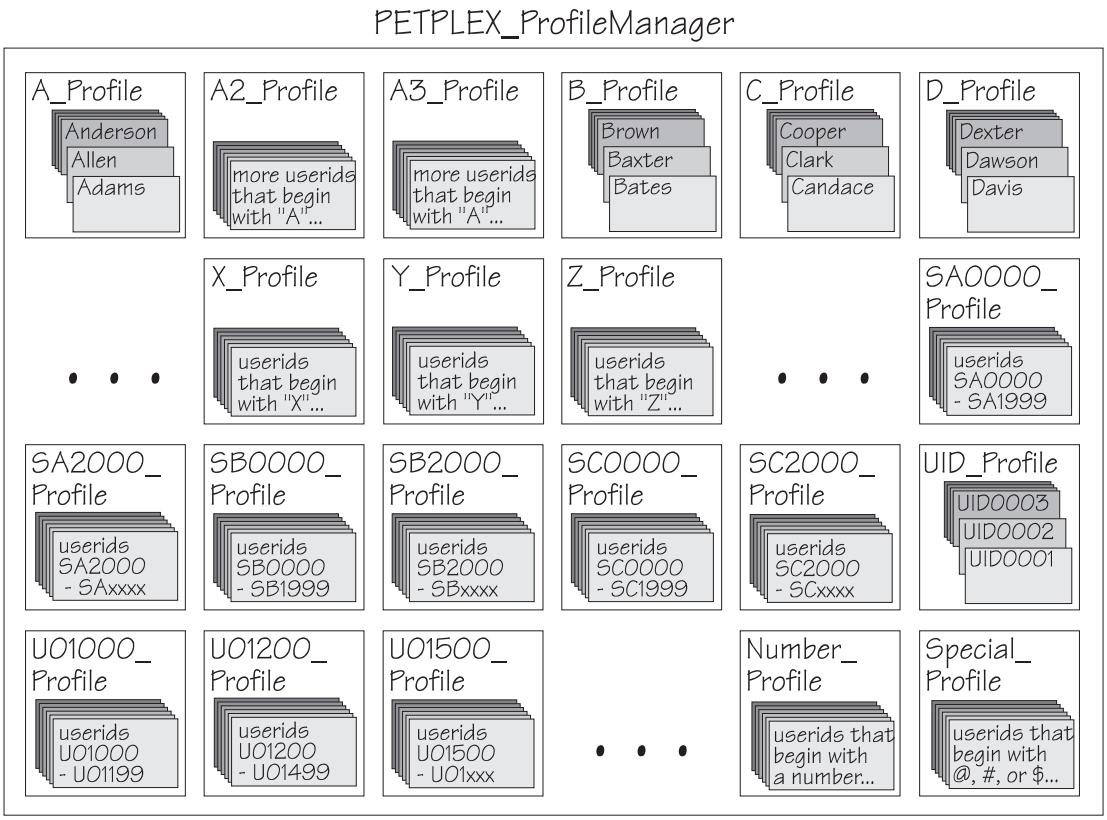


Figure 59. Our Tivoli User ID Profiles

Setting Up Profiles on the TMR Server

You can use policies to enforce your installation's standards for various attribute field values in user ID records.

Policy Region and Profile Manager: We created each of the profiles under one policy region and profile manager.

Using Default and Validation Policies: A default policy allows predefined values to be applied to unspecified fields when new user ID records are created. A validation policy allows field values to be verified against predefined rules when a record is created or changed. You might not be aware of these functions, as they are not discussed in the OS/390-specific documentation for User Administration or Security Management. We wanted to highlight these functions because we believe you will find them quite useful. You can read about how to set up default and validation policies in *TME 10 User Administration User and Group Management Guide Version 3.6*.

We customized the OS/390 default policy scripts to automatically fill in some fields, such as a default group and owner, when a new user ID is created.

Disabling Non-OS/390 Default and Validation Policies: After we created each profile, we disabled the default and validation policies for the UNIX, Windows NT, and Netware segments using the **wsetdefpol** command. We did this because we are distributing these profiles only to OS/390 and, therefore, there is no reason to run default validation scripts for those other segments every time a new user ID is created.

Populating the Tivoli Database with wpopusers

Following the instructions in *TME 10 User Administration User's Guide Supplement for OS/390 Version 3.6*, we populated the Tivoli database with RACF user IDs. Here are the steps we used:

1. Ran the RACF utility IRRDBU00 to unload the RACF database.
2. Ran the DFSORT utility to sort through all of the RACF records, select only the user ID (type 0200) records, and copy them to a data set with one user ID per line.
3. Ran DFSORT again to further sort the user IDs into separate data sets for each of the profiles that we created. When we were done with this step, we had one data set for each profile.
4. Used **ftp** to place these data sets on the TMR server and used each of them as input to the Tivoli **wpopusrs** utility.

Before Populating With wpopusrs: Before doing any work with User Administration, we recommend that you install PTF UW55092. This PTF provides fixes for several APARs and allows populated user IDs to be modified without having to change the password. It also prevents a CEEDUMP from occurring during some distributes.

Problems With Revoke and Resume Dates After a Populate: If any user ID has ever had a revoke or resume date entered on RACF, **wpopusrs** populates the date to the Tivoli database—even if the date has passed and is no longer displayed on RACF. Such dates, if they have already passed, cause a distribute of the user record back to RACF to fail. This is because RACF does not allow dates in the past to be entered for revoke or resume dates. Therefore, after populating the Tivoli database from RACF, make sure there are no old revoke or resume dates on any user record because Tivoli does not remove these dates before distributing the records back to RACF. Furthermore, a user ID that fails distribution because of this problem will be corrupted. PTFs UW63749 and UW63750 are now available to fix this problem in the OS/390-based TMR server environment. However, patches are not yet available for the non-OS/390 TMR server environment. Note that these PTFs introduce a characteristic, described in "User Administration" on page 181, where the date 01/01/71 is sometimes populated into the revoke or resume date fields.

Validation Problems With wpopusrs: All user records populated into the Tivoli database using **wpopusrs** are processed through the validation policy scripts. This is a good feature if you are trying to set new policy for your installation and believe that not all existing user IDs on RACF will comply. When running **wpopusrs**, user IDs failing validation will not be populated to the database. Instead you will receive an error message notifying you that the user ID failed to be added to the database. To get that user ID into the Tivoli database, you must change the field on RACF (through TSO) to comply with your validation scripts.

You may want to change the validation scripts to accommodate all of your RACF user IDs or completely disable validation. If you choose to disable validation, remember it must be disabled in *each* profile.

WPOPUSRS Takes a Long Time With Validation: In our testing, with validation enabled, it took 3 hours to populate 1000 user IDs. With validation disabled, the same 1000 user IDs were populated in less than 10 minutes. We recommend running with validation disabled.

Tivoli Management Framework for OS/390 Framework Endpoint

Clarifying the Release Notes Workaround: User IDs that have a TSO region size (tso tlsize) specified, but do not have TSO maximum region size (tso maxsize) specified on RACF, will fail validation when being populated to a Tivoli profile. A workaround to this problem is provided in *TME 10 User Administration for OS/390 Release Notes V3.6*, but we'd like to clarify some of the information presented there. We are referring to Known Defect 10, the description of which begins on page 6 of the Release Notes.

On page 7, you are provided with 4 steps to remove the MAXSIZE check from the validation policy for racf_tso_tlsize. Those steps should read as follows:

Note: If you are on a Windows NT system, open a DOS window and start a BASH shell. The commands that follow will not run correctly if the bash shell is not started. If you are on an AIX system, there is already a shell running.

a. Extract the TSO size validation policy and write it to a file:

```
wgetpolm -v @UserProfile:Profile_Name racf_tso_tlsize > tlsize.orig
```

where `Profile_Name` is the actual profile name

b. Copy this original file to a new file

```
cp tlsize.orig tlsize.new
```

c. Then edit tlsize.new. Comment out 7 lines after the set of comments. Following is an example of what tlsize.new should look like:

```
##########################################
# SIZE is valid, now make sure it is less than
# the possibly passed in MAXSIZE.
# We first syntactically check MAXSIZE
# so we don't cause shell errors.  If MAXSIZE
# is bad, we assume that SIZE is good because
# this script is to test SIZE not MAXSIZE.
# It's not SIZE's fault if MAXSIZE is bad.
##########################################
# if [ -n "$MAXSIZE" ]; then
#    # Now make sure SIZE <= MAXSIZE
#    if [ $SIZE -gt $MAXSIZE ]; then
#         echo FALSE
#         exit $E_OK
#    fi
# fi
```

d. Put the changed (or fixed) tso size policy back into the profile.

Note: There are very subtle changes in the following command in comparison to the Release Notes (the single quotes have been moved and tmsize has been changed to tlsize). Following is the command as it should be run:

```
wputpolm -v 'args=$racf_tso_tlsize' @UserProfile:profile_name \
    racf_tso_tlsize < tlsize.new
```

where `profile_name` is the actual profile name.

Figure 60. Release Notes Page 7—Removing the MAXSIZE Check from the Validation Policy for racf_tso_tlsize.

On page 8, you are provided with 4 steps to remove the MAXSIZE check from the validation policy for racf_tso_tmsize. Those steps should read as follows:

Note: If you are on a Windows NT system, open a DOS window and start a BASH shell. The commands that follow will not run correctly if the bash shell is not started. If you are on an AIX system, there is already a shell running.

a. Extract the TSO maximum size validation policy and write it to a file:

```
wgetpolm -v @UserProfile:Profile_Name racf_tso_tmsize > tmsize.orig
```

where `Profile_Name` is the actual profile name.

b. Copy this original file to a new file:

```
cp tmsize.orig tmsize.new
```

c. Then edit tmsize.new. Comment out the following code:

```
#     # Now make sure MAXSIZE >= SIZE
#     if [ -n "$SIZE" ]; then
#        if [ $MAXSIZE -lt $SIZE ]; then
#           echo FALSE
#           exit $E_OK
#        fi
#     fi
```

Then add the following code to replace the above:

```
# Now make sure MAXSIZE >= SIZE
if [ -n "$SIZE" ]; then
     # If non-zero, ensure MAXSIZE >= SIZE
          if [ $MAXSIZE -ne 0 ] ; then
               if [ $MAXSIZE -lt $SIZE ]; then
                    echo FALSE
                    exit $E_OK
               fi
          fi
fi
```

d. Put the changed (or fixed) TSO maximum size policy back into the profile:

```
wputpolm -v @UserProfile:profile_name racf_tso_tmsize < tmsize.new
```

where `profile_name` is the actual profile name.

Figure 61. Release Notes Page 8—Removing the MAXSIZE Check from the Validation Policy for racf_tso_tmsize.

Populating From the GUI: Populate can only be done with the command line interface (CLI) **wpopusrs** command. It is not supported on the graphical user interface (GUI).

Our Operational Experiences

We have the following experiences to share based on our User Administration usage and testing.

Ensuring a Secure Network Connection: The transmission of data between the TMR server and the OS/390 endpoint uses simple encryption rather than the Data Encryption Standard (DES). While we have not been able to read the data stream, we believe it would not be difficult to break the encryption. For a secure network connection, we recommend that the TMR server communicate with the OS/390 endpoint using a dedicated local LAN that is not accessible by others.

Distributing a Profile For the First Time: After populating your Tivoli database with **wpopusrs**, the first time a profile is distributed to OS/390 with a change to any of the user records, all user records in that profile will be distributed. Because of this, the first distribute of a profile with just one minor change to one record will take

much more time than expected as it sends all records to RACF. All subsequent distributions with minor changes will execute in a much shorter time than the initial distribution.

Operational Problems With Revoke and Resume Dates: If you enter a revoke or resume date on a user record in Tivoli and distribute it to RACF, you cannot successfully distribute the record again once those dates pass. You must remove any old revoke or resume dates from the Tivoli record before you can successfully distribute the record to RACF. This is because RACF does not allow dates in the past to be entered for revoke or resume dates, and Tivoli does not remove such dates before distributing to RACF. Furthermore, a user ID that fails distribution because of this problem will be corrupted. PTFs UW63749 and UW63750 are now available to fix this problem in the OS/390-based TMR server environment. However, patches are not yet available for the non-OS/390 TMR server environment. Note that these PTFs introduce a characteristic, described in "User Administration" on page 181, where the date 01/01/71 is sometimes populated into the revoke or resume date fields.

Deleting User IDs That Own Resources:When you attempt to delete a user ID from Tivoli/RACF, if that user ID owns any resources, it cannot be deleted. This is a RACF rule which can cause problems in the Tivoli environment. Since the user record has been deleted from the Tivoli database, if the delete fails on RACF, there is no entry remaining on Tivoli. To delete this user ID, you must do the following:

1. Correct the reason for the delete failure on RACF.
2. Do *either* of the following:
   - Add and then delete the user ID again through User Administration

   or
   - Manually delete the user ID through TSO/E

Ensuring Successful Distributes: Make sure that all distributes to RACF are successful. Check the User Management Notice Group after every distribute for error messages. If the distribute was unsuccessful, correct the problem for the user records that failed and repeat the distribute.

Experiences with TME 10 Security Management for OS/390

Also during our testing, we planned to use Tivoli Security Management to manage all of our RACF groups, data sets, and resources. However, due to the large effort involved in populating the Tivoli Security Management database, we chose to narrow our implementation and focus on a new concept in RACF and Security Management called *roles.* Roles provide a new security element that links groups and users with resources and defines the authority the groups and users have to those resources.

We are currently managing one role, one group (with five user IDs connected), and nine generic data set resources. We expect to manage more of our RACF resources in the future. Currently, we have the following observations to share with you:

Recommended Service:Before doing any work with Security Management, we recommend that you install PTFs UW55460 and UW55461. These will correct two possible out-of-memory exceptions as well as some problems populating resources. We also recommend you install patch 3.6-SEC-0005 which is available from the Tivoli support web site. This patch will enable the administrator to modify data set resource records using Security Management.

GUI Problems With Roles Under Security Management 3.6.1: Although we have not tested extensively under Tivoli version 3.6.1, we have experienced the following problems when creating or modifying role records under Security Management 3.6.1:

- When using the GUI to create or modify a role record on Tivoli, you cannot add the access rights to a resource defined to the role.
- When using the GUI to modify a role record on Tivoli, you cannot delete a resource from the role.

However, while these functions do not work under the GUI, note that you can use the Tivoli CLI **wmodsec** command to add the access rights or delete a resource from a role.

After further investigation, Tivoli support determined that both of the above problems resulted from installing the Tivoli 3.6.1 upgrade. If you upgrade your TMR Server from version 3.6 to version 3.6.1, make sure the Security Management and Security Management for OS/390 products are both at the same level. If you upgrade Security Management to version 3.6.1 but leave Security Management for OS/390 at version 3.6, you will experience the above problems. Furthermore, the Security Management for OS/390 version 3.6.1 upgrade patch contains a defect which also causes these problems. To avoid this, do one of the following:

- **New Installation of Security Management for OS/390:** If this is a new installation of Security Management for OS/390, ***do not*** install Security Management for OS/390 version 3.6 and then attempt to upgrade to version 3.6.1. Instead, there is a Security Management for OS/390 version 3.6.1 product (not a patch) on the same CD-ROM. Directly install this product instead of the upgrade patch.
- **Upgrade of Security Management for OS/390 Version 3.6 to 3.6.1:** If you already have Security Management for OS/390 version 3.6 installed, you will need to force the installation of the Security Management for OS/390 version 3.6.1 product (not the patch) by mounting the Tivoli Security Management for OS/390 Version 3.6.1 CD-ROM and issuing the following command:

  ```
  winstall -c d: -i SEC390 ALIDB=!
  ```

  In the above command *d:* is the drive specification of the CD-ROM drive on our Windows NT TMR server. This installation will complete with errors, but the required fixes will already be installed by the time the failure occurs.

  After force installing the Tivoli Security Management for OS/390 v3.6.1 - Tivoli Server install pack, you must then install the Tivoli Security Management for OS/390 Upgrade to v3.6.1 - Tivoli Server install pack using the normal Tivoli Desktop patch installation process.

Note that these problems do not affect any of the Security Management Gateway software. You can install and upgrade the Gateway software normally.

Command Line Interface (CLI) Differences: If the User Administration **wrunusrcmd** command works successfully, but you receive a `net_recv: connection reset` message when running the Security Management **wrunseccmd** command, it is because the user ID under which Security Management runs needs super user authority. Service is investigating the correct way to do this. However, we have found that stopping the lcfd daemon, issuing an **su** command, and then starting the lcfd.sh daemon works best in our environment. Check the PSP bucket for more information on this problem.

Security Management and User Administration Use Different Profiles: Security Management is separate from User Administration. While they both use profile managers and profiles to contain records, the profiles are different. There is a profile type for User Administration (users) and a profile type for Security Management (groups, resources, and roles). Please note that the group record available in User Administration is provided for managing groups on UNIX operating systems only. Group record support for RACF is provided through Security Management.

Creating, Modifying, and Deleting Groups: Through Tivoli, you can create, modify and delete groups. However, not all RACF group fields are supported. Only ″Group Name″ and ″User IDs Connected to the Group″ are supported. Therefore, when creating a group, the Tivoli Administrator can specify only the group name and the user IDs to connect to the group. All other group fields are created on RACF with default values. When modifying groups, only the user IDs connected to the group can be changed. All other fields are left unchanged.

Identical Records In the Same Profile Manager: If you are populating groups from RACF to a Security Management profile and an identical group name already exists in another profile under the same profile manager, a message will be received indicating the group name(s) already exist. The populate will continue, but only unique group names will be populated to the specified profile.

Security Management Resource Support: RACF resources include data set and general resource profiles. While support for these profiles is available, not all fields in these profiles can be manipulated through Tivoli. Only the following resource fields are supported:

- Resource Name
- Default Access (UACC)
- Access Time Restrictions (if supported for that resource on RACF)
- Audit Control Parameters

All other fields are created on RACF with default values. When modifying resources, only the Default Access, Access Time Restriction, and Audit Control Parameter fields can be changed. All other fields are left unchanged.

New RACF Role Resource Profile: RACF provides a new *role* resource profile in support of Tivoli roles. In addition to this resource type, new TME segments have been added to the group, data set, and general resource RACF profiles. While the RACF commands such as **ralter**, **altdsd**, and **altgroup** have been changed to allow you to add, modify or delete the TME segment, the TME segment is intended to be manipulated *only* by Tivoli.

RACF Role Limitation: During our testing, we found a RACF limitation that may affect your use of roles. The limitation is that there can only be a maximum of 253 combined groups and resources added to a role at any one time. This number also includes all groups and resources of a parent role, if one is defined. For example, if you want to add 50 groups and 100 resources to a role called ROLEB, and that role also has a parent role, ROLEA, which has 25 groups and 75 resources associated with it, the total number of groups and resources that would be associated with ROLEB is 250 (150 from ROLEB and 100 from ROLEA). This is below the 253 limitation and should work correctly. However, to extend this example, if ROLEB has 125 resources instead of 100, then the total number would be 275 combined resources, and the distribute would not work correctly. If you inadvertently exceed this limitation, neither Security Management nor RACF display any error messages to this effect. However, you will find in RACF that the groups and resources that

exceed the first 253 groups and resources (ROLEB is first, then ROLEA) are the *only* ones that will be attached to the role in RACF—they will appear in the TME segment on the role. The first 253 groups and resources will not show up in the TME segment of the role on RACF. We suggest you add groups and resources to a role in combinations of less than 253 until this limitation can be resolved.

Defining Groups:When defining groups, you can connect user IDs to the group. The user IDs can come from two sources. The first source is the TME user list, that is, user IDs populated from RACF to Tivoli. The second source is the RACF user list—user IDs that *you* defined to the group, which are known to exist in RACF. A couple of things to remember about these two sources:

1. When you populate a group from RACF to Tivoli, any user IDs connected to the populated group are added to the RACF user list. They are *not* added to the TME user list, even if the user IDs can already be found in the TME user list.
2. When creating or modifying groups, if you want to connect user IDs to the group and your User Administration profiles contain thousands of user records, it may take several minutes to display the list of user IDs after selecting EDIT MEMBER LIST from the Edit Options List. This is because Tivoli is preparing to display all user administration user IDs from the TME user list for you to make a selection. If you intend to enter user IDs directly into the RACF user list and do not need to list the TME user list, select SHOW ALL and then select RACF USER LIST. This will bypass all of the processing associated with TME user list.

Defining Roles:When defining roles, you can connect resources and groups to the role. Just like user IDs, there are two sources for resources and two sources for groups. For resources, the sources are:

1. The RACF TME resource list—resources populated from RACF to a resource profile and resources newly created in a profile on the Tivoli desktop.
2. The RACF resource list—a list of resources known to exist on RACF.

For groups, the sources are:

1. The TME group list—groups populated from RACF to a group profile and groups newly created in a profile on the Tivoli desktop.
2. The RACF group list—a list of groups known to exist on RACF.

The same populate and create/modify rules apply as they did for User Administration.

Populating From the GUI or CLI: Populating can be done through either the CLI or the GUI (unlike User Administration, which has only the CLI). We thought the GUI was excellent.

Ensuring Successful Distributes:Make sure that all distributes to RACF are successful. Check the notice groups after every distribute for error messages. If the distribute was unsuccessful, correct the problem that caused the failure and repeat the distribute.

Where to Find More Information

In addition to the publications we mentioned in this chapter (which are listed in “Appendix D. Useful Publications and Web Sites” on page 217), we recommend you check out the following Web sites for important information:

```
http://www.tivoli.com
http://www.support.tivoli.com
```

The latter address takes you to a Web site intended for those who have purchased a license for a Tivoli product, and requires you to register and obtain a user ID and password.

Part 4. Appendixes

Appendix A. Some of Our Parmlib Members

This section describes how we have set up some of our parmlib members for OS/390. The following table summarizes the highlights of new and changed parmlib members for OS/390 R9 and R10. Samples of these and other parmlib members are on the samples page of our Web site.

Table 18. Highlights of Parmlib Changes for OS/390 R9 and R10

Parmlib Member	OS/390 Release	Change	Related To
BPXPRMxx	R9	Added VERSION and, SYSPLEX(YES) statements, added MOUNT statements for HFS files systems mounted at the sysplex root, and added NOAUTOMOVE to the MOUNT statements for the system specific HFSs.	Shared HFS support
	R10	No changes were needed for OS/390 R10.	
IFAPRDxx	R9	No changes were needed for OS/390 R9.	
	R10	No changes were needed for OS/390 R10.	
ISFPRMxx	R9	No changes were needed for OS/390 R9.	
	R10	Added a server group definition.	SDSF sysplex support.
LOADxx (Actually in SYS0.IPLPARM. See note below.)	R9	Changed PARMLIB statement to use SYS1.PETR9.PARMLIB	Concatenated parmlib
	R10	Changed PARMLIB statement to use SYS1.PETR10.PARMLIB	Concatenated parmlib
LPALSTxx	R9	Added: SYS1.V2R9M0.SIATLPA	JES3
	R10	Added: SYS1.V2R10M0.SIATLPA	JES3
PROGxx (APF additions)	R9	Added: SYS1.V2R8M0.SHASLINK SYS1.V2R8M0.SHASMIG (Functionally equivalent and unchanged from R8 JES2.)	JES2
	R10	Added: SYS1.V2R10M0.SHASLINK SYS1.V2R10M0.SHASMIG	JES2

Parmlib Members

Table 18. Highlights of Parmlib Changes for OS/390 R9 and R10 (continued)

Parmlib Member	OS/390 Release	Change	Related To
PROGyy (LNKLST)	R9	Added to LNKLSTxx: SYS1.V2R8M0.SHASLINK SYS1.V2R8M0.SHASMIG (Functionally equivalent and unchanged from R8 JES2.)	JES2
		Added to LNKLSTxx: SYS1.V2R9M0.SIATLIB SYS1.V2R9M0.SIATLINK SYS1.V2R9M0.SIATMIG	JES3
	R10	Added to LNKLSTxx: SYS1.V2R10M0.SHASLINK SYS1.V2R10M0.SHASMIG	JES2
		Added to LNKLSTxx: SYS1.V2R10M0.SIATLIB SYS1.V2R10M0.SIATLINK SYS1.V2R10M0.SIATMIG	JES3

Note: In OS/390 R6, we changed LOADxx to use a generic name, IEASYMPT, for our IEASYMxx member, and we have successfully used the name IEASYMPT for our migrations through all subsequent OS/390 releases. Only the entries in SYS0.IPLPARM changed.

Comments about IFAPRD00

When you order OS/390, IBM builds and ships to you a tailored IFAPRD00 parmlib member reflecting your order. The IFAPRD00 member we have on our Web site looks like the member we would receive if we were a customer installing OS/390, except for the modifications we made at the end of the member to customize systems Z0, JE0, and JF0. We share IFAPRD00 across all systems in the sysplex.

We have almost every element and feature of OS/390 enabled because we are a test organization. Most customers would have more of the optional features disabled. Do not add or enable any product elements or features in your IFAPRD00 parmlib member without notifying IBM. See *OS/390 MVS Product Management* for more information, along with the following comments about our IFAPRD00 member:

- The first product statement in the member enables OS/390 itself. Note that OS/390 ignores this statement and always registers as enabled. IBM puts this statement in the IFAPRD00 parmlib member shipped with every OS/390 order to be consistent with OS/390's policy of providing IFAPRD00 product statements for every OS/390 priced feature or program product equivalent.
- We did the following to customize our IFAPRD00 member:
  - We left in place the WHEN(HWNAME(*)) statement at the beginning of the IFAPRD00 member so that the product statements following it would apply to all systems using this member.
  - We added WHEN(SYSNAME...) statements at the end of the member to enable JES3 and the BDT-related features on systems Z0, JE0, and JF0 only. That's because we are treating the optional JES3 and BDT-related features as licensed only to CPCs that run SYSNAMEs of Z0, JE0, and JF0. The WHEN(SYSNAME...) statements at the end of the member selectively change the enablement for just those features on just those systems.

The enablement policy still applies for all other products and features whose product statements precede, and are not changed by, the WHEN(SYSNAME...) statements.

We could have customized for JES3 and the BDT-related features using different approaches:

- We could have used WHEN(HWNAME...) statements instead of WHEN(SYSNAME...) statements to specify the hardware names of the CPCs as defined to HCD, because systems Z0, JE0, and JF0 each run on their own CPC.
- We could have created separate parmlib members (such as IFAPRDZ0, IFAPRDE0, and IFAPRDF0) containing only the unique WHEN(SYSNAME...) and product statements for those systems, and IPLed with an IEASYSxx parmlib member containing PROD=(00,Z0,E0,F0).

We make a point of explaining this customization process because software is licensed on a physical CPC basis. When you receive your tailored IFAPRD00 member from IBM and want to use it in a sysplex, you might have certain elements, features, or products that you will run only on the selected CPCs for which they are licensed. If you are using 1 IFAPRDxx member for all systems to share, you must selectively enable those elements, features, or products for the CPCs for which they are licensed.

If you make copies of the IFAPRD00 member for systems on different CPCs, you must disable the elements, features, or products that are not licensed for those CPCs.

You must contact IBM if you enable any products or features that are disabled in the IFAPRD00 member that IBM ships to you. See *OS/390 MVS Product Management* for more information.

- We ordered and installed our OS/390 system through ServerPac. We included the DCF and PSF for OS/390 program products as part of the ServerPac order, so the appropriate IFAPRD00 statements were supplied for those products. See *OS/390 Planning for Installation* for information about ServerPac compared with CBPDO delivery mechanisms for OS/390.

  In cases where OS/390 features have program product equivalents, we followed the OS/390 recommendation of ordering the OS/390 features rather than the program product equivalents (DFSORT, for example, is enabled as an OS/390 feature and disabled as a stand-alone program product).

  That's because we have only OS/390 systems in our sysplex. If your computing environment also contains MVS/ESA SP level systems, there might be some distributed systems license option (DSLO) advantages to ordering the program product equivalents rather than the OS/390 features.

Appendix B. Some of Our RMF Reports

In this appendix we include some of our RMF reports, as indicated in “OS/390 Performance” on page 34.

RMF Monitor I Post Processor Summary Report

The following figure contains information from our *RMF Monitor I Post Processor Summary Report*. Some of the information we focus on in this report includes CP (CPU) busy percentages and I/O (DASD) rates. This report contains information for the same date and time interval as the report in Figure 64 on page 213. We edited the report for presentation purposes, making it more compact; the numbers, however, are the actual numbers from our report. Highlighted figures represent machine totals for CPU busy.

```
                        R M F   S U M M A R Y   R E P O R T

OS/390                                START 05/07/1998-18.30.09  INTERVAL 00.30.00
REL. 02.05.00         RPT VERSION 2.4.0          END   05/07/1998-19.00.09  CYCLE 0.100 SECONDS

                                 TOTAL LENGTH OF INTERVALS 00.30.00
 INT   CPU   DASD  DASD   JOB   JOB   TSO   TSO   STC   STC  ASCH  ASCH  OMVS  OMVS SWAP DEMAND
MM.SS  BUSY  RESP  RATE   MAX   AVE   MAX   AVE   MAX   AVE   MAX   AVE   MAX   AVE RATE PAGING

                        SYSTEM ID JD0
29.59 67.1    26 153.4     0     0     0     0   249   247     0     0     3     3 0.00   3.22

                        SYSTEM ID JA0
29.59  73.5   27 211.4     1     0     0     0   251   248     0     0     2     2 0.00   0.01
                        SYSTEM ID JB0
30.00  77.7   27 227.4     0     0     0     0   252   248     0     0     3     3 0.00   0.22
      99.98
                        SYSTEM ID J90
30.00  72.6   25 472.2     1     0     0     0   250   249     0     0     2     2 0.00   0.00
                        SYSTEM ID JF0
29.59  72.9   24 472.5     1     0     0     0   348   347     0     0     3     3 0.00   0.00
      87.63
                        SYSTEM ID J80
29.54  42.4   23 491.3     2     0     7     7   251   250     0     0     3     3 0.00   6.27
                        SYSTEM ID JC0
30.00  73.1   26 336.6     1     0     0     0   252   249     0     0     1     1 0.00   1.77
                        SYSTEM ID JE0
30.01  73.6   27 307.8     0     0     0     0   350   349     0     0     5     5 0.00   2.07
      97.28
                        SYSTEM ID Z0
29.59  28.1   24 168.9     1     0     0     0   355   354     0     0     9     9 0.01   0.00
                        SYSTEM ID Z1
30.00   8.4    6  15.2     0     0     0     0   222   221     0     0     2     2 0.00   0.00
      35.37
                        SYSTEM ID JG0
29.59  17.8   15 110.8     0     0     0     0   231   230     0     0     0     0 0.00   0.00
                        SYSTEM ID JH0
30.00  17.7   17 112.5     0     0     0     0   231   230     0     0     0     0 0.00   0.00
                        SYSTEM ID JI0
30.00  17.8   17 113.6     0     0     0     0   231   230     0     0     0     0 0.00   0.00
      32.63
                        SYSTEM ID TPN
30.00  85.9    3  35.6     0     0     0     0   226   224     0     0     2     2 0.00   0.00
      89.39
```

Figure 62. Example RMF Monitor I Post Processor Summary Report

RMF Monitor III Online Sysplex Summary Report

The following figure contains information from the RMF *Monitor III Online Sysplex Summary Report*. This is a real-time report available if you are running WLM in goal mode. We highlighted some of our goals and actuals for various service classes and workloads. At the time this report was captured we were running 903.7 CICS

transactions/second. Note that this report is a snapshot, as opposed to Figure 64 on page 213 , which is based on a 1/2-hour interval.

```
HARDCOPY  RMF 2.4.0  Sysplex Summary - UTCPLXJ8       Line 1 of 62
Command ===>                                                Scroll ===> CSR
WLM Samples: 480     Systems: 14 Date: 05/07/98 Time: 18.45.00 Range: 120   Sec

                     >>>>>>>>XXXXXXXXXXXXXXXXXX<<<<<<<<

Service Definition: WLMDEF01              Installed at: 03/02/98, 14.13.47
     Active Policy: WLMPOL01              Activated at: 03/02/98, 14.13.19
               ------- Goals versus Actuals --------  Trans --Avg. Resp. Time-
               Exec Vel  --- Response Time ---  Perf  Ended  WAIT EXECUT ACTUAL
Name     T  I  Goal Act  ---Goal--- --Actual--  Indx  Rate   Time   Time   Time
CICS     W          N/A                          903.7 0.000  0.232  0.217
CICS     S  2       N/A  0.600 80%         92%  0.50  416.2 0.000  0.186  0.231
CICSCONV S  3       N/A  1.000 90%        5.4%  ****  0.617 0.000  16.55  16.55
CICSDEFA S  3       N/A  1.000 90%         97%  0.50  406.3 0.000  0.353  0.211
CICSMISC S  3       N/A  1.000 90%        100%  0.50  80.67 0.000  0.045  0.045
CICSRGN  S  2    60  53                         1.14  0.000
STC      W          54                          0.025 0.006  5.093  5.099
DB2HIGH  S  2    50  20                    2.46  0.000 0.000  0.000  0.000
IMS      S  2    50  58                    0.86  0.000 0.000  0.000  0.000
OMVS     S        0.0                         0.000 0.000  0.000  0.000
         3  5    10 0.0                  N/A  0.000 0.000  0.000  0.000
OMVSKERN S  1    40  77                    0.52  0.025 0.006  5.093  5.099
TPNS     S  2    70  75                    0.93  0.000 0.000  0.000  0.000
SYSTEM   W          42                          0.042 0.986  0.723  1.499
SYSSTC   S      N/A  72    N/A                   0.042
SYSTEM   S      N/A  34    N/A                   0.000 0.000  0.000  0.000
TSO      W          86                          0.033 0.000  59.79  59.79
TSO      S  2       86  2.000 AVG  59.79  AVG  29.9  0.033 0.000  59.79  59.79

⋮
```

Figure 63. Example RMF Monitor III Online Sysplex Summary Report

RMF Workload Activity Report in WLM Goal Mode

The following figure illustrates a couple of sections from our RMF *Workload Activity Report* in goal mode. This report is based on a 1/2-hour interval. Highlighted on the report you see 89.2% of our CICS transactions are completing in 0.6 seconds, and our CICS workload is processing 901.89 transactions per second.

```
                                   W O R K L O A D   A C T I V I T Y

      OS/390                  SYSPLEX UTCPLXJ8          DATE 05/07/1998          INTERVAL 30.08.534   MODE = GOAL
      REL. 02.05.00           RPT VERSION 2.4.0         TIME 18.30.00

                                        POLICY ACTIVATION DATE/TIME 03/02/1998 14.13.47

REPORT BY: POLICY=WLMPOL01   WORKLOAD=CICS       SERVICE CLASS=CICS       RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=2

TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT
AVG      0.00   ACTUAL                324
MPL      0.00   EXECUTION             216
ENDED  743227   QUEUED                  0
END/S  413.06   R/S AFFINITY            0
#SWAPS      0   INELIGIBLE              0
EXECTD 734169   CONVERSION              0
                STD DEV               857

            -------------------------------RESPONSE TIME BREAKDOWN IN PERCENTAGE-------------------------------  ------STATE------
SUB    P   TOTAL  ACTIVE  READY   IDLE  ----------------------------WAITING FOR-----------------------------  SWITCHED TIME (%)
TYPE                                     CONV  I/O PROD LOCK MISC                                             LOCAL SYSPL REMOT
CICS  BTE   83.1     1.9   24.9    0.0  55.4  0.3  0.0  0.6  0.0                                               53.9  19.9   0.0
CICS  EXE   22.3     4.1    7.2    0.1   0.0  0.7  9.8  0.0  0.4                                                9.8   7.1   0.0
IMS   EXE   10.7    10.7    0.0    0.0   0.0  0.0  0.0  0.0  0.0                                                0.0   0.0   0.0
SMS   EXE   15.4     1.9    0.0    0.0   0.0 13.4  0.0  0.2  0.0                                                0.0   0.0   0.0

VELOCITY MIGRATION:   I/O MGMT   N/A     INIT MGMT  N/A

        ---RESPONSE TIME---  EX   PERF
        HH.MM.SS.TTT         VEL  INDX
GOAL    00.00.00.600  80.0%
ACTUALS
*ALL                  89.2%  N/A   0.6
JA0                   90.4%  N/A   0.7    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JB0                   91.9%  N/A   0.7    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JC0                   88.9%  N/A   0.7    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JD0                   87.8%  N/A   0.8    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JE0                   87.9%  N/A   0.7    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JF0                   90.4%  N/A   0.6    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JG0                   99.0%  N/A   0.5    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
JH0                   99.0%  N/A   0.5    0.0   0.0   0.0    0.0
JI0                   99.0%  N/A   0.5    0.0   0.0   0.0    0.0
J80                   72.1%  N/A   2.0    0.0   0.0   0.0    0.0
J90                   89.3%  N/A   0.6    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0
Z0                     0.0%  N/A   N/A    0.0   0.0   0.0    0.0                                               0.0  0.0  0.0

                                          ----------RESPONSE TIME DISTRIBUTION----------
   ----TIME----     --NUMBER OF TRANSACTIONS--     -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
   HH.MM.SS.TTT     CUM TOTAL        IN BUCKET    CUM TOTAL   IN BUCKET  |....|....|....|....|....|....|....|....|....|....|
<  00.00.00.300          577K              577K         77.6        77.6  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<= 00.00.00.360          604K            27231          81.3         3.7  >>>
<= 00.00.00.420          625K            20630          84.1         2.8  >>
<= 00.00.00.480          641K            16143          86.2         2.2  >>
<= 00.00.00.540          653K            12317          87.9         1.7  >>
<= 00.00.00.600          663K            10053          89.2         1.4  >
<= 00.00.00.660          671K             7803          90.3         1.0  >
<= 00.00.00.720          678K             6624          91.2         0.9  >
<= 00.00.00.780          683K             5569          91.9         0.7  >
<= 00.00.00.840          688K             4552          92.5         0.6  >
<= 00.00.00.900          688K                0          92.5         0.0  >
<= 00.00.01.200          703K            15580          94.6         2.1  >>
<= 00.00.02.400          727K            23501          97.8         3.2  >>
>  00.00.02.400          743K            16392           100         2.2  >>

--------------------------------------------------------------------------------------------------------------------------------

REPORT BY: POLICY=WLMPOL01   WORKLOAD=CICS WORKLOAD

TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT   --DASD I/O--   ---SERVICE----   --SERVICE RATES--   PAGE-IN RATES    ----STORAGE----
AVG    128.37   ACTUAL                316   SSCHRT 259.3   IOC     47018K   ABSRPTN      5673   SINGLE     0.0   AVG     6690.99
MPL    128.37   EXECUTION             246   RESP    13.6   CPU    406590K   TRX SERV     5673   BLOCK      0.0   TOTAL    858905
ENDED 1622820   QUEUED                  0   CONN     1.6   MSO    671850K   TCB       22578.3   SHARED     0.0   CENTRAL  814207
END/S  901.89   R/S AFFINITY            0   DISC     7.4   SRB    191545K   SRB       11348.3   HSP        0.0   EXPAND  44697.9
#SWAPS      0   INELIGIBLE              0   Q+PEND   4.5   TOT      1317M   RCT           0.0   HSP MISS   0.0
EXECD 1030521   CONVERSION              0   IOSQ     0.1   /SEC   781853    IIT          10.7   EXP SNGL   0.0   SHARED     0.00
                STD DEV             3.374                                  HST           2.5   EXP BLK    0.0
                                                                           APPL %     1876.6   EXP SHR    0.
:
```

Figure 64. Example RMF Workload Activity Report in WLM Goal Mode

Appendix C. Document Distribution

We make our test report available through as many distribution mechanisms as possible. Because we publish quarterly, it is impractical to print hardcopy books for timely distribution of every edition. To provide you with the latest information, the report is currently available on the IBM Server Sales intranet site, the Internet, and through the IBMLink InfoCenter host environment.

Note that Internet World Wide Web page formats often change, so the steps indicated below might be slightly different from what you see on the day you access a particular page.

Note also that printed copies of the following editions are available through your normal publications ordering system:

- our December 1999 edition (GC28-1963-15)
- our December 1998 edition (GC28-1963-11)
- our December 1997 edition (GC28-1963-07) (as refreshed in March, 1998)
- our December 1996 edition (GC28-1963-03)
- our December 1995 edition (GC28-1236-02)
- *OS/390 Parallel Sysplex Recovery* (GA22-7286-00)

All of the above are also available on the IBM MKTTOOLS conference disk, the IBM intranet, the Internet, and IBMLink, as indicated below.

Our Own Web Site

We've built a Web site for our team, OS/390 Integration Test, where you can access all editions and formats of our test report (including our other publication, *OS/390 Parallel Sysplex Recovery*):

`http://www.s390.ibm.com/os390/support/os390tst/`

If you can't get to our Web site for some reason, listed below are alternate ways to access our report.

Availability on MKTTOOLS: IBM customer account representatives can access our test report packages on the MKTTOOLS conference disk, either for their own use or to give to customers. The packages contain tersed LIST3820 files for printing to a 3820-style printer. Complete instructions are in the packages. The following packages are available:

- OS390T99—our December 1999 edition
- OS390T98—our December 1998 edition
- OS390T97—our December 1997 edition (as refreshed in March, 1998)
- OS390TST—our December 1996 edition
- S390TEST—our December 1995 edition
- OS390PSR—*OS/390 Parallel Sysplex Recovery*

Starting in 2000, new editions will no longer be sent to MKTTOOLS, but will be available on the IBM intranet.

Availability on the IBM intranet: IBM customer account representatives can also access all editions and formats of our test report through the Server Sales intranet site, either for their own use or to give to customers. The intranet site links to the same information that's available on our external Web site.

Document Distribution

Internet Access from the IBMLink Home Page: On the Internet World Wide Web, you can access the IBMLink Home Page at *http://www.ibmlink.ibm.com*

To browse the report online:
1. Under “United States”, select “IBMLink”.
2. On the “Public Information and Services Main Menu”, select “IBMManuals”.
3. Select “OS/390” or “MVS/ESA” for the OS/390 or the S/390 MVS editions of the report, respectively.
4. Select “Parallel Sysplex Test Report”.

To download files for printing to a PostScript or 3820-style printer, or viewing with IBM BookManager/READ:
1. Under “United States”, select “IBMLink”.
2. On the “Public Information and Services Main Menu”, select “InfoLinkCenter”.
3. Select “S390—System 390 Information”.
4. Select “SYSPLEX—Parallel Sysplex Test Report (Download Packages)”.
5. Click on “SYSPLEX Parallel Sysplex Test Report” and click on “SelectFiles”.
6. Select the edition and format (either the S/390 MVS or the OS/390 version of the report in either a LISTPS, which is the PostScript format, a LIST3820 format, or a BOOK format for use with IBM BookManager/READ).

Availability through the IBMLink InfoCenter Host Environment: If you are a registered IBMLink subscriber, you can access the report as a download package (in PostScript format for printing to a PostScript printer and in LIST3820 format for printing to a 3820-style printer) in the InfoCenter host environment:
1. Select “InfoLink” on the IBMLink Main Menu.
2. Select “INFOCTR” on the InfoLink Main Menu.
3. Select “S390” on the InfoLink Center Main Menu.
4. Select “SYSPLEX” on the System 390 Information Menu.
5. Select the type of transfer desired (PC, Host, or Internet).
6. Select the edition and format (either the S/390 MVS or the OS/390 version of the report in either a LISTPS, which is the PostScript format, a LIST3820 format, or a BOOK format for use with IBM BookManager/READ).

Appendix D. Useful Publications and Web Sites

This chapter contains listings of the IBM product publications, Redbooks, and Web sites, as well as other non-IBM publications, that we reference in this edition or previous editions of our test report:

IBM Product Publications

The following are the official IBM product publications, listed by product, that we reference in our test reports. For further details about OS/390 documentation, see *OS/390 Information Roadmap*.

ATM Publications: (see also Table 70 on page 230)

Table 19. ATM Publications

Publication Title	Order Number
IBM 8285 Nways ATM Workgroup Switch: Installation and User's Guide	SA33-0381
IBM 8260/8285 ATM Command Reference Guide	SA33-0385
IBM 8285 Safety and Service Guide	SA33-0398
IBM 8260 ATM Control Point and Switch Module Installation and User's Guide	SA33-0326
IBM 8260 ATM 155 Mbps Flexible Concentration Module Installation and User's Guide	SA33-0358
IBM 8260/8285 ATM TR/Ethernet LAN Bridge Module Installation and User's Guide	SA33-0361

AIX Publications: (see also Table 70 on page 230)

Table 20. AIX Publications

Publication Title	Order Number
AIX Distributed Database Connection Services/6000 Guide	SC09-1568

BookManager READ/MVS Publications:

Table 21. BookManager READ/MVS Publications

Publication Title	Order Number
IBM BookManager READ/MVS: Installation Planning and Customization	SC38-2035

CICS Version 4 Publications: (see also Table 70 on page 230)

Table 22. CICS Version 4 Publications

Publication Title	Order Number
CICS for MVS/ESA Recovery and Restart Guide	SC33-1698
CICS/ESA Installation Guide Version 4 Release 1	SC33-1163
CICS/ESA Messages and Codes Version 4 Release 1	SC33-1177
CICS/ESA Migration Guide Version 4 Release 1	GC33-1162

Table 22. CICS Version 4 Publications (continued)

Publication Title	Order Number
CICS/ESA Release Guide Version 4 Release 1	GC33-1161
CICS/ESA Resource Definition Guide Version 4 Release 1	SC33-1166
CICS/ESA System Definition Guide Version 4 Release 1	SC33-1164

CICS TS Publications:

Table 23. CICS TS Publications

Publication Title	Order Number
CICS Customization Guide	SC33-1683
CICS Transaction Server for OS/390 Installation Guide	GC33-1681
CICS Intercommunication Guide	SC33-1695
CICS Transaction Server for OS/390 Migration Guide	GC34-5353
CICS Operations and Utilities Guide	SC33-1685
CICS Recovery and Restart Guide	SC33-1698
CICS System Definition Guide	SC33-1682
CICS Transaction Server for OS/390 Release Guide	GC34-5352
CICS Resource Definition Guide	SC33-1684
CICS Transaction Server for OS/390: Planning for Installation	GC33-1789

CICSPlex SM Publications:

Table 24. CICSPlex SM Publications

Publication Title	Order Number
CICSPlex SM Concepts and Planning	GC34-5732
CICSPlex SM User Interface Guide	SC34-5743
CICSPlex SM Administration	SC34-5729

CICSVR Publications:

Table 25. CICSVR Publications

Publication Title	Order Number
CICSVR MVS/ESA V2R3 Implementation Guide	SH19-6971

COBOL Publications:

Table 26. COBOL Publications

Publication Title	Order Number
COBOL for OS/390 & VM Version 2 Release 1 Compiler and Run-Time Migration Guide	GC26-4764-04

DB2 Publications: (see also Table 70 on page 230)

Table 27. DB2 Publications

Publication Title	Order Number
DB2 Administration Guide	SC26-8957
DB2 for OS/390 Application Programming and SQL Guide	SC26-8958
DB2 for OS/390 Data Sharing: Planning and Administration	SC26-8961
DB2 for OS/390 Installation Guide	GC26-8970
DB2 UDB for OS/390 Administration Guide	SC26–9003
DB2 UDB for OS/390 Command Reference	SC26–9006
DB2 UDB for OS/390 Data Sharing: Planning and Administration	SC26-9007
DB2 UDB for OS/390 Installation Guide	GC26-9008
DB2 UDB for OS/390 Release Planning Guide	SC26-9013
DB2 UDB for OS/390 SQL Reference	SC26–9014
DB2 UDB for OS/390 Utility Guide and Reference	SC26-9015

DCE and DFS Publications: (see also Table 70 on page 230)

Table 28. DCE Publications

Publication Title	Order Number
OS/390 Distributed File Service DFS Administration Guide and Reference	SC28-1720
OS/390 Distributed File Service DFS Configuring and Getting Started	SC28-1722
OS/390 DCE Configuring and Getting Started	SC28-1583
OS/390 DCE Planning	SC28-1582

DFSMS/MVS Publications:

Table 29. DFSMS/MVS Publications

Publication Title	Order Number
OS/390 DFSMS Access Method Services for Catalogs	SC26-7326
OS/390 DFSMSdfp Storage Administration Reference	SC26-7331
OS/390 DFSMS: Implementing System-Managed Storage	SC26-7336
OS/390 DFSMS: Managing Catalogs	SC26-7338
OS/390 DFSMS Migration	SC26-7329

DRDA Publications:

Table 30. DRDA Publications

Publication Title	Order Number
Distributed Relational Database Architecture: Evaluation and Planning Guide	SC26-4650

eNetwork Communications Server Publications: (see also Table 70 on page 230)

Table 31. eNetwork Communications Server Publications

Publication Title	Order Number
eNetwork Communications Server: IP CICS Sockets Guide	SC31-8518
eNetwork Communications Server: IP Configuration Guide	SC31-8726
eNetwork Communications Server: IP Planning and Migration Guide	SC31-8512
OS/390 eNetwork Communications Server: IP User's Guide	GC31-8514
eNetwork Communications Server: IP Programmer's Reference	SC31-8515
eNetwork Communications Server: SNA Planning and Migration Guide	SC31-8622
eNetwork Communications Server: SNA Messages	SC31-8569
eNetwork Communications Server: SNA Network Implementation	SC31-8563
eNetwork Communications Server: SNA Resource Definition Reference	SC31-8565
eNetwork Communications Server: SNA Operation	SC31-8567

Firewall Technologies Publications: (see also Table 70 on page 230)

Table 32. Firewall Technologies Publications

Publication Title	Order Number
OS/390 SecureWay Security Server Firewall Technologies Guide and Reference	SC24-5835
Stay Cool on OS/390: Installing Firewall Technology	SG24-2046

Hardware Publications: (see also Table 70 on page 230)

Table 33. Hardware Publications

Publication Title	Order Number
Hardware Management Console Guide	GC38-0457
Processor Resource/Systems Manager Planning Guide	GA22-7236
Magstar 3494 Tape Library Introduction and Planning Guide	GA32-0279
Magstar 3494 Tape Library Operator Guide	GA32-0280
S/390 9672 and 9674 Operations Guide	GC38-0454
S/390 9672 and 9674 Operations Guide For R2 and R3 Based Models	GC38-3104
S/390 Coupling Facility Model C04; S/390 Parallel Enterprise Server - Generation 3; S/390 Multiprise 2000; Support Element Operations Guide	GC38-3108
S/390 9672 and 9674 System Overview	GA22-7148
S/390 9672 and 9674 System Overview For R1/R2/R3 Based Models	GA22-7148
S/390 Parallel Enterprise Server - Generation 3; S/390 Coupling Facility Model C04; System Overview	GA22-7150
S/390 Parallel Enterprise Server - Generation 4; S/390 Coupling Facility Model C05; System Overview	GA22-7154
S/390 Parallel Enterprise Server - Generation 5 System Overview	GA22-7158

HCD Publications:

Table 34. HCD Publications

Publication Title	Order Number
OS/390 HCD Planning	GC28-1750
OS/390 HCD User's Guide	SC28-1848

High Level Assembler Publications:

Table 35. High Level Assembler Publications

Publication Title	Order Number
HLASM Installation and Customization Guide	SC26-3494

IBM HTTP Server Publications:

Table 36. IBM HTTP Server Publications

Publication Title	Order Number
HTTP Server Planning, Installing, and Using V5.3 for OS/390	SC31-8690
HTTP Server Planning, Installing, and Using V5.3 for OS/390	SC31-8690

For the latest IBM HTTP Server documentation, go to the following Web address:

`http://www.ibm.com/software/webservers/httpservers/library.html#os390`

IBM WebSphere Application Server Publications:

Table 37. IBM WebSphere Application Server for OS/390 Publications

Publication Title	Order Number
Application Server Planning, Installing, and Using V1.1	GC34-4765
Application Server Planning, Installing, and Using V1.2	GC34-4757
WebSphere Application Server Standard Edition Planning, Installing, and Using V3.02	GC34–4806
WebSphere Application Server for OS/390 Getting Started V3.02	GA22–7331

For the latest WebSphere Application Server documentation, go to the following Web address:

`http://www.ibm.com/software/websphere/appserv/library_390.html`

IMS Publications:

Table 38. IMS Publications

Publication Title	Order Number
IMS/ESA Administration Guide: System	SC26-8730
IMS/ESA Administration Guide: Transaction Manager	SC26-8731
IMS/ESA Common Queue Server (CQS) Guide and Reference	LY37-3730
IMS/ESA Database Recovery Control (DBRC) Guide and Reference	SC26-8733

Table 38. IMS Publications (continued)

Publication Title	Order Number
IMS/ESA Installation Volume 1: Installation and Verification IMS/ESA Database Recovery Control (DBRC) Guide and Reference, SC26-8733	GC26-8736
IMS/ESA Installation Volume 2: System Definition and Tailoring	GC26-8737
IMS/ESA Messages and Codes	GC26-8739
IMS/ESA Operations Guide	SC26-8741
IMS/ESA Operator's Reference	SC26-8742
IMS/ESA Release Planning Guide	GC26-8744

Intelligent Miner Publications

Table 39. Intelligent Miner Publications

Publication Title	Order Number
IBM Intelligent Miner for AIX, AS/400, and OS/390 User's Guide	SH12-6213

ISPF Publications:

Table 40. ISPF Publications

Publication Title	Order Number
OS/390 ISPF Planning and Customizing	SC28-1298
OS/390 ISPF User's Guide	SC34-4791

JES2 Publications:

Table 41. JES2 Publications

Publication Title	Order Number
OS/390 JES2 Commands	GC28-1790
OS/390 JES2 Diagnosis	SY28-1086
OS/390 JES2 Initialization and Tuning Guide	SC28-1791
OS/390 JES2 Initialization and Tuning Reference	SC28-1792
OS/390 JES2 Installation Exits	SC28-1793
OS/390 JES2 Messages	GC28-1796
OS/390 JES2 Migration	GC28-1797

JES3 Publications:

Table 42. JES3 Publications

Publication Title	Order Number
OS/390 JES3 Commands	GC28-1798
OS/390 JES3 Migration	GC28-1799
OS/390 JES3 Customization	SY28-1089
OS/390 JES3 Diagnosis	SY28-1090
OS/390 JES3 Initialization and Tuning Guide	SC28-1802

Table 42. JES3 Publications (continued)

Publication Title	Order Number
OS/390 JES3 Initialization and Tuning Reference	SC28-1803
OS/390 JES3 Messages	GC28-1804

LAN CID Publications:

Table 43. LAN CID Publications

Publication Title	Order Number
LAN Configuration, Installation, and Distribution Utility Guide	S10H-9742

Language Environment Publications:

Table 44. Language Environment Publications

Publication Title	Order Number
OS/390 Language Environment Concepts Guide	GC28-1945
OS/390 Language Environment Programming Guide	SC28-1939

LANRES Publications:

Table 45. LANRES/MVS Publications

Publication Title	Order Number
OS/390 LANRES Configuration Files and Commands	SC28-1735
OS/390 LANRES Installation Guide	GC28-1736

LAN Server Publications:

Table 46. LAN Server Publications

Publication Title	Order Number
OS/390 LAN Server Configuration Files and Commands	SC28-1732
OS/390 LAN Server Guide	SC28-1731
OS/390 LAN Server Installation Guide	GC28-1733

LDAP Server Publications:

Table 47. LDAP Server Publications

Publication Title	Order Number
OS/390 SecureWay Security Server LDAP Server Administration and Usage Guide	SC24-5861
OS/390 SecureWay Security Server LDAP Client Application Development Guide and Reference	SC24-5878

MQSeries Publications:

Table 48. MQSeries Publications

Publication Title	Order Number
MQSeries Intercommunication	SC33–1872

Table 48. MQSeries Publications (continued)

Publication Title	Order Number
MQSeries for OS/390 V2R1 System Management Guide	SC34–5374
MQSeries Application Programming Guide	SC33–0807
MQSeries Application Programming Reference	SC33–1673
MQSeries Command Reference	SC33–1369
MQSeries for OS/390 V2R1 Messages and Codes	GC34–5375

MVS Publications:

Table 49. MVS Publications

Publication Title	Order Number
OS/390 MVS Programming: Assembler Services Guide	GC28-1762
OS/390 MVS Programming: Assembler Services Reference	GC28-1910
OS/390 MVS Conversion Notebook	GC28-1747
OS/390 MVS Initialization and Tuning Guide	SC28-1751
OS/390 MVS Initialization and Tuning Reference	SC28-1752
OS/390 MVS Installation Exits	SC28-1753
OS/390 MVS Planning: APPC/MVS Management	GC28-1807
OS/390 MVS Planning: Global Resource Serialization	GC28-1759
OS/390 MVS Planning: Workload Management	GC28-1761
OS/390 MVS Programming: Resource Recovery	GC28-1739
OS/390 MVS Setting Up a Sysplex	GC28-1779
OS/390 MVS System Codes	GC28-1780
OS/390 MVS System Commands	GC28-1781
OS/390 MVS System Management Facilities (SMF)	GC28-1783
OS/390 MVS System Messages, Vol 1 (ABA-ASA)	GC28-1784
OS/390 MVS System Messages, Vol 2 (ASB-ERB)	GC28-1785
OS/390 MVS System Messages, Vol 3 (EWX-IEB)	GC28-1786
OS/390 MVS System Messages, Vol 4 (IEC-IFD)	GC28-1787
OS/390 MVS System Messages, Vol 5 (IGD-IZP)	GC28-1788
OS/390 MVS Programming: Writing TPs for APPC/MVS	GC28-1775

Net.Data Publications:

Publication Title	Order Number
Net.Data Administration and Programing Guide for OS/390	N/A

See the following URL for Net.Data publications:

`http://www.ibm.com/software/data/net.data/library.html`

NetQuestion Publications:

Table 50. NetQuestion Publications

Publication Title	Order Number
OS/390 Text Search: NetQuestion Solution	SH12-6360
OS/390 Text Search: Installation and Administration for the Text Search Engine	SH12-6387

NetView V3 Publications:

Table 51. NetView V3 Publications

Publication Title	Order Number
NetView for MVS Administration and Security Reference	SC31-7080
NetView for MVS Automation and Implementation	SC31-8050
NetView for MVS Command Reference	SC31-8047
NetView for MVS Customization Guide	SC31-8052
NetView for MVS Installation and Administration Guide	SC31-8043
NetView for MVS User's Guide	SC31-8056

Network File System Publications:

Table 52. Network File System Publication

Publication Title	Order Number
OS/390 Network File System Customization and Operation	SC26-7253
OS/390 Network File System User's Guide	SC26-7254

Net.Data Publications:

Table 53. Net.Data Publications

Publication Title	Order Number
Program Directory for Net.Data for OS/390	N/A

Note: This publication is only available as part of the product.

OSA Publications:

Table 54. OSA Publications

Publication Title	Order Number
OSA Planning	GC23-3870
OSA-Express Customer's Guide and Reference	SA22-7403
OSA/SF User's Guide for OSA-2	SC28-1855

OS/390 Overview Publications:

Table 55. OS/390 Publications

Publication Title	Order Number
OS/390 Information Roadmap	GC28-1727

Table 55. OS/390 Publications (continued)

Publication Title	Order Number
OS/390 Introduction and Release Guide	GC28-1725
OS/390 MVS Product Management	GC28-1730
OS/390 Planning for Installation	GC28-1726

OS/390 IBM Communications Servers Publications: (see also Table 70 on page 230)

Table 56. OS/390 IBM Communications Servers Publications

Publication Title	Order Number
OS/390 IBM Communications Server: IP Configuration Reference	SC31-8726
OS/390 IBM Communications Server: IP Migration	SC31-8512

OS/390 UNIX System Services Publications: (see also Table 70 on page 230)

Table 57. OS/390 UNIX System Services Publications

Publication Title	Order Number
OS/390 UNIX System Services Command Reference	SC28-1892
OS/390 UNIX System Services Planning	SC28-1890
OS/390 UNIX System Services User's Guide	SC28-1891

Parallel Sysplex Publications: (see also Table 70 on page 230)

Table 58. Parallel Sysplex Publications

Publication Title	Order Number
OS/390 Parallel Sysplex Application Migration	GC28-1863
OS/390 Parallel Sysplex Hardware and Software Migration	GC28-1862
Note: This book is being removed from the library as of OS/390 V2R10. Refer to the online Parallel Sysplex Configuration Assistant at `http://www.ibm.com/s390/pso/psotool`	
OS/390 Parallel Sysplex Overview	GC28-1860
OS/390 Parallel Sysplex Recovery	GA22-7286
OS/390 Parallel Sysplex Systems Management	GC28-1861
Note: This book is being removed from the library as of OS/390 V2R10. Refer to the online Parallel Sysplex Configuration Assistant at `http://www.ibm.com/s390/pso/psotool`	
OS/390 Parallel Sysplex Test Report	GC28-1963
System/390 MVS Parallel Sysplex Test Report	GC28-1236

PSF Publications:

Table 59. PSF Publications

Publication Title	Order Number
Print Services Facility for OS/390 Customization	S544-5622
Print Services Facility for OS/390 User's Guide	S544-5630

Table 59. PSF Publications (continued)

Publication Title	Order Number
Program Directory for PSF for OS/390	GI10-0228

Print Server Publications:

Table 60. Print Server Publications

Publication Title	Order Number
IBM IP PrintWay Guide	S544-5379
OS/390 Infoprint Server Customization	G544-5694
OS/390 Infoprint Server Introduction	G544-5696
OS/390 Infoprint Server User's Guide	S544-5692

RMF Publications:

Table 61. RMF Publications

Publication Title	Order Number
RMF Report Analysis	SC28-1950
RMF User's Guide	SC28-1949

SAP Publications:

Table 62. SAP Publications

Publication Title	Order Number
SAP R/3 on DB/2 for OS/390: Planning Guide	SC33–7962
SAP R/3 on DB/2 for OS/390: Connectivity Guide	SC33–7695

See also Table 70 on page 230 and "Other Publications" on page 231 for SAP information.

SA OS/390 Publications:

Table 63. SA OS/390 Publications

Publication Title	Order Number
System Automation for OS/390 Defining Automation Policy	GC28-1566
System Automation for OS/390 User's Guide	GC28-1550
System Automation for OS/390 Planning and Installation	CG28-1549
System Automation for OS/390 Technical Reference	GC28-1593

Security Server (RACF) Publications:

Table 64. RACF Publications

Publication Title	Order Number
OS/390 SecureWay Security Server RACF General User's Guide	SC28-1917
OS/390 SecureWay Security Server RACF Security Administrator's Guide	SC28-1915

Table 64. RACF Publications (continued)

Publication Title	Order Number
OS/390 SecureWay Security Server RACF System Programmer's Guide	SC28-1913
OS/390 Security Server (RACF) Support for Digital Certificates	N/A

Note: This publication is only available in the IRR26930 member of SYS1.PARMLIB.

Softcopy Print Publications:

Table 65. Softcopy Print Publications

Publication Title	Order Number
OS/390 Printing Softcopy BOOKs	S544-5354

SOMobjects Publications:

Table 66. SOMobjects Publications

Publication Title	Order Number
OS/390 SOMobjects Service Classes Programmer's Guide	GC28-1809
OS/390 SOMobjects Programmer's Guide	GC28-1859

Tivoli Publications:

Table 67. Tivoli Publications

Publication Title	Order Number
Program Directory for Tivoli Management Framework for OS/390 Framework Endpoint, Version 3.6.1	GI10-4789
Program Directory for Tivoli Management Framework for OS/390 Server and Gateway V3.6.1	GI10-8039
Tivoli Distributed Monitoring for OS/390 Release Notes Version 3.6.1	GI10-8044
Tivoli Global Enterprise Manager Installation User's Guide, V2.2	GC31-5152
Tivoli Inventory for OS/390 Release Notes Version 3.6.1	GI10-8048
Tivoli Management Framework for OS/390 Release Notes: Server, Gateway, and Endpoint	GI10-9186
Tivoli Software Distribution for OS/390 Release Notes Version 3.6.1	GI10-8042
TME 10 Desktop for Windows User's Guide Version 3.6	GC31-8435
TME 10 Distributed Monitoring User's Guide, Version 3.6	GC31-8382
TME 10 Framework Planning and Installation Guide, Version 3.6.1	SC31-8432
TME 10 Framework Reference Manual, V3.6	SC31-8434
TME 10 Framework User's Guide, Version 3.6	GC31-8433
TME 10 Framework Release Notes, V3.6	GI10-3028
TME 10 Framework 3.6.1 Release Notes	GI10-8014
TME 10 Security Management for OS/390 Release Notes, Version 3.6	GI10-4846
TME 10 Security Management Release Notes, Version 3.6	GI10-3049

Table 67. Tivoli Publications (continued)

Publication Title	Order Number
TME 10 Security Management User's Guide, Version 3.6	GC31-8475
TME 10 Security Management User's Guide Supplement for OS/390, Version 3.6	GC32-0297
TME 10 Software Distribution AutoPack User's Guide Version 3.6	GC32-0294
TME 10 Software Distribution Reference Manual Version 3.6	SC31-8331
TME 10 Software Distribution User's Guide Version 3.6	GC31-8330
TME 10 Software Installation Service Release Notes Version 3.6.1	GI10-8015
TME 10 Software Installation Service User's Guide Version 3.6	GC31-5121
TME 10 Tivoli Inventory Supplement for OS/390 Version 3.6.1	GC32-0315
TME 10 User Administration for OS/390 Release Notes, Version 3.6	GI10-4845
TME 10 User Administration Release Notes, Version 3.6	GI10-3040
TME 10 User Administration User and Group Management Guide Version 3.6	GC32-0291
TME 10 User Administration User's Guide Supplement for OS/390, Version 3.6	GC32-0296

TSO/E Publications:

Table 68. TSO/E Publications

Publication Title	Order Number
OS/390 TSO/E Customization	SC28-1965

VisualLift Publications:

Table 69. VisualLift Publications

Publication Title	Order Number
OS/390 VisualLift Run-Time Environment	SC33-6693
VisualLift for MVS, VM, VSE & OS/390 User's Guide	SC33-6691

For the most current IBM WebSphere Application Server books, see the web site at http://www.ibm.com/software/websphere/appserv/library.html.

IBM Redbooks

The following is a list of the IBM Redbooks (so named because they have red covers) we reference in our test reports. These experience-based books can be helpful in setting up and using MVS/ESA and OS/390. They are not shipped with MVS/ESA or OS/390; they must be obtained separately. You can get these books in softcopy formats from IBM's ITSO Redbooks Web site at:

`http://www.redbooks.ibm.com`

Once at the Web site, use the books order number or title as your search string to locate the publication.

Note that these books have not been subjected to any formal review nor have they been checked for technical accuracy, but they represent current product

understanding (at the time of their publication) and provide valuable information.

Table 70. IBM Systems Center Publications (Redbooks)

Publication Title	Order Number
A Beginner's Guide to MVS TCP/IP Socket Programming	GG24-2561
Accessing OS/390 OpenEdition MVS from the Internet	SG24-4721
CICS/ESA and TCP/IP for MVS Sockets Interface	GG24-4026
CICSPlex SM Business Application Services: A New Solution to CICS Resource Management.	SG24-5267
DB2 for MVS Connections with AIX and OS/2	SG24-4558
DB2 UDB for OS/390 Version 6 Performance Topics	SG24-5351
Enhanced Catalog Sharing and Management	SG24-5594
Enterprise Web Serving With the Lotus Domino Go Webserver for OS/390	SG24-2074
Getting Started With DB2 Stored Procedures	SG24-4693
IBM Magstar Virtual Tape Server: Implementation Guide	SG24-2229
IBM 8260 As a Campus ATM Switch	SG24-5003
IBM TCP/IP Version 3 Release 2 for MVS Implementation Guide	SG24-3687
MVS/ESA OpenEdition DCE: Installation and Configuration Experiences	GG24-4480
MVS/ESA SP Version 5 Sysplex Migration Guide	SG24-4581
Object REXX for OS/2 REXX Bytes Objects Now	SG24-4586
OS/2 Installation Techniques: The CID Guide	SG24-4295
OS/390 eNetwork Communications Server TCP/IP Implementation Guide Volume 1: Configuration and Routing	SG24-5227
OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook	SG24-2076
Parallel Sysplex Automation: Using System Automation for OS/390	SG24-5442
Parallel Sysplex Capacity Planning	SG24-4680
S/390 Parallel Sysplex Performance	SG24-4356
SAP R/3 on DB2 for OS/390: Implementing with AIX or Windows NT Application Servers	SG24–4945
SAP R/3 on DB2 UDB for OS/390: Application Servers on OS/390	SG24–5840
Stay Cool on OS/390: Installing Firewall Technology	SG24-2046
TCP/IP in a Sysplex	SG24-5235
Understanding LDAP	SG24-4986
VTAM V4R4 for MVS/ESA Implementation Guide	SG24-2100

IBM Web Sites

The following is a list of some useful IBM Web sites that we reference in this edition or previous editions of our test report:

Table 71. Some Useful IBM Web Sites

Web Site Name	Web Site Address
IBM Glossary of Computing Terms	`http://www.ibm.com/networking/nsg/nsgmain.htm`

Table 71. Some Useful IBM Web Sites (continued)

Web Site Name	Web Site Address
IBMLink	http://www.ibm.com/ibmlink
IBM Redbooks	http://www.ibm.com/redbooks
IBM Systems Center Publications	http://www.ibm.com/support/techdocs
Net.Data Library	http://www.ibm.com/software/data/net.data/library.html
OS/390 Internet Library	http://www.ibm.com/s390/os390/bkserv/
OS/390 Integration Test	http://www.ibm.com/s390/os390/support/os390tst/
Parallel Sysplex Configuration Assistant	http://www.ibm.com/s390/pso/psotool
System Automation for OS/390	http://www.ibm.com/s390/products/sa
Tivoli Systems Home Page	http://www.tivoli.com
Tivoli Systems Support	http://www.tivoli.com/support
UNIX System Services	http://www.ibm.com/s390/unix

Other Publications

We have also used the following non-IBM publications in our testing:

- *DNS and Bind*, by Paul Albitz and Cricket Liu, O'Reilly & Associates, Inc., 1997
- *Learning the KORN Shell*, by Bill Rosenblatt, O'Reilly & Associates, Inc., 1993
- *Learning the vi Editor*, by Linda Lamb, O'Reilly & Associates, Inc., 1990
- *R/3 Installation on UNIX: DB2 for OS/390*, Abbreviation: r3INU Material number: 5100 2695
- *R/3 Installation on Windows NT: DB2 for OS/390*, Abbreviation: r3INN Material number: 5100 2660
- *BC SAP Database Administration Guide: DB2 for OS/390*, Abbreviation: r3DBA Material number: 5100 2661. (SAP documents can be ordered through SAPNet -- R/3 Frontend (formerly: OSS) under: XX-SER-SWFL-SHIP.)
- *UNIX for Dummies*, by J. Levine and M. Young, IDG Books Worldwide, Inc., An International Data Group Company, 1995

IBM Web Sites

Appendix E. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Request

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee. The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

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

Trademarks

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

ADSTAR
Advanced Peer-to-Peer Networking
AFP
AIX
APPN
AS/400
AT
BatchPipes
BookManager
CICS
CICS/ESA
CICSPlex
DATABASE 2
DB2
DB2 Connect
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
Distributed Relational Database Architecture
DRDA
ES/9000
ESA/390
ESCON
FFST
FICON
First Failure Support Technology
GDDM
Hiperbatch
Hummingbird
IBM
IBMLink
ImagePlus
IMS
IMS/ESA
InfoPrint
Intelligent Miner
IP PrintWay
Language Environment
Magstar
MQ
MQSeries
Multiprise
MVS/ESA
Net.Data
NetSpool
Network Station
Nways
Open Class
OpenEdition
Operating System/2
OS/2
OS/390
OS/400
Parallel Sysplex
PR/SM
Print Services Facility
PrintWay
Processor Resource/Systems Manager
QMF
RACF
RAMAC
RETAIN
RISC System/6000
RMF
RS/6000
S/390
S/390 Parallel Enterprise Server
SecureWay
SOM
SOMobjects
SP
Sysplex Timer
System/390
SystemView
VisualAge
VisualLift
VTAM
WebSphere
3090

Ami, Domino, Lotus, and Notes are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli, Tivoli Management Environment, Tivoli Management Framework, TME, TME 10, and TME 10 Global Entrprise Manager are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

The following terms are trademarks of other companies:

AFS and DFS are trademarks or registered trademarks of Transarc Corporation.

Notices

ActionMedia and Pentium are trademarks or registered trademarks of Intel Corporation.

Java is a trademark or registered trademark of Sun Microsystems, Inc.

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

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

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

Glossary

Sources of Terms and Definitions

This glossary defines technical terms and abbreviations used in the *OS/390 Parallel Sysplex Test Report*. If you do not find the term you are looking for, refer to the *IBM Glossary of Computing Terms*, located on the Internet at:

`http://www.ibm.com/networking/nsg/nsgmain.htm`

This glossary also includes terms and definitions from:

- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- Open Software Foundation (OSF). These are identified by the symbol (OSF) after the definition. Copies of OSF documents may be obtained from Open Software Foundation, Inc., 11 Cambridge Center, Cambridge, MA 02142.

Explanation of Cross-References

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

See. This refers the reader to multiple-word terms in which this term appears.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

A

ACB. Access method control block.

ACDS. Active control data set.

ACF. Automation control file.

ACS. Automatic class selection.

ADE. TME 10 Application Development Environment.

ADSM. ADSTAR Distributed Storage Manager.

AEF. TME 10 Application Extension Facility.

AFP. Advanced Function Presentation.

AFS. Andrew file system.

AHHC. APPN host-to-host channel protocol.

AIX. Advanced Interactive Executive.

alternate index (AIX). A collection of index entries related to a given base cluster and organized by an alternate key; that is, a key other the prime key of the associated base cluster data records. The AIX gives an alternative directory for finding records in the data component of the base cluster.

ANO/MVS. Automated Network Operations/MVS.

AOC/MVS. Automated Operations Control/MVS.

AON/MVS. Automated Operations Network/MVS.

AOR. Application-owning region.

APAR. Authorized program analysis report.

APF. Authorized program facility.

API. Application program interface.

APPC. Advanced program-to-program communication.

APM. Application Policy Manager.

APPN. Advanced Peer-to-Peer Networking.

ATL. Automated tape library.

ATM. Asynchronous transfer mode.

AVM. Availability manager.

B

base cluster. A key-sequenced or entry-sequenced data set that one or more alternate indexes can be built over, or a relative-record data set.

basic mode. A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

batch message processing (BMP) program. An IMS batch processing program that has access to online databases and message queues. BMPs run online, but like programs in a batch environment, they are started with job control language (JCL).

batch-oriented BMP program. A BMP program that has access to online databases and message queues

while performing batch-type processing. A batch-oriented BMP does not access the IMS message queues for input or output. It can access online databases, GSAM databases, and MVS files for both input and output.

BDT. Bulk data transfer.

BIND. Berkeley Internet Name Domain.

BMP. Batch message processing (BMP) program.

boot diskette. A diskette created to perform a redirected installation on a client.

BP. Buffer pool.

BPE. Base primitive environment.

BSAM. Basic sequential access method.

C

CA. Certification authority.

cache structure. A coupling facility structure that enables high-performance sharing of cached data by multisystem applications in a sysplex. Applications can use a cache structure to implement several different types of caching systems, including a store-through or a store-in cache.

cache structure services. MVS services that enable applications in a sysplex to perform operations such as the following on a coupling facility cache structure:
- Manage cache structure resources
- Store data into and retrieve data from a cache structure
- Manage accesses to shared data
- Determine when shared data has been changed
- Determine whether a local copy of shared data is valid.

CAE. Client Application Enabler.

CAF. Call attachment facility.

CAS. (1) Catalog address space. (2) Coordinating address space.

CBPDO. Custom Built Product Delivery Offering.

CBR. Cluster bus receiver.

CBS. Cluster bus sender.

CDRM. Cross-domain resource manager.

CDS. Cell directory service.

CDSD. CDS daemon.

CDT. Class descriptor table.

CEC. Synonym for central processor complex (CPC).

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load, and other machine operations.

central processor complex (CPC). A physical collection of hardware that includes main storage, one or more central processors, timers, and channels.

CFR. Coupling facility receiver.

CFRM. Coupling facility resource management.

CFS. Coupling facility sender.

CGI. Common gateway interface.

channel-to-channel (CTC). Refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter (CTCA).

channel-to-channel adapter (CTCA). An input/output device that is used by a program in one system to communicate with a program in another system.

CHPID. Channel path id.

CI. Control interval.

CICS. Customer Information Control System.

CICS TS. CICS Transaction Server.

CICSplex. A group of connected CICS regions.

CICSPlex SM. CICSPlex System Manager.

CICSPlex SM address space (CMAS). A CICSPlex SM component that is responsible for managing a CICSplex. A CMAS provides the single-system image for a CICSplex by serving as the interface to other CICSplexes and external programs. There must be at least one CMAS for each MVS image on which you are running CICSPlex SM. A single CMAS can manage CICS systems within one or more CICSplexes.

CICSVR. CICS VSAM Recovery.

CID. The configuration, installation, and distribution architecture.

CID-enabled product. A product that is CID enabled is a product based on the OS/2 program that can be installed and configured with redirection in a LAN environment with little or no user interaction.

CLAW. Common link access to workstation protocol.

CLI. Command line interface.

CLIST. Command list.

CMAS. CICSPlex SM address space.

CMOS. Complementary metal-oxide semiconductor.

CN. Connection network.

COMMDS. Communications data set.

complementary metal-oxide semiconductor (CMOS). A technology that combines the electrical properties of positive and negative voltage requirements to use considerably less power than other types of semiconductors.

coordinating address space (CAS). An MVS subsystem that provides ISPF end-user access to the CICSplex. There must be at least one CAS on any MVS image from which CICSPlex SM is to be accessed. A CAS must be installed on any MVS image on which a CMAS is installed.

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the MVS systems in a sysplex. See also *sysplex couple data set*.

coupling facility. A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

coupling facility channel. A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a coupling facility and the central processor complexes directly attached to it.

coupling services. In a sysplex, the functions of XCF that transfer data and status between members of a group residing on one or more MVS systems in the sysplex.

COVR. CICS open VTAM retry (COVR) transaction.

CP. (1) Central processor. (2) Control point (VTAM).

CPC. Central processor complex.

CPF. Command prefix.

CPSM. CICSPlex System Manager.

CQS. Common queue server.

cross-system coupling facility (XCF). XCF is a component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

CSA. Common service area.

CSD. CICS system definition.

CSI. Consolidated software inventory.

CSM. Communications storage manager.

CTC. Channel-to-channel.

CTRACE. Component trace.

D

DAE. Dump analysis and elimination.

daemon. A long-lived process that runs unattended to perform continuous or periodic system functions, such as network control. Some daemons are triggered automatically to perform their task; others operate periodically. (OSF)

DASD. Direct access storage device.

data sharing. The ability of concurrent subsystems (such as DB2 or IMS DB) or application programs to directly access and change the same data while maintaining data integrity.

data sharing group. See *IMS DB data sharing group* and *DB2 data sharing group*.

database request module (DBRM). A data set member created by the DB2 precompiler that contains information about SQL statements. DBRMs are used in the bind process.

DBCTL. IMS Database Control.

DBRC. Database Recovery Control.

DBRM. Database request module.

DB2. DATABASE 2.

DB2 data sharing group. A collection of one or more concurrent DB2 subsystems that directly access and change the same data while maintaining data integrity.

DB2 PM. DB2 Performance Monitor.

DCE. Distributed Computing Environment.

DCE AS. Distributed Computing Environment Application Support.

DCF. Document Composition Facility.

DCT. Destination control table.

DDCS. Distributed database connection services.

DDF. Distributed data facility.

DEDB. Data entry database.

DES. Data Encryption Standard.

DFS. Distributed File System.

DFSCM. DFS cache manager (DFS client).

DFSMS. Data Facility Storage Management Subsystem.

DFSMShsm. DFSMS Hierarchical Storage Management.

DFSMSrmm. DFSMS Removable Media Manager.

DFSMSsms. DFSMS Storage Management Subsystem.

distributed data facility (DDF). A set of DB2 components through which DB2 communicates with another RDBMS.

Distributed Relational Database Architecture (DRDA). A connection protocol for distributed relational database processing that is used by IBM's relational database products. DRDA includes protocols for communication between an application and a remote relational database management system, and for communication between relational database management systems.

DL/I. Data Language/I.

DLL. Dynamic link library.

DLUR. Dependent logical unit requestor.

DNS. Domain Name System.

DOS. Disk Operating System.

dpAM. IBM SystemView Data Processing Accounting Manager/MVS.

DQM. Distributed queue management facility.

DRDA. Distributed Relational Database Architecture.

DSI. Dynamic system interchange.

DSLO. Distributed systems license option.

DTS. Distributed time service.

E

ECS. Enhanced catalog sharing.

ECSA. Extended common service area.

EDT. Eligible device table.

EE. Enterprise Extender.

EIF. TME 10 Event Integration Facility.

element. (1) A batch job or started task that uses the IXCARM macro to register for automatic restart management services. In the event of an unexpected failure of the element itself or of the system it is running on, MVS automatically restarts the element. (2) One of the products that makes up the base of OS/390.

EMHQ. Expedited message handler queue.

EMIF. ESCON Multiple Image Facility.

EN. End node.

ENF. Event notification facility.

Enterprise Systems Connection (ESCON). A set of products and services that provides a dynamically connected environment using optical cables as a transmission medium.

ENTR. Ethernet or token ring.

entry-sequenced data set (ESDS). A VSAM data set whose records are physically in the same order in which they were added to the data set. ESDS is processed by addressed direct access, or addressed sequential access and has no index. Records are added at the end of the data set.

EPDM. IBM SystemView Enterprise Performance Data Manager/MVS.

EREP. Environmental error record editing and printing program.

ESCD. ESCON Director.

ESCM. ESCON Manager.

ESCON. Enterprise Systems Connection.

ESDS. Entry-sequenced data set.

ESI. End station identifier.

ESM. External security manager.

ETR. External Time Reference. See also *Sysplex Timer*.

F

FDBR. Fast database recovery region.

FDDI. Fiber distributed data interface.

FEP. Front end processor.

Fibre Connectivity (FICON). A set of products and services based on NCITS Fibre Channel standards that provides high-performance connectivity using optical cables as a transmission medium.

FICON. Fibre Connectivity.

FFST. First Failure Support Technology

FMID. Function modification identifier.

FOR. File-owning region.

forward recovery. The CICSVR function that reapplies all changes to the VSAM sphere since the last backup. The sphere can be a KSDS, ESDS, RRDS, or VRRDS.

CICSVR gets the information it needs to construct the recovery job from the RCDS. The contents of the logs are applied to the VSAM sphere to return it to its exact state before the data was lost. With CICSVR forward recovery, CICSVR restores a DFSMShsm backup for you.

frame. For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

FSA. Functional subsystem application.

FSS. Functional subsystem.

FSSDEF. Functional subsystem initialization statement.

FTP. File transfer protocol.

G

GA. General availability.

GBP. Group buffer pool.

GDAD. Global directory agent daemon.

GDDM. Graphical Data Display Manager.

GDG. Generation data group.

GEM. TME 10 Global Enterprise Manager.

global resource serialization. A function that provides an MVS serialization mechanism for resources (typically data sets) across multiple MVS images.

global resource serialization complex. One or more MVS systems that use global resource serialization to serialize access to shared resources (such as data sets on shared DASD volumes).

GO CA. Domino Go Webserver Certificate Authority.

GRECP. Group buffer pool recovery pending.

GSAM. Generalized Sequential Access Method.

GTF. Generalized trace facility.

GUI. Graphical user interface.

H

Hardware Management Console (HMC). A console used to monitor and control hardware such as the System/390 microprocessors.

HCD. Hardware Configuration Definition.

HCM. Hardware Configuration Manager.

HFS. Hierarchical file system.

highly parallel. Refers to multiple systems operating in parallel, each of which can have multiple processors. See also *n-way*.

HIPER. High impact or pervasive APAR.

HLQ. High-level qualifier.

HMC. Hardware Management Console.

HPDT. High performance data transfer.

HPNS. High performance native sockets.

HPR. High performance routing.

HSA. Hardware system area.

HTML. Hypertext markup language.

HTTP. Hypertext transfer protocol.

HTTPS. Hypertext transfer protocol with SSL.

I

ICB. Integrated cluster bus.

ICF. (1) Internal Coupling Facility. (2) Integrated catalog facility.

ICMF. Integrated Coupling Migration Facility.

ICN. Interchange node.

ICR. Internal coupling receiver.

ICS. Internal coupling sender.

ICSF. Integrated Cryptographic Service Facility.

ICSS. Internet Connection Secure Server.

IFP. IMS fast path program.

IKEYMAN. IBM key management utility.

image server. A high-capacity optical storage device or a computer that each computer and image workstation on a network can use to access and retrieve image objects that can be shared among the attached computers and image workstations.

IMS. Information Management System.

IMS DB. Information Management System Database Manager.

IMS DB data sharing group. A collection of one or more concurrent IMS DB subsystems that directly access and change the same data while maintaining data integrity.

IMS/ESA. Information Management System/Enterprise Systems Architecture.

IMS TM. Information Management System Transaction Manager.

in-doubt period. The period during which a unit of work is pending during commit processing that involves two or more subsystems. See also *in-doubt work unit*.

in-doubt work unit. In CICS/ESA and IMS/ESA, a piece of work that is pending during commit processing; if commit processing fails between the polling of subsystems and the decision to execute the commit, recovery processing must resolve the status of any work unit that is in doubt.

integrated operations workstation. A programmable workstation (PWS) from which an individual can access multiple products to perform a set of tasks, in some cases without knowing which particular product performs a specific task.

IOCDS. Input/output configuration data set.

IOCP. Input/output configuration program.

IODF. Input/output definition file.

IODM. ImagePlus Object Distribution Manager MVS/ESA.

IOPD. Input/output problem determination.

IP. Internet protocol.

IPCS. Interactive problem control system.

IPFAF. ImagePlus Folder Application Facility MVS/ESA.

IPL. Initial program load.

IPX. Internet Packet Exchange.

IRLM. Internal resource lock manager.

ISA. Industry standard architecture.

ISPF. Interactive System Productivity Facility.

ITSO. International Technical Support Organization (IBM).

IUCV. Inter-user communication vehicle.

J

JCT. (1) Journal control table (CICS). (2) Job control table (JES3).

JDK. Java Development Kit

JES2. Job Entry Subsystem 2.

JES3. Job Entry Subsystem 3.

K

key-sequenced data set (KSDS). A VSAM data set whose records are loaded in key sequence and controlled by an index.

KSDS. Key-sequenced data set.

L

LAN. Local area network.

LANRES. Local area network resource extension and services.

LCF. Lightweight client framework.

LE. Language Environment.

LEC. LAN emulation client.

LECS. LAN emulation configuration server.

LES. LAN emulation server.

LIC. Licensed Internal Code.

list structure. A coupling facility structure that enables multisystem applications in a sysplex to share information organized as a set of lists or queues. A list structure consists of a set of lists and an optional lock table, which can be used for serializing resources in the list structure. Each list consists of a queue of list entries.

list structure services. MVS services that enable multisystem applications in a sysplex to perform operations such as the following on a coupling facility list structure:
- Read, update, create, delete, and move list entries in a list structure
- Perform serialized updates on multiple list entries in a list structure
- Monitor lists in a list structure for transitions from empty to non-empty.

lm. Login monitor.

lock structure. A coupling facility structure that enables applications in a sysplex to implement customized locking protocols for serialization of application-defined resources. The lock structure supports shared, exclusive, and application-defined lock states, as well as generalized contention management and recovery protocols.

lock structure services. MVS services that enable applications in a sysplex to perform operations such as the following on a coupling facility lock structure:
- Request ownership of a lock
- Change the type of ownership for a lock
- Release ownership of a lock
- Manage contention for a lock

- Recover a lock held by a failed application.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also *logically partitioned (LPAR) mode*.

logically partitioned (LPAR) mode. A central processor complex (CPC) power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including central processors, central storage, expanded storage, and channel paths) among logical partitions. Contrast with *basic mode*.

loosely coupled. A multisystem structure that requires a low degree of interaction and cooperation between multiple MVS images to process a workload. See also *tightly coupled*.

LP. Logical partition.

LPAR. Logically partitioned (mode).

LPD. Line printer daemon.

LPL. Logical page list.

LPR. Line printer requestor.

LPT. Line printer.

LSR. Local shared resources.

LU. Logical unit.

M

m-image. The number (m) of MVS images in a sysplex. See also *n-way*.

managed address space (MAS). A CICS system that is being managed by CICSPlex SM.

MAS. (1) Multi-access spool (JES). (2) Managed address space (CICSPlex SM).

MAU. Multistation Access Unit.

massively parallel. Refers to thousands of processors in a parallel arrangement.

MCS. Multiple console support.

MCT. Monitor control table.

MDH. Migration data host.

MEC. Machine engineering change.

member. A specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

MICR/OCR. Magnetic ink character recognition/reader/optical character reader.

microprocessor. A processor implemented on one or a small number of chips.

MIH. Missing interrupt handler.

MIME. Multipurpose internet mail extensions.

mixed complex. A global resource serialization complex in which one or more of the systems in the global resource serialization complex are not part of a multisystem sysplex.

MKKF. Make key file utility.

MLQ. Middle-level qualifier.

MNPS. Multiple node persistent sessions.

MP. Multiprocessor.

MPC. Multipath channel.

MPF. Message processing facility.

MPR. Message processing region.

MPTS. Multi-Protocol Transport Services.

MQI. Message queue interface.

MRO. Multiregion operation.

MSC. Multiple Systems Coupling.

MSGQ. Message queue.

MSU. Millions of service units.

multi-MVS environment. An environment that supports more than one MVS image. See also *MVS image* and *sysplex*.

Multiple Systems Coupling (MSC). An IMS facility that permits geographically dispersed IMS subsystems to communicate with each other.

multiprocessing. The simultaneous execution of two or more computer programs or sequences of instructions. See also *parallel processing*.

multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application. An application program that has various functions distributed across MVS images in a multisystem environment.

multisystem environment. An environment in which two or more MVS images reside in one or more

processors, and programs on one image can communicate with programs on the other images.

multisystem sysplex. A sysplex in which two or more MVS images are allowed to be initialized as part of the sysplex. See also *single-system sysplex*.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

MVS system. An MVS image together with its associated hardware, which collectively are often referred to simply as a system, or MVS system.

MVS/ESA. Multiple Virtual Storage/ESA.

MVSCP. MVS configuration program.

N

NAT. Network address translation.

n-way. The number (*n*) of CPs in a CPC. For example, a 6-way CPC contains six CPs.

NDB. Network database.

NetBIOS. Network basic input output system.

NetView DM. NetView Distribution Manager.

NetView FTP. NetView File Transfer Program.

NetView PM. NetView Performance Monitor.

NFS. Network File System.

NGMF. NetView Graphic Monitor Facility.

NJE. Network job entry.

NLM. NetWare loadable module.

NN. Network node.

NTP. Null time provider.

O

OAM. Object access method.

OAT. OSA address table.

OCS. Outboard Communication Server.

OLDS. Online log data sets.

OLTP. Online transaction processing.

OPC/ESA. Operations Planning and Control.

operating system (OS). Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

OPRCTL. Operator control.

OS/2. Operating System/2.

OS/390. Operating System/390.

OS/390 UNIX System Services (OS/390 UNIX). The set of functions provided by the Shell and Utilities, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the OS/390 operating system that allow users to write and run application programs that conform to UNIX standards.

OSA. Open Systems Adapter.

OSA/SF. Open Systems Adapter Support Facility.

OSAM. Overflow sequential access method.

OSF. Open Software Foundation, Inc.

P

parallel processing. The simultaneous processing of units of work by many servers. The units of work can be either transactions or subdivisions of large units of work (batch). See also *highly parallel*.

Parallel Sysplex. A sysplex that uses one or more coupling facilities.

partitionable CPC. A CPC that can be divided into 2 independent CPCs. See also *physical partition, single-image mode, MP, side*.

PC. Personal computer.

PCE. Processor controller element.

PCI. Peripheral component interconnect.

PDF. Portable Document Format.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

PE. PTF error.

physical partition. Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physically partitioned (PP) configuration. A system configuration that allows the processor controller to use both central processor complex (CPC) sides as individual CPCs. The A-side of the processor controller

controls side 0; the B-side of the processor controller controls side 1. Contrast with *single-image (SI) configuration*.

PID. Process identifier

PlexManager. A service utility that can be used to manage the communication connections between multiple coordinating address spaces (CASs) or between a CAS and its associated CICSPlex SM address spaces (CMASs) and CICSplexes.

PLT. Program list table.

PLU. Primary logical unit.

POR. Power-on reset.

PR/SM. Processor Resource/Systems Manager.

processor controller. Hardware that provides support and diagnostic functions for the central processors.

Processor Resource/Systems Manager (PR/SM). The feature that allows the processor to use several MVS images simultaneously and provides logical partitioning capability. See also *LPAR*.

PS. Physical sequential.

PSF. Print Services Facility.

PSP. Preventive service planning.

PTF. Program temporary fix.

PU. (1) Processor unit (2) Physical unit (SNA).

Q

QOR. Queue-owning region.

QSAM. Queued sequential access method.

R

RACF. Resource Access Control Facility.

RCDS. Recovery control data set.

RCT. Resource control table.

RDBMS. Relational database management system.

recovery control data set (RCDS). One of three identical linear VSAM data sets that contain information about the contents of archived logs and the ISPF dialog interface default values. CICSVR uses this stored information to construct recovery jobs. CICSVR uses three identical RCDSs to reduce the possibility of data loss.

redirected installation. Installation on a client by means of a drive that is remotely attached over a LAN to a code server.

relative-record data set (RRDS). A VSAM data set whose records are loaded into fixed-length slots. The records are accessed by a relative record number (RRN).

RFC. Request for comments.

RFS. Record file system.

RISC. Reduced instruction set computer/cycles.

RLS. Record level sharing.

RMF. Resource Measurement Facility.

RODM. Resource Object Data Manager.

RPL. Request parameter list.

RRDS. Relative-record data set.

RRS. Recoverable resource services.

RRSAF. RRS attachment facility.

RS/6000. RISC System/6000.

RSA. Ring system authority.

RSH. Remote shell.

RSHD. Remote shell server.

RSU. Recommended service upgrade.

RTE. Runtime environment.

RTL. Runtime library.

RTP. Rapid transport protocol.

RTPN. Remote transaction program name.

S

SA OS/390. System Automation for OS/390.

SAD. (1) System activity display. (2) Standalone dump.

SAF. Security authorization facility.

SAPR. Systems assurance product review.

SCA. Systems communication area.

SCDS. Source control data set.

SCP. System control program.

SDEP. Sequential dependent.

SDF. Status display facility.

SDSF. System Display and Search Facility.

SEC. System Engineering Change.

serialized list structure. A coupling facility list structure with a lock table containing an array of exclusive locks whose purpose and scope are application-defined. Applications can use the lock table to serialize on parts of the list structure, or resources outside the list structure.

service point. One of the combinations of products and contexts that is known to the coordinating address space (CAS) to which you are connected.

SFM. Sysplex failure management.

SHCDS. Sharing control data set.

side. A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

single point of control. The characteristic a sysplex displays when you can accomplish a given set of tasks from a single workstation, even if you need multiple IBM and vendor products to accomplish that particular set of tasks.

single system image. The characteristic a product displays when multiple images of the product can be viewed and managed as one image.

single-image (SI) mode. A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with *physically partitioned (PP) configuration*.

single-MVS environment. An environment that supports one MVS image. See also *MVS image*.

single-system sysplex. A sysplex in which only one MVS system is allowed to be initialized as part of the sysplex. In a single-system sysplex, XCF provides XCF services on the system but does not provide signalling services between MVS systems. See also *multisystem sysplex, XCF-local mode*.

SIP. System initialization parameters.

SLA. Service level agreement.

SLR. Service Level Reporter.

SMF. System management facilities.

SMIT. System Management Interface Tool.

SMP/E. System Modification Program Extended.

SMS. (1) Storage Management Subsystem. (2) System-managed storage.

SMS communication data set. The primary means of communication among systems governed by a single SMS configuration. The SMS communication data set (COMMDS) is a VSAM linear data set that contains the current utilization statistics for each system-managed volume, which SMS uses to help balance space usage among systems.

SMS configuration. The SMS definitions and routines that the Storage Management Subsystem uses to manage storage.

SMS system group. All systems in a sysplex that share the same SMS configuration and communications data sets, minus any systems in the sysplex that are defined individually in the SMS configuration.

SNA. Systems network architecture.

SOM. System object model.

SPUFI (SQL Processor Using File Input). A facility of the TSO attachment subcomponent that enables the DB2I user to execute SQL statements without embedding them in an application program.

SQL. Structured query language.

SRDS. Structure recovery data set.

SRT. System recovery table.

SSCP. System services control point.

SSI. Subsystem interface.

SSL. Secure socket layer.

structure. A construct used by MVS to map and manage storage on a coupling facility. See *cache structure, list structure*, and *lock structure*.

support element. A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

symmetry. The characteristic of a sysplex where all systems, or certain subsets of the systems, have the same hardware and software configurations and share the same resources.

SYSCTL. SCP manual control.

SYSLOG. System log.

SYSRES. System residence volume.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. See also *MVS system, parallel sysplex*.

sysplex couple data set. A couple data set that contains sysplex-wide data about systems, groups, and

members that use XCF services. All MVS systems in a sysplex must have connectivity to the sysplex couple data set. See also *couple data set*.

Sysplex Timer. An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the MVS generic name for the IBM Sysplex Timer (9037).

system control element (SCE). Hardware that handles the transfer of data and control information associated with storage requests between the elements of the processor.

System/390 microprocessor cluster. A configuration that consists of central processor complexes (CPCs) and may have one or more coupling facilities.

T

TBM. Terminal buffer manager.

Tcl. Tool Control Language used for DCE scripts.

TCP/IP. Transmission Control Protocol/Internet Protocol.

TDS. Topology Display Server.

TFS. Temporary file system.

TG. Transmission group.

THT. Time history table.

tightly coupled. Multiple CPs that share storage and are controlled by a single copy of MVS. See also *loosely coupled, tightly coupled multiprocessor*.

tightly coupled multiprocessor. Any CPC with multiple CPs.

TIOC. Terminal input output controller.

TMA. Tivoli Management Agent.

TME. Tivoli Management Environment.

TME 10. Tivoli Management Environment 10.

TMR. Tivoli management region.

TOR. Terminal-owning region.

TPNS. Teleprocessing network simulator.

TRLE. Transport resource list element.

TSCF. Target System Control Facility.

TSO/E. Time sharing option extensions.

TSR. Token state recovery.

U

UCB. Unit control block.

UDP. User Datagram Protocol.

uniprocessor (UP). A CPC that contains one CP and is not partitionable.

UP. Uniprocessor.

URL. Universal resource locator.

V

variable relative-record data set (VRRDS). A VSAM data set whose records are loaded into variable-length slots. The records are accessed by a relative record number (RRN).

VB. Variable blocked format.

VIPA. Virtual IP address.

VM. Virtual Machine.

VPN. Virtual private networks.

VR. Virtual route.

VRRDS. Variable relative-record data set.

VRTG. Virtual-route-based transmission group.

VSAM. Virtual Storage Access Method.

VSAM sphere. A base cluster, together with any alternate indexes defined with it.

VSE. Virtual Storage Extended.

VSO. Virtual storage option.

VTAM. Virtual Telecommunications Access Method.

VTS. Virtual tape server.

VVDS. VSAM volume data set.

W

WLM. MVS workload management.

WTE. Web Traffic Express.

X

XCA. External communications adapter.

XCF. Cross-system coupling facility.

XCF PR/SM policy. In a multisystem sysplex on PR/SM, the actions that XCF takes when one MVS

system in the sysplex fails. This policy provides high availability for multisystem applications in the sysplex.

XCF-local mode. The state of a system in which XCF provides limited services on one system and does not provide signalling services between MVS systems. See also *single-system sysplex*.

XES. Cross-system extended services.

XRF. Extended recovery facility.

Index

Numerics

370 Parallel Channel Adapter
  using to channel-attach a workstation to system Z0 66

A

acronyms
  See the Glossary 237
application
  DB2 LOBs 45
    buffer pools 47
    DB2 objects 48
    DB2 set up 47
    HFS set up 47
    IBM HTTP Server set up 47
    Net.Data set up 51
    setting up 46
application enablement
  configuration 63
  workloads 73
ATM LAN emulation
  configuration 65
automation 147
  Parallel Sysplex 147
automount
  using to manage user home directories in an HFS 133

B

BPXPRMxx
  for individual systems 121
  shared 121

C

channel connectivity
  ESCON channels 10
  FICON channels 10
CLIST
  CLISTs we use with NetView and SA OS/390 152
  using with automation 151
compatibility APARs
  OS/390 R10 28
  OS/390 R9 22
concatenated parmlib
  using to migrate from OS/390 R8 to OS/390 R9 22
  using to migrate from OS/390 R9 to OS/390 R10 27
configuration
  application enablement 63
  ATM LAN emulation 65
  coupling facility channels 9
  Ethernet LAN 64
  Infoprint Server 104
  networking 63
configuration *(continued)*
  Parallel Sysplex automation 149
  sysplex hardware 3
  sysplex hardware details 4
    coupling facilities 6
    other sysplex hardware 9
    S/390 servers 5
  sysplex software 11
  Tivoli
    Tivoli Management Framework for OS/390 Framework Endpoint 188
    Tivoli Management Framework for OS/390 Server and Gateway 170
  token-ring LAN 66
  VTAM 13

D

DB2 LOBs
  application 45
    buffer pools 47
    DB2 objects 48
    DB2 set up 47
    HFS set up 47
    IBM HTTP Server set up 47
    Net.Data set up 51
    setting up 46
DB2 UDB for OS/390 V6
  migrating to 41
    planning for 41
distribution
  of this document 215
dynamic enablement
  relation to IFAPRDxx parmlib member 11
dynamic LPA
  description 20

E

e-business Integration Test (ebIT) 64
environment
  networking and application enablement 63
  Parallel Sysplex 3
  workloads 14
EQQXIT74 sample JES2 exit 22, 28
ESCON channels 10
Ethernet
  10BASE-T 65
  Fast Ethernet 65
  Gigabit Ethernet 65
Ethernet LAN
  configuration 64

F

FICON channels 10

H

hardware
  coupling facility channel configuration 9
  sysplex configuration 3
HCD
  UCBs above the 16 megabyte line 19
HFS 109
  /usr
    /spool 125
    mounting file systems 122
    product file systems 122
  BPXPRMxx
    customizing 121
  for the IBM WebSphere Application Server 75
  highlights of our strategy 109
  migrating to shared HFS 109
    procedure 119
  naming conventions 130
  OMVS couple data set 122
  our post-install customization EXEC 131
  samples available on our web site 130
  scenarios 116
    accessing /bin directory 118
    accessing /etc directory 116
    accessing another system's /etc directory 117
  service environment 126
    procedure 128
  set -o logical command 119
  shared HFS
    logical and physical view 110
  shared HFS structures 110
  sysplex root HFS 111
  system specific HFS 114
  using the automount facility to manage user home directories 133
  version HFS 113

I

IATUX19 and IATUX29 sample JES3 exits 22, 28
IBM HTTP Server
  LDAP protection 98
    customizing for 99, 100
  migrating to V5.3 97
    system SSL support 97
  Net.Data 90
  scalable server subsystem 89
  shared HFS 91
    reducing HFS size 94
    setting up 93
  V5.2 89
  WebSphere troubleshooter 89
IBM WebSphere Application Server
  migrating to V1.1 75
  migrating to V1.2
    OS/390 R9 81
  migrating to V3.02 83
    shared HFS 84
  read-only HFS 75
    moving customizable directories from the root 77
IBM WebSphere Application Server *(continued)*
  read-only HFS 75 *(continued)*
    samples 77
    WAS.sh shell script 79
  using V1.2
    OS/390 R10 83
IBMLink InfoCenter host environment
  test report availability 216
IC channels
  description 8
ICB channels
  description 8
ICSF
  configuration 37
  hardware 38
  setting up 37, 38
IFAPRDxx parmlib member
  contents of our IFAPRDxx member 208
  relation to dynamic enablement 11
indirect volume serial support
  SYSRES layout 20
Infoprint Server 103
  Infoprint Server Transforms 106
    adding a transform printer 107
    sample print JCL 108
  IP PrintWay 105
  IP PrintWay functional subsystem (FSS) 105
  IP PrintWay functional subsystem application (FSA) 105
  IP Printway routing 105
  issuing JES3 commands for IP PrintWay 105
  migrating PSF to use AOP Printer Inventory 106
  migrating to 106
  our configuration 104
  port monitor 105
  Print Interface 105
  printer definitions 105
  Printer Inventory 105
  printing strategy overview 103
Integrated cyrptographic service facility
  see ICSF 37
Internet
  test report availability 215
IPX
  used by Novell NetWare server and clients 66

J

JES2
  EQQXIT74 sample exit for OPC/ESA 22, 28
  new data set names for OS/390 R10 27
  new data set names for OS/390 R9 22
  recompiling the JES2 OPC/ESA exit for OS/390 R10 28
  recompiling the JES2 OPC/ESA exit for OS/390 R9 22
JES3
  IATUX19 and IATUX29 sample exits for OPC/ESA 22, 28
  new data set names for OS/390 R10 27
  new data set names for OS/390 R9 22

JES3 *(continued)*
recompiling the JES3 OPC/ESA exit for OS/390 R10 28
recompiling the JES3 OPC/ESA exit for OS/390 R9 22

L

LAN Server
NFS
description 72
LANs
LAN A description 67
LAN B description 68
LAN C description 70
LAN D description 71
token-ring backbone description 66
lcfd 191
LDAP
protecting HTTP Server resources 98
LDAP server
customizing automatic IPL startup 141
migrating to OS/390 R10 141
migrating to TDBM 141

M

MKTTOOLS
test report availability 215
MQSeries
MQSeries-CICS bridge 57
setting up 57
multiple SYSRES volumes 20

N

naming conventions
CICS and IMS subsystem jobnames 13
Net.Data 90
macros
samples 52
NetBIOS
used by OS/2 LAN servers and requesters 66
NetView 148
NetWare
IPX communications protocol 66
Network Station
description 20
networking
configuration 63
ATM LAN emulation 65
Ethernet LAN 64
token-ring LAN 66
workloads 73
NFS
comparing OS/390 NFS and LAN Server NFS 72
migrating to the OS/390 NFS 73

O

OMVS couple data set 122
OPC
description 20
EQQXIT74 sample JES2 exit 22, 28
IATUX19 and IATUX29 sample JES3 exits 22, 28
recompiling the JES2 and JES3 OPC/ESA exit for OS/390 R10 28
recompiling the JES2 and JES3 OPC/ESA exit for OS/390 R9 22
OS/2 LAN server
NetBIOS communications protocol 66
OS/390
data set names for JES2 libraries (OS/390 R10) 27
data set names for JES2 libraries (OS/390 R9) 22
migrating from OS/390 R8 to OS/390 R9 21
migrating from OS/390 R9 to OS/390 R10 27
mixed product levels for R10 27
mixed product levels for R9 21
new data set names for JES3 libraries (OS/390 R10) 27
new data set names for JES3 libraries (OS/390 R9) 22
performance 34
R10 high-level migration process 27
R10 migration 26
R9 high-level migration process 21
R9 migration 21
recompiling REXX EXECs for automation 22, 28
recompiling the JES2 and JES3 OPC/ESA exit for OS/390 R10 28
recompiling the JES2 and JES3 OPC/ESA exit for OS/390 R9 22
summary of new and changed parmlib members for OS/390 R10 207
summary of new and changed parmlib members for OS/390 R9 207
updating parmlib for R10 27
updating parmlib for R9 21
updating RACF templates for R10 26
updating RACF templates for R9 21
OS/390 R10
compatibility APARs 28
OS/390 R9
compatibility APARs 22
OS/390 UNIX System Services
managing an HFS 109
associating a root HFS with a SYSRES volume 128
HFS target volume 127
HFSs mounted read-write 131
naming conventions 130
our post-install customization EXEC 131
samples available on our Web site 130
service environment 126
servicing the HFS 128
SMP/E ZONEEDIT command 129
using the automount facility to manage user home directories 133
migrating to shared HFS 109
OSA-2
ATM feature 63
ENTR feature 63, 65, 66

OSA-2 *(continued)*
FENET feature 63, 65
using FENET between LPARs 19
OSA-Express
ATM feature 63
FENET feature 63, 65
Gigabit Ethernet 63
Gigabit Ethernet feature 65

P

Parallel Sysplex automation 147
CLISTs we use with NetView and SA OS/390 152
experiences using SA OS/390 sysplex automation 165
high-level implementation process 148
migrating to SA OS/390 V1R3 149
NetView
automation NetView 151
command flow 151
illustration of message and command flow in a dual NetView environment 151
networking NetView 151
NGMF 148
RODM 148
running a dual NetView environment 151
SSI 151
recommended service 166
SA OS/390
backup focal point system 148
components 147
coupling facility drain 163
Parallel Sysplex operation center 156
primary focal point system 148
setting trigger conditions for a subsystem 152
steady-state automation 148
sysplex automation enhancements 149, 156
transitional automation 148
setup 148
illustration 149
parmlib members
comments about IFAPRD00 208
related to OS/390 R10 207
related to OS/390 R9 207
performance 211
OS/390 34
RMF reports 211
Print Server 103
publications
used by our team 217

R

REXX
recompiling SA OS/390 REXX EXECs when migrating to OS/390 R10 28
recompiling SA OS/390 REXX EXECs when migrating to OS/390 R9 22
RMF
Monitor I Post Processor Summary Report 211
Monitor III Online Sysplex Summary Report 211
RMF *(continued)*
Workload Activity Report in WLM Goal Mode 212

S

SA OS/390 147
recompiling REXX EXECs when migrating to OS/390 R10 28
recompiling REXX EXECs when migrating to OS/390 R9 22
SAP R/3
configuration 137
installing SAPOSCOL 139
setting up 137
SAPOSCOL
installing 139
scalable server subsystem 89
Security Management for OS/390, TME 10 200
shared HFS 109
BPMPRMxx 121
IBM HTTP
reducing HFS size 94
IBM HTTP Server 91
migrating to
procedure 119
software
sysplex configuration 11
SSL support
IBM HTTP Server V5.3 97

T

TDBM
LDAP backend database 141
Tivoli 167, 169, 188
Application Policy Manager (APM) 168
endpoints 168
gateways 168
GEM Topology Server 168
how OS/390 fits into the Tivoli environment 168
lightweight clients (LCFs) 168
support web sites 203
terminology transition 167
Tivoli Enterprise 167
Tivoli Global Enterprise Manager 168
Tivoli Management Agents (TMAs) 168
Tivoli Management Environment 10 (TME 10) 167
Tivoli Management Framework 167
Tivoli Management Framework for OS/390 167
Tivoli Management Region (TMR) servers and managed nodes 167
TME 10 Framework 167
Topology Display Server (TDS) 168
Tivoli Management Framework for OS/390 Framework Endpoint 188
hardware configuration 188
installation experiences 190
avoiding common problems 194
choosing Ethernet or token ring 191
confirming installation using a web browser 194
figuring out what to install where 190

Tivoli Management Framework for OS/390 Framework Endpoint 188 *(continued)*
installing the lcfd daemon 191
setting up the lcfd daemon 191
starting the lcfd daemon 192
stopping the lcfd daemon 193
TME 10 Security Management for OS/390 200
command line interface (CLI) differences 201
creating, modifying, and deleting groups 202
defining groups 203
defining roles 203
ensuring successful distributes 203
experiences 200
GUI problems with roles under security management 3.6.1 201
identical records in the same profile manager 202
installing version 3.6.1 as a product instead of a patch 201
profiles differ from User Administration 202
RACF role limitation 202
RACF role resource profile 202
recommended service 200
support for RACF resources 202
TME 10 User Administration for OS/390 195
before populating with WPOPUSRS 197
clarifying the release notes workaround 198
deciding what profiles to create 195
deleting user IDs that own resources 200
disabling non-OS390 default and validation policies 196
distributing a profile for the first time 199
ensuring a secure network connection 199
ensuring successful distributes 200
experiences 195
populating the Tivoli database with wpopusrs 197
problems with revoke and resume dates 197, 200
relating performance to profile size 195
setting up profiles on the TMR server 196
using default and validation policies 196
validation problems with WPOPUSRS 197
validation takes longer to populate 197
Tivoli Management Framework for OS/390 Server and Gateway 169
experiences 174
connecting to the OS/390 Tivoli Management Server through the Tivoli Desktop 176
Distributed Monitoring 185
importing software into SIS 181
Inventory 186
potential problem with /etc/services 175
problem starting oserv and lcfd under OS/390 V2R9 177
resolving the ++HOLD action for PTF UW61819 175
running the wserver command 174
running Tivoli CLI commands from OMVS 176
Security Management 182
Software Distribution 184
Tivoli Management Framework for OS/390 Server and Gateway 169 *(continued)*
experiences 174 *(continued)*
Software Installation Service (SIS) 178
Tivoli HFS Service Procedure 177
User Administration 181
hardware configuration 170
planning and prerequisites 170
configuring OMVS environment variables 173
configuring OMVS system variables 173
installing Java on OS/390 170
installing Java virtual machine and X server on the NT 4.0 workstation 174
installing the Tivoli software 171
installing TME 10 Desktop for Windows on Window NT 4.0 Workstation 173
understanding HFS DASD space usage 174
TME 10 Security Management for OS/390 200
TME 10 User Administration for OS/390 195
token-ring LAN
backbone description 66
configuration 66
LAN A description 67
LAN B description 68
LAN C description 70
LAN D description 71
troubleshooter
WebSphere Application Server for OS/390 75

U

UCB
above the 16 megabyte line 19
User Administration for OS/390, Tivoli 195
user home directories in an HFS
managing with automount 133

V

VTAM
configuration 13

W

WebSphere Application Server 75
WebSphere Application Server for OS/390
troubleshooter 75
using 75
WebSphere troubleshooter 89
workload
application enablement 15
batch automatic tape switching 15
database product 17
DB2 batch 18
DB2 data sharing 18
IMS data sharing 17
networking 16
networking and application enablement 73
OS/390 base 15
sysplex batch 18
sysplex OLTP 17

workload *(continued)*
VSAM RLS data sharing 17
workloads 14
World Wide Web
test report availability 215

Readers' Comments — We'd Like to Hear from You

OS/390
Parallel Sysplex Test Report
R9+R10

Publication No. GC28-1963-18

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
GC28-1963-18

IBM

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department B6ZH, Mail Station P350
2455 South Road
Poughkeepsie, NY
12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC28-1963-18